Machine Learning applications in spectroscopy and dynamics

Inaugral-Dissertation

to obtain the academic degree
Doctor rerum naturalium (Dr. rer. nat.)
submitted to the Department of Biology, Chemistry, Pharmacy
of Freie Universität Berlin

by

Kanishka Singh

Berlin, 2023

This thesis was prepared under the supervision of
Prof. Dr. Annika Bande at Helmholtz-Zentrum Berlin
between Dec 2019 and Dec 2023

1st Reviewer: Prof. Dr. Annika Bande
2nd Reviewer: Prof. Dr. Beate Paulus

Date of defence: March 7, 2024

Acknowledgements

This dissertation marks the completion of an eventful academic journey that started in a suburb of Mumbai and ended in the wonderful city of Berlin. For the past four years, I have had the support of many people, without whom an effort as mammoth as a doctoral project would not have been possible. Now that we have arrived at its end, I would like to express my gratitude to them.

First and foremost I express my great appreciation to Prof. Annika Bande, who provided me the opportunity and platform to fully develop as a researcher during my doctoral project. Her measured approach always encouraged me to embark on novel research directions without losing sight of the main project at hand. I would like to thank her for always patiently listening to my research ideas, engaging in scientific discussions, and most of all, being always available to listen to my troubles and taking the time to solve them. I have been fortunate to have had a mentor who has influenced my personal and academic values in such a positive manner.

Through my projects, I have had the opportunity to work with several excellent mentors, each of whom has provided me with valuable lessons that helped me grow as a researcher. As a co-supervisor, Prof. Leser always inspired me to work harder, write better, and be more comprehensive in my research. I would like to thank Prof. Daniel Pelaéz for putting his trust in me to take on and complete the dynamics project, as well as guiding me at each step of this process. Prof. Jeremy Harvey and Prof. Arnout Ceulemans of KU Leuven inspired and encouraged me to pursue my research fearlessly, and it is thanks to them, that I began this journey.

The HEIBRIDS graduate school provided me with the opportunity and support that helped me complete this dissertation. The community of young researchers, professors, and non-academic staff have all helped in the completion of this project in a myriad ways and I would like to thank them for their efforts and support. My research group at Helmholtz Zentrum Berlin helped me settle in Berlin and has always supported me with various issues that arose during the four years I spent in the group. Thorren Kirschbaum has been an exceptional colleague and friend, who has been kind enough to help me in various situations and engaged in various scientific discussions with me that have helped improve my work during this thesis. I have had the incredible chance to support Amir Kotobi and Lisa Sahlmann in their projects, during their research stay in Berlin, and thank them for their co-operation during this time. I also had the opportunity to mentor two great students during this thesis, Qingyuan Zhou and Ka Hei Lee, both of whom were committed during their projects and helped further my research.

This journey would not have commenced, sustained, or completed without the support of my partner, Shreeya, who has motivated me to follow my passion for science and who has patiently supported me through the ups and downs in these four years. This dissertation is a realization of not just my efforts, but also of the tremendous hard work and sacrifices made by my parents who supported me each step of the way. I also would like to thank my brother, who inspired me to take up science as a career. A thank you to all my friends, who have always inspired me to improve in life.

Kanishka Singh

“I believe the target of anything in life should be to do it so well it becomes an art.”

Arsene Wenger

Declaration of Independence

Herewith I certify that I have prepared and written my thesis independently and that I have not used any sources and aids other than those indicated by me. This dissertation has not yet been presented to any other examination authority in the same or a similar form and has not yet been published.

Date: December 19, 2023

Kanishka Singh

To my parents. . .

Abstract

The discovery of useful molecules and new molecular phenomena is one of the cornerstones of human progress. Until the last two centuries, this process was largely driven by empirical evidence and serendipitous discovery. The understanding of physical phenomena at the macro level, driven by Newtonian mechanics, electromagnetism, and thermodynamics, and at the micro level, driven by quantum mechanics, has allowed for a more targeted approach to the discovery of new functional molecules for various applications. Despite these advances, the pace at which such molecules are discovered lags behind the rate of demand for green catalysts, sustainable materials, and effective medicines. A significant factor influencing this is the vastness of the chemical space of molecules. It has been approximated that within this chemical space, there exist approximately 10^{60} organic molecules (with a molecular weight less than 500, containing atoms H, C, N, and S). This count will be several magnitudes higher if larger molecules and extended structures are taken into account. Cataloging the properties of these molecules is not currently possible with our computational existing capabilities, but it is essential to find better materials and more effective drugs. As a result, the search for methods that can help speed up the assessment of the properties of molecules and accelerate the discovery of new molecules is an issue of paramount importance in modern chemistry. Machine Learning (ML) algorithms for predicting chemical properties represent an important step in this direction. Not only are ML algorithms capable of learning accurate structure-property relationships, but they are also faster than experiments or quantum chemical simulations. Furthermore, some ML methods leverage the structure-property relationships learned from data to generate novel molecules with desired properties, providing a cost-efficient way to identify useful molecules for laboratory synthesis.

The spectrum of a molecule is one such important molecular property that helps scientists identify different molecules without destroying them. Amongst the various techniques of spectroscopy, X-ray Absorption Spectroscopy (XAS) is a well-established technique that provides information about the structure and composition of various materials. The identification of materials using XAS, however, is not straightforward and requires using a combination of experimental data and quantum-chemical calculations performed on large computing clusters. These computational evaluations are resource-intensive and one often needs several such calculations to achieve successful molecular identification. Access to methods that can accelerate the prediction of spectra through structure-property relationships in spectroscopy can greatly enhance the ability to identify compounds synthesized in laboratories. Therefore, a major part of this dissertation is dedicated to employing and understanding ML methods that speed up the prediction of spectra by learning structure-property relationships from data. This work lays a foundation for future applications, where ML models can be used in experimental setups to identify molecules from spectra without human intervention, thereby helping accelerate the synthesis and identification of novel compounds. One downside of ML applications is the lack of model interpretability, which decreases the trust of the end-users. Investigations in this dissertation focus on devising a technique that helps humans understand why ML models make certain predictions, thereby helping build trust between the ML model and its end user.

The creation of chemical data for ML applications itself usually requires quantum chemical calculations that involve solving the Schrödinger equation. The time-dependent Schrödinger equation (TDSE) helps understand the behavior of quantum

systems and allows for the calculation of time-dependent properties of molecules. The area of research that concerns itself with techniques for solving the TDSE is termed quantum dynamics. Using computer simulations of numerical methods for solving this equation, researchers have modeled several quantum dynamical systems, which have improved our understanding of photo-catalysis (reactions driven by light), surface phenomena such as chemisorption, and chemical reaction pathways. The second part of this dissertation focuses on using ML methods to solve the TDSE.

The TDSE, which is a partial differential equation (PDE) in space and time, is one of the many fundamental equations that help model the behavior of physical systems. Some other notable PDEs that play an important role in physics and engineering are the Navier-Stokes equation for modeling fluids, the Heat equation in thermodynamics, and the wave equation in acoustics. Numerical techniques for solving PDEs are based on the discretization of the coordinate space into finite elements. As the size and dimensions of the grids increase, these methods become computationally expensive. As a result, solving PDEs such as the TDSE for large molecular systems is computationally demanding or even impossible. Advances in ML for solving PDEs aim at accelerating the solution of PDEs through a data-driven approach. In the second part of this thesis, ML models were trained on simulation data from quantum dynamical systems. Once trained, these models are capable of providing accurate descriptions of the behavior of systems that were not seen during training. A key advantage of such methods is their ability to generate novel simulations accurately and at high speed. As a proof of concept, the work in this dissertation shows how this speed can be exploited for downstream applications in quantum dynamics.

Zusammenfassung

Die Entdeckung funktionaler Moleküle und neuer molekularer Phänomene ist einer der Eckpfeiler des menschlichen Fortschritts. Bis vor zwei Jahrhunderten wurde dieser Prozess weitgehend durch empirische Beweise und zufällige Entdeckungen vorangetrieben. Das durch die Newtonsche Mechanik, den Elektromagnetismus und die Thermodynamik geförderte Verständnis physikalischer Phänomene auf der Makroebene und das durch die Quantenmechanik ermöglichte Verständnis auf der Mikroebene haben ein gezielteres Vorgehen bei der Entdeckung neuer funktioneller Moleküle für verschiedene Anwendungen ermöglicht. Trotz dieser Fortschritte bleibt das Tempo neuer Entdeckungen solcher Moleküle hinter dem Bedarf an umweltfreundlichen Katalysatoren, nachhaltigen Materialien und wirksamen Medikamenten zurück. Ein wichtiger Faktor, der dies beeinflusst, ist die enorme Anzahl der existierenden Molekülstrukturen. Schätzungen zufolge gibt es etwa 10^{60} organische Moleküle (mit einem Molekulargewicht von weniger als 500 und den Atomen H, C, N und S). Diese Zahl ist noch deutlich höher, wenn größere Moleküle und weitere mögliche Strukturen berücksichtigt werden. Die vollständige Katalogisierung der Eigenschaften dieser Moleküle ist mit den derzeit verfügbaren Methoden nicht möglich, aber für die Suche nach besseren Materialien und wirksameren Arzneimitteln ist sie unerlässlich. Daher ist die Suche nach Methoden, die eine schnellere Einschätzung der Eigenschaften von Molekülen ermöglichen und damit die Entdeckung neuer Moleküle beschleunigen können derzeit von größter Bedeutung. Algorithmen des maschinellen Lernens (ML) zur Vorhersage chemischer Eigenschaften sind ein wichtiger Schritt in diese Richtung. ML-Algorithmen sind nicht nur in der Lage, genaue Struktur-Eigenschafts-Beziehungen zu lernen, sondern sie sind auch schneller als Experimente oder quantenchemische Simulationen. Darüber hinaus nutzen einige ML-Methoden die aus den Daten erlernten Struktur-Eigenschafts-Beziehungen, um neuartige Molekülstrukturen mit den gewünschten Eigenschaften zu erzeugen. Dies stellt eine kosteneffiziente Möglichkeit zur Identifizierung neuer funktionaler Moleküle dar, die anschließend im Labor synthetisiert werden können.

Das Spektrum eines Moleküls ist eine wichtige molekulare Eigenschaft, die Wissenschaftlern hilft, die Eigenschaften von Molekülen zu identifizieren, ohne sie zu zerstören. Unter den verschiedenen Techniken der Spektroskopie ist die Röntgenabsorptionsspektroskopie (*X-ray absorption spectroscopy*, XAS) eine etablierte Technik, die Informationen über die Struktur und Zusammensetzung verschiedener Materialien liefert. Die Identifizierung von Materialien anhand von XAS ist jedoch nicht einfach und erfordert eine Kombination aus experimentellen Methoden und quantenchemischen Berechnungen, die auf großen Computerclustern durchgeführt werden. Die rechnerischen Auswertungen sind ressourcenintensiv und können mehrere Iterationen erfordern, um zu einer erfolgreichen molekularen Identifizierung zu gelangen. Der Zugang zu Methoden, die die Vorhersage von Struktur-Eigenschafts-Beziehungen in der Spektroskopie beschleunigen, kann die Fähigkeit zur Identifizierung von in synthetischen Verbindungen erheblich verbessern. Daher ist ein großer Teil dieser Arbeit der Anwendung von ML-Methoden gewidmet, die die effiziente Vorhersage von Spektren durch das Lernen von Struktur-Eigenschafts-Beziehungen aus Daten ermöglichen. Diese Arbeit legt den Grundstein für künftige Anwendungen, bei denen ML-Modelle in Versuchsaufbauten verwendet werden können, um Moleküle aus Spektren ohne menschliches Eingreifen zu identifizieren und so die Synthese und Identifizierung neuer Verbindungen zu beschleunigen. Ein Nachteil von ML-Anwendungen ist die mangelnde Interpretierbarkeit der Modelle, was das

Vertrauen der Endnutzer und manchmal auch die Genauigkeit der ML-Modelle beeinträchtigt. Weitere Untersuchungen im Rahmen dieser Arbeit konzentrieren sich auf die Entwicklung einer Technik, die den Menschen hilft zu verstehen, warum ML-Modelle bestimmte Vorhersagen treffen, und so dazu beiträgt, das Vertrauen der Endnutzer in die ML-Modelle zu stärken.

Die Erstellung chemischer Daten selbst für ML erfordert normalerweise quantenchemische Berechnungen, bei denen die Schrödinger-Gleichung gelöst wird. Der zweite Teil dieser Arbeit konzentriert sich auf die Verwendung von ML zur Lösung der zeitabhängigen Schrödinger-Gleichung (*time-dependent Schrödinger equation*, TDSE), die nicht nur zum Verständnis des Verhaltens von Quantensystemen beiträgt, sondern auch die Berechnung zeitabhängiger Eigenschaften von Molekülsystemen ermöglicht. Das Forschungsgebiet, das sich mit Techniken zur Lösung der TDSE beschäftigt, wird als Quantendynamik bezeichnet. Durch die Anwendung numerischer Methoden zur Lösung dieser Gleichung haben Forschende verschiedene quantendynamische Systeme modelliert, die unser Verständnis der Photokatalyse (durch Licht ausgelöste Reaktionen), von Oberflächenphänomenen wie der Chemisorption und von chemischen Reaktionswegen maßgeblich verbessert haben.

Die TDSE ist eine partielle Differentialgleichung (*partial differential equation*, PDE) in Raum und Zeit und ist eine der vielen grundlegenden Gleichungen, die dazu beitragen, das Verhalten von chemischen Systemen zu modellieren. Einige andere nennenswerte PDEs, die in der Physik und im Ingenieurwesen eine wichtige Rolle spielen, sind die Navier-Stokes-Gleichung für die Modellierung von Flüssigkeiten, die Wärmeleitungsgleichung in der Thermodynamik und die Wellengleichung in der Akustik. Numerische Verfahren zur Lösung von PDEs beruhen auf der Diskretisierung des Koordinatenraums in eine endliche Zahl von Elementen. Mit ansteigender Größe und Dimension des Gitters werden diese Methoden zunehmend rechenintensiv. Infolgedessen ist die Lösung von PDEs wie der TDSE für große molekulare Systeme sehr aufwändig oder sogar unmöglich. Die Anwendung von ML für die Lösung von PDEs zielt darauf ab, sie durch einen datengetriebenen Ansatz zu beschleunigen. Im zweiten Teil dieser Arbeit wurden ML-Modelle auf Simulationsdaten von quantendynamischen Systemen trainiert. Die so trainierten Modelle sind anschließend in der Lage, genaue Beschreibungen des Verhaltens von Systemen zu liefern, die während des Trainings nicht gesehen wurden. Ein entscheidender Vorteil solcher Methoden ist ihre Fähigkeit, neue Simulationen mit hoher Genauigkeit und Geschwindigkeit zu berechnen. In dieser Arbeit wird gezeigt, wie diese Methode für nachgelagerte Anwendungen in der Quantendynamik genutzt werden kann.

List of Publications

Peer-reviewed and submitted publications that are part of this thesis sorted by date

1. Kanishka Singh, Jannes Münchmeyer, Leon Weber, Ulf Leser and Annika Bande
   Graph Neural Networks for learning molecular excitation spectra
   J. Chem. Theory Comput. 18,7, 4408-4417 (2022)
   DOI: 10.1021/acs.jctc.2c00255

2. Amir Kotobi, Kanishka Singh, Daniel Höche, Sadia Bari, Robert Meißner and Annika Bande
   Integrating Explainability into Graph Neural Network Models for the prediction of X-ray Absorption Spectra
   J.Am.Chem.Soc 145,41, 22584-22598 (2023)
   DOI: 10.1021/jacs.3c07513

3. Kanishka Singh, Ka Hei Lee, Daniel Pelaéz and Annika Bande
   Accelerating wavepacket propagation with Machine Learning
   Submitted to J.Comput.Chem.
   DOI: NA

Contents

List of Figures

Chapter 1

Introduction

The progression of science is often categorized into four paradigms [1, 2], each reflective of the dominant techniques that characterized scientific investigations during that era. While early science was dominated by experimental discovery, a foundational understanding of scientific phenomena developed only after the discovery of laws that led to the rise of theoretical science. As theoretical models became too complicated to solve for humans, simulations on computers followed giving rise to the age of computational modeling. The vast amounts of data that can be generated from these computer models have enabled the rise of the fourth paradigm of scientific development, the data-driven exploration of scientific phenomena.

Computational simulations based on quantum chemical methods are an integral part of modern chemical research. Various disciplines within this field have not only helped understand the underlying science behind various molecular phenomena but have also aided in the design of structure-property relationships [3] that ultimately accelerate the design of novel molecules. The ability to reliably understand and evaluate the chemical properties of structures without relying on expensive experimental investigations has made computational chemistry increasingly crucial to fields such as drug discovery and materials research. While the past century has been a testament to these advances made in quantum chemistry, driven by methods and algorithms aimed at solving the Schrödinger equation [4], existing computational capabilities are still deficient for providing quick and accurate solutions for large molecular systems, or a large number of small systems. With the growing demand for novel materials to support human technological development and the search for novel drugs to cure human diseases while maintaining sustainability, the need to develop fast and accurate methods for predicting chemical properties is greater than ever.

Data-driven chemistry offers a promising way out of these problems. Machine Learning (ML) methods have already shown that they can accurately model relationships from chemical data [5]. In contrast to the conventional methods of quantum chemistry, these methods do not suffer from computational issues that lead to high evaluation times for large molecular systems. This ability to infer complex relationships from various types of chemical data without suffering from computational scaling issues has led to the rapid rise of ML applications in the prediction of various chemical properties [6, 7], predicting chemical reaction outcomes [8, 9], generating potential energy surfaces [10], and molecular dynamics [11, 12].

This dissertation explores the use of ML for two applications in computational chemistry, i.e. spectroscopy and quantum dynamics. The first part is devoted to the generation of structure-property relationships for the prediction of molecular spectra using ML. Understanding the decision-making process of ML models is an arduous task. Therefore, an attempt is made in this thesis to develop a method that can explain the working of ML models in spectroscopy. The second application of ML in this dissertation explores the solution of the time-dependent Schrödinger Equation (TDSE) using a data-driven approach. The forthcoming sections of this chapter will be devoted to providing a brief introduction to each of these topics. Chapters 2-6 provide details on the methods of computational chemistry and the ML frameworks that were employed during this dissertation. The results of the dissertation are gathered in Chapter 7.

1.1 Machine Learning for the prediction of spectra

Molecules have been identified from their characteristic experimental spectroscopic fingerprints for more than a century. The basic principle underlying spectroscopy is the energy transfer between a photon and the material under investigation. The interactions with different energy photons cause bonds in molecules to vibrate or lead to electronic excitations which lend them different fingerprints allowing for their identification in different energy ranges. These varied effects arising from interactions with photons of different energies have led to the rise of different spectroscopic techniques such as X-ray, infrared (IR), Raman, ultraviolet (UV), etc.

The knowledge of spectroscopic fingerprints precedes the development of quantum mechanics with techniques such as the Rydberg formula [13, 14] being used for understanding the absorption spectra of alkali metals. However, the rapid development of quantum mechanics in the early 1900s helped provide interpretations of the spectra from a molecular structure perspective, thereby making spectroscopy a crucial technique for identifying structures of complex materials. Investigation of molecular spectra today is a combination of experimental and theoretical methods. Once a spectral fingerprint is obtained from experiments, using a combination of other ancillary techniques and human expertise several candidate structures to which the spectrum could belong are identified. Then using accurate computational simulations, spectra are generated for candidate molecules, and the closest matching spectrum is used as a confirmation of the candidate structure.

The computational simulation techniques based on quantum chemistry methods for predicting spectra have been developed and fine-tuned over the past years, and are central to the process of structural identification. Prominent theoretical methods of spectroscopy usually employ quantum chemical techniques based on Density Functional Theory (DFT) or Hartree Fock methods, which scale in the order of N^3 to N^4 [15, 16] (where N is the number of electrons in the system). This poor scaling with respect to system size makes the simulation of a spectrum for large molecules a computationally expensive and time-consuming process. A structure-property relationship that can quickly and accurately provide the spectrum of a molecule given its structure, therefore, can be especially useful when a molecule contains a large number of atoms, and the number of possible candidate structures is large. An ML method that can accurately model the structure-property relationship of molecules and their spectra is therefore of immense value in various chemical applications to automatically screen over several candidates quickly and provide the closest matching spectrum to the one obtained from an experiment. This dissertation investigates structure-property relationships for two types of spectra, an excitation spectrum obtained from the calculation of orbital energies of small molecules and X-ray absorption spectra (XAS) of molecules. A discussion of the former is provided later in this dissertation, while the next paragraph focuses on the technique of XAS.

X-ray spectroscopic techniques such as XAS are one of the most prominent tools for the identification and characterization of materials. The development of X-ray spectroscopy can be traced to the discovery of the photoelectric effect [17], with X-ray Photoemission Spectroscopy (XPS) [18] being the first technique formally developed for the investigation of material composition. XPS originates when the bound core electrons of a molecule are excited to a vacuum, ionizing the system. XPS depends on the excitation energies of the atoms in different environments. When the core electrons of a molecule are excited to a higher unoccupied state instead of a vacuum, however, the XAS of a molecule is obtained. Unlike XPS, XAS depends on the unoccupied states of the molecule giving more insights into the chemical structure surrounding the atom of interest than XPS [19]. The origin of XAS from an excitation caused by an incident photon is schematically depicted in Figure 1.1. The figure depicts the core and unoccupied orbitals of a molecule interacting with an incident photon. This photon in the XAS range excites the core electrons of the molecule, which occupy the low-lying energy levels in the figure. Once excited, the electron has access to a variety of unoccupied orbitals which it can end up in, depending on the energy of the impacting photon. These varied electron transitions lead to the absorption spectrum shown on the top right side of the figure. Computational methods for evaluating XAS require calculations of transition energies between the different occupied and unoccupied states. As the number of atoms in a molecule increases, so does the number of electrons, and the number of orbitals, requiring a large number of calculations to obtain these energies. As a result, the calculation of XAS of large structures is computationally expensive. ML methods predicting XAS can help predict quick and accurate spectra, as well as improve our understanding of the technique.

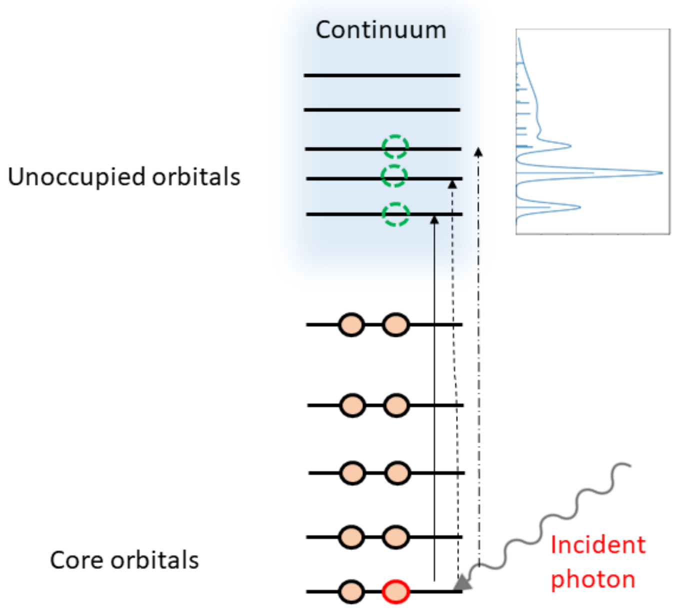


FIGURE 1.1: *A pictorial description of the emergence of an X-ray absorption spectrum. An electron excited by an incident photon can be excited to a wide variety of unoccupied orbitals. Different transitions lead to the formation of various peaks in the final XAS spectrum, as depicted in the top right part of the figure.*

There are several recent works where ML has been used for the prediction of UV-Vis [20–22], IR [23–25] and Raman spectra [26–28]. Ghosh et al. [29] presented a comprehensive benchmarking of various ML methods for spectroscopic predictions in their work. ML methods have also been successfully employed for the characterization of different atomic environments that appear as peaks in the XAS spectrum [30–33]. Other recent investigations have also been made using ML to predict the XAS of molecules [34–36]. Different types of ML architectures can be used for the prediction of spectra (or any other chemical property).

1.2 Explaining XAS predictions made by ML models

With the increase in available computing power and datasets larger ML models are now being trained for several applications every day. Such ML models are often neural networks with a large number of parameters. The complex scale of operations between these parameters makes it humanly impossible to decode the decision-making process of such models. Therefore, while we know that ML models excel at developing relationships for large datasets understanding the reason for their effectiveness often is elusive. ML models that provide little to no insight into their decision-making mechanisms are called black box models.

Using black box ML models without understanding them, can have adverse consequences. An ML model can, for example, give incorrect decisions that have a significant impact (e.g. in healthcare applications). These errors can arise if the ML model develops misleading correlations caused by dataset bias or erroneously assigns excessive importance to features that, based on human understanding, should not heavily impact decision-making. Thus quantifying model performance based on a single accuracy parameter is not often sufficient to justify its usage in applications, and knowing why the model made that prediction is essential [37]. Such knowledge also enhances the trust between models, developers, and their users. It is for these reasons that developing techniques that make ML model predictions interpretable is crucial. For ML models being used in scientific applications interpretability can become a source of knowledge by providing insights into relationships between features in the input data and output predictions. Explainable Artificial Intelligence (XAI) techniques are mathematical tools that help impart interpretability to these otherwise black-box models. Several XAI approaches have already improved our understanding of models used in natural language processing [38] and computer vision [39]. These methods have also been successfully employed in various applications in chemistry to unravel the logic behind different ML models [40–46].

An essential element in developing an XAI approach is having access to a ground truth logic, which serves as a means to validate the outcomes obtained from the application of an XAI method. Such ground truths can be in the form of information about active regions of a molecule that contribute to a property. As an example, in an ML application predicting the solubility of molecules in a polar solvent, the ground truth highlights groups that positively or negatively impact solubility. Sometimes, such ground-truth logic can be obtained using empirical evidence or is often composed of patterns that are easily obtainable using elementary chemical rules. In contrast to this, developing a ground truth logic for properties such as spectra is complicated. X-ray spectra can be especially difficult to interpret as they arise from a combination of structural and physical features of a molecule. The second project in this dissertation uses an XAI method to help understand the XAS predictions made by GNNs. This is a first-of-its-kind approach that provides a recipe for using quantum chemical data for developing a ground truth logic and can also be applied to ML models beyond spectroscopy.

1.3 Machine Learning for quantum dynamics

A large portion of research in quantum mechanics involves investigating phenomena that consider stationary states of electrons in motion under the influence of the electrostatic field of nuclei of molecules. The methods developed in these fields of research can be used for the computation of properties such as ground state energies, dipole moments, orbital densities of systems, and so on. However, several other properties and phenomena require solving the TDSE. Such situations arise in the investigation of phenomena such as quantum tunneling [47], chemical reactions [48], and first principles prediction of vibrational spectra [49].

The solution of the TDSE for systems at molecular, atomic, and subatomic systems forms the basis of research in quantum dynamics. Many approaches have been designed to efficiently solve the propagation of wavepackets representing quantum mechanical systems changing in time. While these methods can accurately model time-dependent phenomena, they scale poorly with the number of variables (degrees of freedom) of the system, limiting their usability for large systems. The Multi Configuration Time Dependent Hartree (MCTDH) [50–52] method for high dimensional quantum dynamics is one such method that uses several efficient procedures for the successful modeling of quantum dynamical phenomena but suffers from scaling issues that prevent its application to larger systems.

As an alternative, ML methods offer the possibility to learn wavepacket propagation from datasets containing quantum dynamical simulations. This area of research in ML, the data-driven solving of Partial Differential Equations (PDEs) has been successfully used to model systems from the domains of fluid dynamics [53], geology [54], and weather modeling [55]. Data-driven solutions of PDEs offer the ability to predict simulations accurately without suffering from the scaling issues of conventional solvers. The final project of this dissertation involves using a popular ML framework from this domain, the Fourier Neural Operator (FNO) architecture [56] to model solutions of the TDSE. The work done in this part of the dissertation shows that FNO models can not only accurately model the quantum dynamics of two model systems but also that their high speed of generating propagations for unseen data can be used for downstream applications, thus making them ideal low-cost surrogate models for approximating quantum dynamical phenomena.

Inverse problems that involve exploring a set of initial conditions that lead to a desired final state form an integral part of many research activities related to PDE solving. An example of such a task in quantum dynamics is laser pulse shaping, where the goal is to attain a desired final state for a system using a laser pulse. Finding such a laser pulse requires optimizing over a large number of system variables, which has been traditionally tackled using methods of optimal control [57–59]. Alternatively, such an optimization can be carried out using Bayesian methods [60]. These methods however require sampling through a large number of wavepacket propagations, which can be time-consuming using conventional solvers. The high speed of ML methods like FNO however allows for sampling quickly through a large number of samples thereby making such Bayesian approaches viable. In this dissertation, the Bayesian technique of Markov Chain Monte Carlo (MCMC) [61] is combined with a trained FNO model to optimize a laser pulse shape which brings about a rotation in a molecule.

Chapter 2

Quantum chemical calculations of spectra

The creation of the datasets used for ML applications in this dissertation relies on quantum chemical methods based on Density Functional Theory (DFT) and Time-Dependent Density Functional Theory (TDDFT). This chapter briefly recaps the concepts and terminologies associated with DFT and TDDFT, their implementations in modern software packages, and how these methods are used to obtain spectra that are present in the datasets of this dissertation. For the discussion of DFT methods, the content from the introductory chapters of textbooks by Koch and Holthausen [62] and Parr and Yang [63] is referred to, while Gross and Maitra [64] and Fiolhais et al. [65] were referred to, for the theory of TDDFT.

2.1 Hohenberg Kohn theorems

The Time Independent Schrödinger Equation (TISE) $H\Psi = E\Psi$ is the foundation of quantum chemical methods and offers a means to determine a system's wavefunction Ψ and its corresponding energy E. While the Hamiltonian H theoretically includes both electronic and nuclear parts, the Born-Oppenheimer approximation provides a means to calculate the electronic wavefunction by solving the electronic Hamiltonian. Solving this Hamiltonian provides a wavefunction for the electrons of the molecular system. For a system with N electrons, the electronic wavefunction $\Psi(\mathbf{r})$ has $3N$ positional variables corresponding to the positions of the electrons in the system. While wavefunction based methods such as the Hartree-Fock (HF) approach are useful for smaller systems, with an increase in system size the computational expenses that arise due to the $3N$ coordinates in the wavefunction, make the HF and post-HF methods too computationally expensive for large molecules.

Hohenberg and Kohn [66] in 1968, formulated an alternative approach that obviates the need for entire wavefunctions and replaces it with an electron density $\rho(r)$ of the system which is a function of three spatial coordinates. The first of the two Hohenberg-Kohn Theorems establishes a direct mapping between the electronic density resulting from an external potential, and the total energy E of the system, in the form of a functional. The first theorem allows for calculating the energy of the system as

$$E_{\text{HK}}[\rho(r)] = V_{ne}[\rho(r)] + T[\rho(r)] + V_{ee}[\rho(r)] = F_{\text{HK}}[\rho(r)] + \int \rho(r) v_{ne}(r) dr. \tag{2.1}$$

Here, the term $V_{ne}[\rho(r)]$ is a functional that represents the interaction potential between electrons and nuclei, $T[\rho(r)]$ is the kinetic energy functional, and the electron-electron repulsion is given by $V_{ee}[\rho(r)]$. The potential energy due to interaction between the electrons and nuclei is the only term whose form depends on the system (in terms of the number of electrons N, the nuclear charge, or the configuration of electrons in the system.), while the other two terms which only depend on the density are considered universal and independent of the configuration of the system. These terms can therefore be collected as one functional termed the Hohenberg Kohn functional $F_{\text{HK}}[\rho(r)]$. Theoretically, the Hohenberg-Kohn functional, when provided with the density of a system gives out the expectation value associated with the kinetic energy and electron-electron repulsion of the ground state of the system. Access to this functional can provide the exact solution of the TISE, completely independent of the system at hand. Such a functional however remains elusive, and approximate functional mappings have been developed and fine-tuned for various systems to accurately calculate chemical properties using DFT.
The electron-electron interaction term of the Hohenberg-Kohn functional can be written as a sum of two parts, as shown below. The term $J[\rho(r)]$ includes the Coulombic electron-electron repulsion, while non-classical contributions arising out of self-interaction and correlation factors are given by $E_{ncl}[\rho(r)]$ as

$$V_{ee}[\rho(r)] = J[\rho(r)] + E_{\text{ncl}}[\rho(r)] \tag{2.2}$$

Incorporating this split of the electronic energy term into the calculation of the total energy of the system leads to the equation

$$E[\rho(r)] = V_{ne}[\rho(r)] + T[\rho(r)] + J[\rho(r)] + E_{\text{ncl}}[\rho(r)]. \tag{2.3}$$

The second Hohenberg-Kohn Theorem states that the functional F_{HK} gives the value of the exact ground state energy of a system, only for the true density of the system, thereby establishing a minimum bound to the energy of the system calculated using the equation 2.3. The proof of this theorem follows the variational principle and is similar to the lower bounds for ground state energy established in the wavefunction regime of calculating ground state energies for the Schrödinger equation. While DFT is in principle, an exact theory, the degree to which the functional is approximated accurately determines how well it works in practice. The next section elaborates on the practical aspects of DFT.

2.2 Kohn Sham Density Functional Theory

Besides the $E_{\text{ncl}}[\rho(r)]$ term for incorporating the non-classical effects in electron-electron interaction, another term that is not directly available for practical computation using DFT is the kinetic energy term $T[\rho(r)]$. This is because in converting from wavefunctions to densities, the access to the kinetic energy contribution from the individual electron as an entity is lost. To account for this, Kohn-Sham DFT (KS-DFT) [67] introduces a novel formulation of the density to model the kinetic energy using an auxiliary system of non-interacting particle wavefunctions, similar to that in Hartree-Fock or other wavefunction-based methods. The density of the system now is defined as the sum of these densities of non-interacting particles as

$$\rho(r) = \sum_{i=1}^{N} f_i |\Psi_i(r)|^2. \tag{2.4}$$

Introducing this form of density introduces an error in the evaluation of kinetic energy (which arises due to the non-interacting particle assumption) which needs to be accounted for. This new density formulation leads to the governing equation of KS-DFT, in terms of the non-interacting Kohn-Sham orbitals as

$$E[\rho(r)] = \sum_{i=1}^{N} \langle \psi_i(r)| \left(-\frac{1}{2}\nabla^2 + v_{ne}(r) \right) |\psi_i(r)\rangle + \frac{1}{2}\int \frac{\rho(r)\rho(r')}{|r-r'|} dr' + E_{\text{xc}}[\rho(r)]. \quad (2.5)$$

The errors introduced by non-classical interaction and kinetic energy operators are now collected in a single term, famously called the exchange-correlation functional E_{xc}. This is the only term in the equation 2.5 without an explicit functional form, requiring the development of accurate approximations to evaluate the ground state properties of molecules. The exchange-correlation term includes all the errors and correlations that are unaccounted for in KS-DFT and therefore typically problematic to evaluate. Several approaches exist to approach the development of accurate exchange-correlation functionals, including experimental data, corroboration with wave-function approaches, or physical intuition [68]. The search for novel functionals has led to the development of the "Jacobs ladder" [69]. The local density approximation (LDA) [67] proposed in the original DFT paper itself forms the lowest step of this ladder and is often used only for initial qualitative explorations or when using more expensive functionals is computationally expensive. Several Generalised Gradient Approximation (GGA) functionals such as the popular PBE functional [70] introduce variations in density across the surface of molecules through the density gradient term, thereby improving on the LDA, forming the next rung of this ladder. Higher in the Jacobs ladder are hybrid functionals such as the B3LYP functional [71–74], which is one of the most popular functionals used for the evaluation of properties of organic molecules. The functional is particularly useful for the evaluation of energies of organic molecules and hence was employed in the calculation of spectra in both datasets in this thesis.

2.3 Time Dependent Density Functional Theory

KS-DFT formalism considers only time independent Hamiltonians, which while important for many ground state properties, is not an exact representation of time-dependent properties of molecules. The treatment of phenomena such as UV spectra, photochemistry, or optics using a density-based approach requires extending DFT for the time-dependent Schrödinger equation. This extension of the DFT to time dependent systems requires first establishing the uniqueness of a time-dependent density for the evaluation of the properties of such a system. For such a density $\rho(\mathbf{r}, t)$ which can be written as a sum over N particles of a system,

$$\rho(\mathbf{r}, t) = N \sum_{N} \int \cdots \int |\Psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N, t)|^2 \, d\mathbf{r}_2 \ldots d\mathbf{r}_N \quad (2.6)$$

the Runge-Gross theorem [64] establishes a unique mapping between a time dependent external potential v_{ext} and the time dependent density of a system. This theorem allows for the formulation of a calculation scheme similar to that in the time-independent formalism of DFT. The Schrödinger equation for the system then becomes

$$\frac{\partial}{\partial t}\Psi_i(r, t) = -\frac{\nabla^2}{2} + v_{KS}(r, t)\Psi_i(r, t), \quad (2.7)$$

where the time dependent Kohn-Sham potential, like the Kohn-Sham potential, is not known. It can be expressed as the sum of

$$v_{KS}(\mathbf{r},t) = v_{ext}(\mathbf{r},t) + v_{Hartree}(\mathbf{r},t) + v_{xc}(\mathbf{r},t) \tag{2.8}$$

The Hartree potential here accounts for the electrostatic interaction between the various electrons as

$$v_{\text{Hartree}}(\mathbf{r},t) = \int \frac{\rho(\mathbf{r}',t)}{|\mathbf{r}-\mathbf{r}'|}\,dr' \tag{2.9}$$

The exchange-correlation potential v_{xc} is composed of various many-body interaction effects and is approximated using methods similar to KS DFT.

Two branches of TDDFT have evolved for the practical calculation of properties from the time-dependent formulation of the Kohn Sham equation, namely real-time density functional theory (RT-TDDFT) and linear response time-dependent density functional theory (LR-TDDFT). For the calculation of spectra, efficient schemes have been developed using the latter and are implemented in popular quantum chemistry software packager. A formal treatment of this method will therefore be presented in the following paragraphs.

LR-TDDFT originates from the treatment of TDDFT using perturbation theory. Within this theory, time dependent properties are evaluated as a response of the ground state density to an external stimulus such as the time dependent external potential. Consider a system of interacting particles which at time $t = 0$ is in its ground state when a perturbation is switched on. Expressing this perturbation as a change to the otherwise time-independent potential of this system, one can write

$$v_{\text{ext}}(\mathbf{r},t) = v_{\text{ext},0}(\mathbf{r}) + \delta v_{\text{ext}}(\mathbf{r},t) \tag{2.10}$$

where $\delta v_{\text{ext}}(\mathbf{r},t) = 0$ for $t \leq 0$. The response of observables that react to this perturbation of the potential can be expressed in the form of a Taylor series expansion. Thus for the density of the interacting system, the corresponding response then becomes

$$\rho(\mathbf{r},t) = \rho_0(\mathbf{r}) + \rho_1(\mathbf{r},t) + \rho_2(\mathbf{r},t) + \dots \tag{2.11}$$

where the term $\rho_i(\mathbf{r},t)$ represents the i-th order perturbation as given by the Taylor series. As its name suggests LR-TDDFT considers only the first-order response to perturbations caused by external potential and hence orders greater than 1 in equation 2.11 are not considered for evaluating properties. The linear response function $\chi(\mathbf{r},t,\mathbf{r}^{'},t^{'})$ that is ultimately used to calculate the perturbation in density is given by

$$\chi(\mathbf{r},t,\mathbf{r}^{'},t^{'}) = \frac{\delta\rho(\mathbf{r},t)}{\delta v_{\text{ext}}(\mathbf{r}^{'},t^{'})} \tag{2.12}$$

with the perturbation in the density calculated as

$$\rho_1(\mathbf{r},t) = \int_0^{\infty} dt^{'} \int d\mathbf{r}\chi(\mathbf{r},t,\mathbf{r}^{'},t)\delta v_{\text{ext}}(\mathbf{r}^{'},t^{'}). \tag{2.13}$$

Fourier transforming the response function in the time domain with respect to $t - t^{'}$ to the frequency domain gives the time-independent form of the response function as

$$\chi(\mathbf{r},\mathbf{r}^{'},\omega) = \sum_I \frac{\langle\Psi_0|\hat{n}(\mathbf{r})|\Psi_I\rangle\langle\Psi_I|\hat{n}(\mathbf{r}^{'})|\Psi_0\rangle}{\omega - \Omega_I + i0^+} - \frac{\langle\Psi_0|\hat{n}(\mathbf{r}^{'})|\Psi_I\rangle\langle\Psi_I|\hat{n}(\mathbf{r})|\Psi_0\rangle}{\omega + \Omega_I + i0^+} \tag{2.14}$$

which involves a summation over I electronically excited states with energies given by $\omega_I = E_I - E_0$, where E_0 is the energy of the ground state of the system. When the terms in the denominator become exactly equal to zero, the response function is undefined which happens when the frequency ω is equal to the energy of an excited state. This particular feature of the response function in the frequency domain allows for the calculation of excited state energies while being a time-independent function. This formulation of the response function forms the basis of various mathematical schemes used to obtain excitation energies in LR-TDDFT.

The equation for the response function above is for a system of interacting particles, which is difficult to calculate. A non-interacting Kohn-Sham system is therefore used to carry out calculations leading to the response function, and thereby the perturbed densities. The non-interacting Kohn-Sham formalism allows for the conversion of Equation 2.14 into a matrix formalism known as Casida's equations [75, 76]. The proof of how one arrives at these equations is beyond the scope of this dissertation and we only refer to the main results of this formalism. The secular equation that results from the Casida formalism for the calculation of orbital energies in LR-TDDFT is given by

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^* & \mathbf{A}^* \end{bmatrix} \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix} = \omega \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix} \tag{2.15}$$

where the elements of the matrices **A** and **B** are given by

$$A_{ia,jb} = \delta_{ij}\delta_{ab}(\varepsilon_a - \varepsilon_i) + K_{ia,jb} \tag{2.16}$$

and

$$B_{ia,jb} = K_{ia,jb} \tag{2.17}$$

where **K** is the coupling matrix whose elements are given by

$$K_{ia,jb} = \langle \Psi_i(\mathbf{r}_1)\Psi_b(\mathbf{r}_2)| \frac{1}{|\mathbf{r} - \mathbf{r}'|} + v_{\mathrm{xc}}(\mathbf{r}_1, \mathbf{r}_2, \omega)|\Psi_a(\mathbf{r}_1)\Psi_j(\mathbf{r}_2)\rangle \tag{2.18}$$

The matrix elements of **X** and **Y** excitation and de-excitation matrices are given by

$$X_{ia,\sigma}(\omega) = -\sum_{\sigma'}\sum_{jk} \frac{f_{j\sigma} - f_{k\sigma}}{\omega - (\epsilon_j - \epsilon_k)} \langle \Psi_{i\sigma}(\mathbf{r})\Psi_{k\sigma'}(\mathbf{r}^{'})| \frac{1}{|\mathbf{r} - \mathbf{r}'|} + v_{xc}(\mathbf{r}, \mathbf{r}^{'}, \omega)|\Psi_a(\mathbf{r})\Psi_j(\mathbf{r}^{'})\rangle \times \frac{1}{\omega - (\varepsilon_i - \varepsilon_a)} \tag{2.19}$$

and

$$Y_{ia,\sigma}(\omega) = -X_{ai,\sigma}(\omega). \tag{2.20}$$

Here, the terms incorporate the spins of the electrons in these orbitals through the σ and $\sigma^{'}$ parameters respectively, while ε_i is the energy of the i-th orbital. The computation of these matrices can be further accelerated by assuming the elements of the coupling matrix **K** to be zero, in what is called the Tamm-Dancoff Approximation [77] which leads to the secular equation

$$\mathbf{AX} = \boldsymbol{\omega}\mathbf{X}. \tag{2.21}$$

The matrix $\boldsymbol{\omega}$ is the diagonal matrix containing excitation energies. The eigenvector matrix **X** can be calculated from the Kohn Sham orbitals obtained from the ground state density of the system. The TDA approximation along with LR-TDDFT was employed in calculations that required computations of excited states in this work. In addition to the excitation energies, TDDFT also allows for the calculation of oscillator strengths for the transitions between two orbitals. The equation for oscillator strengths associated with a transition between the ground state and an excited state whose wavefunctions are denoted by Ψ_0 and Ψ_i is presented below without proof in Equation 2.22

$$f_i = \frac{2}{3} \left|\langle\Psi_0|\hat{\mu}|\Psi_i\rangle\right|^2 \tag{2.22}$$

where $\hat{\mu}$ is the transition dipole moment between the two states. The excitation energies and oscillator strengths are then used to calculate line spectra for different excitation ranges. Oscillator strengths provide insights into the relative strengths of transitions between different states. Since experimental spectra are affected by effects leading to broadened spectra, techniques such as Gaussian broadening are used to convert line spectra into broadened spectra mimicking experimental spectra of molecules.

Chapter 3

Quantum Dynamics

The final project of this dissertation explored the solution of the Time Dependent Schrödinger Equation (TDSE) using a data-driven approach. Generating data for training ML models for this purpose requires knowledge of the numerical methods used to solve the TDSE. This chapter provides a brief introduction to the concepts and implementations of the methods in quantum dynamics used during the course of this dissertation.

3.1 The Time-Dependent Schrödinger Equation

The TISE $\hat{H}\Psi = E\Psi$ is useful for calculating the wavefunctions and energies for systems at rest but does not provide information on how a system evolves with time. For such systems, the propagation of a wavefunction is given by

$$i\hbar \frac{\partial \Psi(\mathbf{r}, t)}{\partial t} = \hat{H}\Psi(\mathbf{r}, t) \tag{3.1}$$

Given an initial wavepacket $\Psi(\mathbf{r}, t_0)$ at time t_0, modeling the evolution of such a system in time amounts to calculating wavepacket propagation using the TDSE. This involves propagating the wave packet at different time steps until a final desired time state $t = t_{\text{fin}}$ has been reached. The TDSE is, therefore, an initial state problem with boundary conditions, whose solution depends on the initial state and the Hamiltonian under which the system evolves in time, in contrast to the TISE.
The Hamiltonian $\hat{H}$ of the TDSE is the sum of the kinetic and potential energy operators $\hat{T}$ and $\hat{V}$ respectively, where the potential may or may not be independent of time. To obtain the solutions of the TDSE at various time steps, it is convenient to express the evaluation of a wavepacket at time t using a propagator approach. Given the initial wavepacket at t_0, the quantum propagator U is a function that maps the initial wavepacket to an array of wave packets $(\mathbf{r}, t_0), (\mathbf{r}, t_1)...(\mathbf{r}, t_{\text{fin}})$ at different time steps t_i. If the wavefunction at time t is expressed using U and $\Psi(\mathbf{r}, t_0)$ as

$$\Psi(\mathbf{r}, t) = U(t, t_0)\Psi(\mathbf{r}, t_0), \tag{3.2}$$

then for a time dependent Hamiltonian one arrives at the following equation of the propagator

$$\hat{U}(t + \Delta t, t) = \exp\left(-\frac{i}{\hbar}\int_t^{t+\Delta t} \hat{H}(t')\, dt'\right). \tag{3.3}$$

The TDSE is exactly solvable only for a few model systems. Numerical schemes have therefore been developed to implement wavepacket propagation for quantum dynamical systems. Of these, the Chebyshev Scheme [78], the second order difference Scheme (SOD) [79] and the split operator method [80] focus on efficient schemes that approximate the propagation operator. Wavepacket-based approaches such as the

Multi Configuration Time Dependent Hartree (MCTDH) [50–52] or variational Multi Configuration Gaussian (vMCG) [81], on the other hand, define a specific ansatz for the form of wavepackets. In these approaches, wavepackets are then propagated using the Hamiltonian in accordance with the variational principle. Doing so leads to equations of motion (EOMs) required for obtaining wavepacket coefficients. Special numerical integration methods are then used to further help solve these equations efficiently and compute the observables of the system. In this dissertation, one method of each type was used to generate wave propagation data.

3.2 The split operator method

The split operator method is a popular numerical technique that has been employed for solving different PDEs, including the TDSE. It takes advantage of the fact that the Hamiltonian operator can be decomposed into position and momentum components. In executing the split operator method, a wavepacket is advanced in time through a sequence of operations in its position and momentum spaces. To optimize the application of these operations, they are carried out in the Fourier spaces of their respective domains. This approach ensures the efficient and accurate simulation of the time evolution of quantum systems. By employing Fourier transforms in both position and momentum spaces, the split operator method provides a simple and efficient way to compute wavepacket propagation for quantum dynamical systems. Consider the Hamiltonian $\hat{H}$ of Eq. 3.1, which can be separated into momentum (p) and position (r) components as

$$\hat{H} = \hat{H}_p + \hat{H}_r \,, \tag{3.4}$$

where the kinetic energy operator and the potential energy operator are given by $\hat{H}_p = -\frac{1}{2}\sum_{i=1}^{f} \nabla_i^2$ and $\hat{H}_r = V(\mathbf{r})$ respectively. For an arbitrary time t, with an arbitrary time step dt, the propagator U from Equation 3.1 can be written as a product of the exponential function of the non-commuting operators $\hat{H}_p$ and $\hat{H}_r$ and further expanded using the Taylor series expansion and the Baker-Campbell-Hausdorff formula [82] as

$$e^{-i\hat{H}_p dt} e^{-i\hat{H}_r dt} = e^{-i(\hat{H}_p+\hat{H}_r)dt + \frac{1}{2}[-i\hat{H}_p dt, -i\hat{H}_r dt] + \dots}. \tag{3.5}$$

Further, with Strang splitting [80] the system's evolution can be divided into two distinct steps. The first step consists of a half-step in the position space, followed by a full step in the momentum space. The wavefunction $\Psi(\mathbf{r}, t+dt)$ thus propagated by the time-step dt in terms of the wavefunction $\Psi(\mathbf{r}, t)$ becomes

$$\Psi(\mathbf{r}, t+dt) = \left[e^{-i\hat{H}_r \frac{dt}{2}} e^{-i\hat{H}_p dt} e^{-i\hat{H}_r \frac{dt}{2}}\right] \Psi(\mathbf{r}, t) + \mathcal{O}(dt^3) \tag{3.6}$$

with an error of $\mathcal{O}(dt^3)$. Higher-order terms are neglected in the split-operator formulation. In practice, the transformation between position and momentum space is done using the Fourier transform of each space, which allows for efficient handling of the various kinetic and potential operations. Defining $\mathcal{F}$ and $\mathcal{F}^{-1}$ as the Fourier transform and its inverse, the wavefunction becomes

$$\Psi(\mathbf{r}, t+dt) = \left[e^{-i\hat{H}_r \frac{dt}{2}} \mathcal{F}^{-1}\left[e^{-i\hat{H}_p dt} \mathcal{F}\left[e^{-i\hat{H}_r \frac{dt}{2}} \Psi(\mathbf{r}, t)\right]\right]\right] + \mathcal{O}(dt^3). \tag{3.7}$$

The Fourier transform $\mathcal{F}$ and its inverse $\mathcal{F}^{-1}$ allow us to move from position space to momentum space and vice versa. This transformation simplifies the time evolution under the kinetic and potential energy operators. The Fourier transform of $\Psi(\mathbf{r},t)$ gives $\mathcal{F}[\Psi(\mathbf{r},t)]$ in momentum space, and the inverse Fourier transform of this result brings us back to position space. The global error term $\mathcal{O}(dt^3)$ is dependent on the grid size. As a result, for grids with large time steps the split operator method can be erroneous. Additionally, as simulation becomes lengthier, error accumulation becomes a significant issue, restricting the usage to simulations of small time lengths.

3.3 Multi Configuration Time Dependent Hartree Method

The split operator method is popular for many applications in PDE solving as it is easy to implement and gives fast and accurate solutions for small systems. Besides error accumulation for long periods, the implementation of the split operator method for problems in multiple dimensions is computationally intensive and sensitive to the choice of the time step for the problem under investigation. Such issues with numerical methods that approximate propagator approaches have led to the development of several algorithms that allow for the computation of quantum dynamical simulations for larger systems. One such method is MCTDH which allows for efficiently computing the dynamics of large quantum dynamical systems with multiple degrees of freedom (DOFs). The origin of the MCTDH method can be traced to the standard wavefunction method in quantum dynamics.

The standard method for solving the TDSE involves the expansion of the wavefunction Ψ into a product-basis form, and then solving for the individual equations of motion, thereby obtaining the coefficients for the basis functions. For a system with f DOFs, a wavefunction is typically written as

$$\Psi(q_1,\ldots,q_f,t) = \sum_{j_1=1}^{N_1}\cdots\sum_{j_f=1}^{N_f} C_{j_1,\ldots,j_f}(t)\chi_{j_1}(q_1)\cdots\chi_{j_f}(q_f). \tag{3.8}$$

The standard procedure for obtaining the coefficients C involves plugging the wavefunction into the TDSE 3.1 and then applying the Dirac-Frenkel variational principle

$$\langle\delta\Psi|\, H - i\partial_t \,|\Psi\rangle = 0 \tag{3.9}$$

which lends to the EOMs

$$i\dot{C}_{j_1,\ldots,j_f} = \sum_{\alpha_1,\ldots,\alpha_f} \chi_{j_1}\cdots\chi_{j_f}\langle H\rangle\chi_{\alpha_1}\cdots\chi_{\alpha_f}C_{\alpha_1,\ldots,\alpha_f}. \tag{3.10}$$

While the standard method was widely successful and easy to implement, it suffers from a multifold scaling in memory and computational time costs with an increase in the number of DOFs f. As an example, a system with six degrees of freedom and twenty basis functions requires 64×10^6 basis functions. While the calculations of coefficients for such large systems are doable with modern computing systems, such methods are not feasible for cases where $f>6$ and more approximate methods are used. The need for computationally efficient algorithms for solving the TDSE led to the development of the MCTDH method.

The basis of the MCTDH method is a novel ansatz that allows for splitting of the wavefunction $\Psi(\mathbf{q},t)$ of a system into a product basis of single-particle functions (SPFs) weighted by a time-dependent coefficient vector $A(t)$ to accommodate several possible configurations of a system. The ansatz for such a wavefunction is given by

$$\Psi(\mathbf{q},t) = \sum_{j_1}^{n_1} ... \sum_{j_f}^{n_f} A_{j_1...j_f}(t) \prod_{\kappa=1}^{f} \varphi_{j_\kappa}^{(\kappa)}(q_\kappa, t) \tag{3.11}$$

where $\mathbf{q} \equiv (q_1, q_2, \ldots, q_f)$ and n_κ represents the index of an SPF. The total number of possible configurations generated using this ansatz is given as $\prod_\kappa^f n^{(\kappa)}$, where f is the dimensionality of the quantum system. $A_{j_1...j_f}(t)$ is the time-dependent coefficient which describes the time evolution of the quantum system. The j_k-th configuration in κ-th DOF can be expressed in a time-independent basis set, unlike the previous ansatz of the standard method. Each SPF is thus further split as

$$\varphi_j^{(\kappa)}(q_\kappa, t) = \sum_{\mu=1}^{N_\kappa} c_{j,\mu}(t) \chi^{(\kappa)}(q_\kappa), \tag{3.12}$$

where $c_{j,\mu}(t)$ are the time-dependent coefficients that determine the contribution of the primitive $\chi_{i_\kappa}(q_\kappa)$ in the discrete variable representation (DVR) function. Using the standard procedure for obtaining coefficients as described above, we obtain the EOMs with respect to time-dependent coefficients A_J and the SPFs $\varphi_j^{(\kappa)}$. The EOM with respect to A_J can be written as

$$i\frac{\partial A_J}{\partial t} = \sum_L \langle \varphi_J | \, H \, | \varphi_L \rangle \, A_L. \tag{3.13}$$

For each $\varphi_j^{(\kappa)}$, is corresponding EOM can be written as

$$i\dot{\varphi}_j^{(\kappa)} = (1 - P^{(\kappa)}) \sum_{k,l=1}^{n_\kappa} (\boldsymbol{\rho}^{(\kappa)^{-1}})_{jl} \langle H \rangle_{lk}^{(\kappa)} \varphi_k^{(\kappa)} \tag{3.14}$$

where $\boldsymbol{\rho}_{jl}^{(\kappa)}$ is the density matrix that is equal to $\left\langle \Psi_j^{(\kappa)} \middle| \Psi_l^{(\kappa)} \right\rangle$ while $P^{(\kappa)}$ is referred to as the MCTDH projector $P^{(\kappa)} = \sum_{j=1}^{n_\kappa} \left| \varphi_j^{(\kappa)} \right\rangle \left\langle \varphi_j^{(\kappa)} \right|$. This set of non-linear coupled differential equations (3.13, 3.14) in MCTDH is solved using a constant mean-field (CMF) approach which numerically integrates the coefficients of the basis functions in time at each step of the wavefunction. In combination with the DVR function for representation, this mean-field integration approach reduces the computational effort for evaluating coefficients thereby making the MCTDH a viable option even for systems with DOFs greater than ten, which is often challenging using other methods.

Chapter 4

Graph Neural Networks

Traditionally, the methods of property prediction in chemistry have either been motivated by physical or mathematical intuition. As an alternative to these techniques, ML methods aim to unravel patterns within chemical data itself and use this information for downstream tasks such as prediction, generation, or classification. Using the example of predicting spectra, this chapter provides a brief overview of the requirements for setting up an ML framework for chemical property prediction. Since the Graph Neural Network (GNN) architecture is employed for predicting spectra in this dissertation an overview of the theory and implementations of various GNN architectures is provided. An introductory overview of the theory of neural networks and an introduction to ML are not provided here, but the interested reader is referred to texts by Murphy [83] or Goodfellow, Bengio, and Courville [84].

4.1 Implementing a Machine Learning model

ML models can be categorized into three main approaches. Supervised ML is typically used on labeled data, where the objective is to learn the mapping between data labeled as input and its corresponding output. Unsupervised learning, on the other hand, works with unlabeled data, where the aim is to understand patterns between the data itself. Finally, the third category of ML models is reinforcement learning which involves training an agent to make decisions by interacting with an environment and receiving feedback in the form of rewards and penalties. The first step in an ML application is to have a clearly defined task for which an ML framework is needed. Having defined such a task, the successful implementation of an ML algorithm depends on four steps: 1. The quality of the dataset, 2. An appropriate input representation, 3. Choosing the right architecture and finally, 4. The right technique for optimizing the parameters of the ML framework to get the most accurate result. Each of these steps is described for a task in supervised ML, which forms the basis of all ML tasks in this dissertation.

4.1.1 Datasets

The first and most important requirement to execute an ML model to predict any property is to have a dataset from which it can learn input-output relationships. A typical dataset is a collection of input-output tuples in the form of pairs of molecules and their properties. In chemistry, datasets can be generated experimentally or theoretically. Theoretically generated data for molecular properties are easier to compute to the desired level of accuracy using methods of quantum chemistry than gathering vast amounts of experimental data. Complex ML models require large datasets for training them, which makes simulated datasets even more attractive, and a large number of ML applications in chemistry use datasets created using quantum chemical simulations of properties. One such popular dataset is the Quantum Machine 9 (QM9) dataset [85] which contains 134k molecules for which properties have been obtained from DFT calculations. While this dataset catalogs several molecular properties such as the HOMO-LUMO gap, polarizability, various formation enthalpies, and energies, it does not contain calculations of spectra. However, given its proven usability for a wide range of ML applications and the large number of diverse molecules within it, the molecules in this dataset provide a good starting point for generating spectroscopy data. The datasets in this dissertation use molecules from the QM9 dataset and spectra computed for these molecules using methods of DFT and TDDFT.

4.1.2 Input representation

Chemists typically identify and distinguish molecules through their IUPAC name, molecular formula, and relevant physio-chemical properties or a combination of them. It is also common to depict molecules using their two-dimensional structure as an image. In quantum chemistry, molecules are often input in a representation that combines atomic symbols and atom coordinates representing the 3D orientation of a molecule. Such representations, while useful for human understanding, cannot be used for ML methods. The way a molecule is represented as an input determines the ML architecture and the quality of predictions made by the ML model. Converting molecular information such as atoms, their positions, and their connectivity in a molecule (or any input for an ML model) into representations for ML models is termed feature engineering [86] and it is an essential part of problem formulation in ML for chemical property prediction.

A rather simple representation of molecular data for ML applications is to employ a vector containing the relevant molecular properties as input for each molecule in the input data. This type of representation allows for flexibly incorporating and removing structural and chemical properties depending on the ML application. Another format used extensively in cheminformatics, and now popular in many ML applications for chemical property prediction is the text-based SMILES representation [87]. A SMILES representation depicts molecules as a string of alphabets and symbols that denote atoms and their connectivity in a molecule. Coulomb matrix [88] representations of molecules, that leverage electronic structural information and convert them into a matrix form similar to images, have also been successfully used in ML models for chemical property prediction. Smooth Overlap of Atomic Positions (SOAP) [89] representations use radial distribution functions to generate unique atomic and molecular environments for ML applications. Another representation that has recently gained traction in molecular ML is the molecular graph representation where molecules are represented as mathematical graphs [90, 91]. A general rule of thumb for designing molecular representations is invariance to translation, rotation, and permutations of atoms [92].

4.1.3 Architectures

Depending on the input representation and output property, different ML architectures can be used to learn relationships between molecules and their properties. Feedforward Neural Networks [93], Convolutional Neural Networks [94], Long Short Term Memory Networks [95], Graph Neural Networks [90, 91], and Recurrent Neural Networks [96] are all different neural network architectures that have been used for predicting different chemical properties in ML applications. Additionally, Kernel Ridge regression [97] and Gaussian process regression [98] are Bayesian ML methods promising in chemical property prediction applications. A major part of this thesis focuses on GNN architectures for predicting relationships between molecules and their spectra.

4.1.4 Hyperparameter optimization

Training an ML model involves optimizing the parameters that make up the model based on the data. Rather than using the entire dataset for training, the dataset is typically divided into three parts: training, test and validation. The training dataset is the largest part of the data, and the model parameters are optimized with respect to the samples within this subset. The test dataset is used to measure the accuracy of the trained model, while the validation set is used to ensure proper model training. For neural networks, training using the training dataset to determine weights (learnable parameters) of neurons in the neural network such that the output is as close as possible to the output data. (This is done, of course, with the caveat that the neural network does not 'memorize' the output, a phenomenon called overfitting, which is monitored by the model's performance on the validation set). Hyperparameter optimization [99] is the process of fine-tuning various parameters that guide a model's learning process. This is done by minimizing a loss function that calculates the difference between the output of the data instance from the training data and the output of the neural network for the same instance. Another key aspect is defining the hyperparameters to tune, which may include learning rates, batch sizes, the number of hidden layers, and more. Upon defining these parameters, this process also involves the selection of algorithms for choosing different parameters, such as grid search [100], random search [101] or Bayesian optimization [102]. Additionally, determining the appropriate evaluation metrics and strategies for cross-validation is vital to assess the model's performance accurately. Some parameters are common to various neural network architectures, but each neural network also has a set of hyperparameters specific to each architecture. Hyperparameter optimization thus helps find the best ML model from a plethora of variables given a training dataset.

4.2 Graph Neural Networks

4.2.1 Molecules as graphs

Graph representations of molecules convert molecular structures into mathematical graphs, where nodes represent atoms in the molecule and edges represent bonds between atoms. Such a graph can convey information about the connectivity of different atoms in molecules, but it is often not enough to distinguish different structures that have similar skeletal structures or conformers of the same molecule. Throught the addition of structured labels containing different atom, bond, and molecular properties in a vector form (feature vectors) to nodes/edges/graphs respectively, the information contained in graph structures can be enhanced and these graph structures can then be employed for ML purposes. Formally, a molecular graph is a data structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes, each representing an atom, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges, each representing a bond between atoms. Edges in a graph can be defined either through pre-existing knowledge of the bonds or by connecting atoms that lie within a certain cutoff physical distance from each other. The properties of individual atoms (nodes) $v \in \mathcal{V}$, such as atomic number or number of hydrogens attached to an atom in a molecule, are modeled as an f-dimensional node feature vector $\mathbf{x}_v \in \mathbb{R}^f$. The properties of an edge connecting nodes v and w, $v \times w \in \mathcal{E}$, are modeled as a d-dimensional edge feature vector $\mathbf{e}_{vw}$ in $\mathbb{R}^d$. An adjacency matrix A is often used to represent the connectivity between different nodes of a graph, where $A_{vw} = 1$ if there exists an edge between nodes v and w, and $A_{vw} = 0$ otherwise.

There are two types of features in any form of data: numerical and categorical. Numerical features can directly be used as elements of a feature vector. Categorical features such as type of atom, aromaticity, or hybridization are first converted into a one-hot encoding vector. A one-hot encoding vector represents a data value in terms of a binary vector, where only one bit is set to a value of one to indicate the presence of a category, while all the others are set to zero. Feature vectors of numerical and categorical data are then concatenated to form an input vector for each node in the molecular graph. This can also be extended to edge feature vectors for bonds and graph feature vectors for representing molecular information. Graph representations of molecules provide information about the connectivity between different atoms and bonds while being translationally and rotationally invariant. Feature vectors allow molecule-specific information to be incorporated into these graphs through feature engineering.

4.2.2 Graph Neural Network architectures

ML architectures have usually been used with structured input data such as vectors of fixed lengths, or images represented as matrices (tensors). Unlike these aforementioned data structures, graphs are unstructured with varying relationships between nodes and atoms. To properly learn the representations for such data architectures different from the conventional ML frameworks, are required, which has led to the development of GNNs for unstructured graph data such as molecules. The underlying principle that forms the basis of most modern GNN architectures is the message-passing algorithm [91]. A GNN takes a graph along with its node/edge/graph feature information as inputs and then iteratively transforms it to a novel representation for each of these features, while preserving the connectivity of the original graph. This process results in a graph that has updated information stored in its node (and edge) features through this message-passing process. This leads to the generation of unique node and edge vectors that incorporate information about the neighborhood of nodes and edges in the feature vectors of the graph. This newly transformed graph (or its nodes/edges) is then used for downstream applications, such as graph property prediction (like predicting the dipole of a molecule), edge prediction (predicting if a bond exists), or predicting a node type (identifying the atom of a molecule).

A pre-requisite for the implementation of a message-passing algorithm is the message function, which defines how the information of each node, edge, and/or graph feature vectors in the neighborhood of a node/edge is transformed before aggregation. It is common practice to transform feature vectors to higher dimensions using a linear neural network layer, transforming them from vectors x_v into embeddings. This transformation helps increase the expressivity of the neural network architecture, improving its ability to learn patterns from input data. Formally a message is defined as

$$\mathbf{m}_v^{l+1} = \mathbf{AGGREGATE}^{l+1}\left(\left\{(\mathbf{x}_w^l, \mathbf{x}_v^l, \mathbf{e}_{v,w}^l)\right\}_{w \in \mathcal{N}(v)}\right) \tag{4.1}$$

where (..) denotes a multiset containing the nodes and edges in the neighborhood $\mathcal{N}(v) \subseteq \mathcal{V}$ of a node v in the graph $\mathcal{G}$. Message passing is iteratively performed in cycles, and l indicates the number of iterations of message passing that have been completed. The AGGREGATE function depends on the type of GNN architecture. A key requirement for an AGGREGATE function is that the information aggregation operation must be invariant with the order(of nodes or edges) in which aggregation is performed. A simple example of aggregation could be adding all the node feature vectors surrounding a target node. Once messages are obtained for every node in the graph, node embeddings can be updated from x^l to x^{l+1} through an UPDATE function as

$$\mathbf{x}_v^{l+1} = \mathbf{UPDATE}^{l+1}(\mathbf{x}_v^l, \mathbf{m}_v^{l+1}). \tag{4.2}$$

The aggregate and update operations are applied at each node/edge individually for each cycle. Once the updated embeddings are completed for a predefined number of cycles, a final pooling of the information may be performed depending on the final task requirements. This pooling step is often termed READOUT formally defined as

$$\mathbf{x}_g = \mathbf{READOUT}(\mathbf{x}_v^L \mid v \in G). \tag{4.3}$$

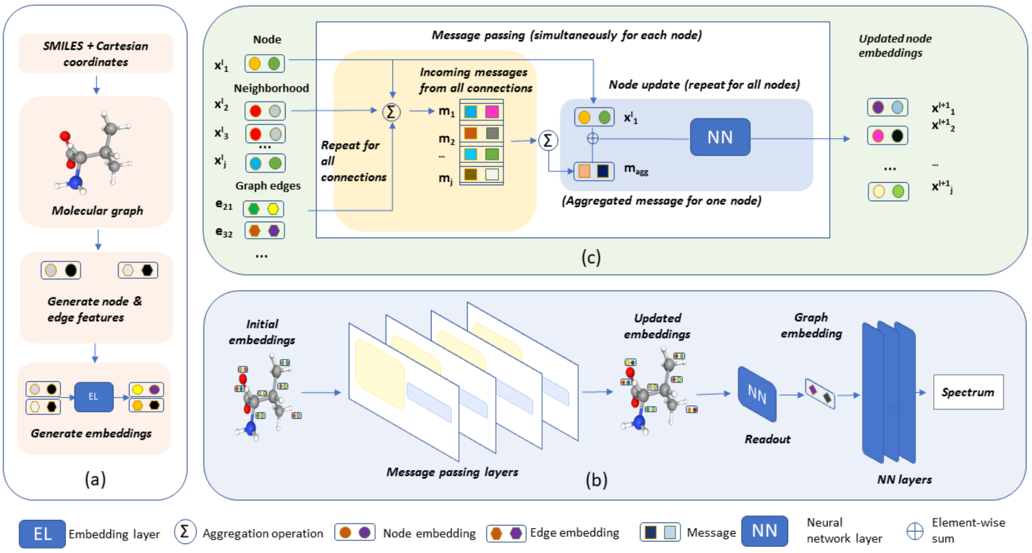


FIGURE 4.1: *Graph Neural Networks for the prediction of molecular spectra. Panel (a) depicts the transformation of a molecular SMILES to a graph with atom/bond features. Message passing to update graph embeddings (c). The complete GNN architecture that predicts spectra for a graph (b). Reprinted with permission from Singh et al. [103]. Copyright 2023 American Chemical Society.*

Such a final pooling can be a simple order invariant operation such as the sum or average of node embeddings of the different atoms in a graph. The final representation $\mathbf{x}_g$ is then used as an input to a neural network layer (there can be one or more such layers) that predicts the output property such as a molecular spectrum. Figure 4.1 illustrates the various steps in using a GNN for spectrum prediction. Panel (a) of this figure depicts the creation of a mathematical graph with atom and bond features from a SMILES structure. This step is performed using the RDKit [104] package. Panel (b) provides an overview of the GNN architecture that predicts a spectrum given a molecular graph. This includes the message-passing layers that give updated embeddings of the nodes in the graph, which are then aggregated into a graph embedding. The graph embedding is then used with one or more neural network layers to provide the final spectrum. Panel (c) shows the working of the message-passing algorithm.GNN architectures can be differentiated by the various types of functions defined at each step of the message-passing process. There is no consensus on which message-passing function works best for both node-level and graph-level tasks and different functions are usually experimented with before determining what works best for the ML task at hand.

4.3 Quantifying spectrum prediction error

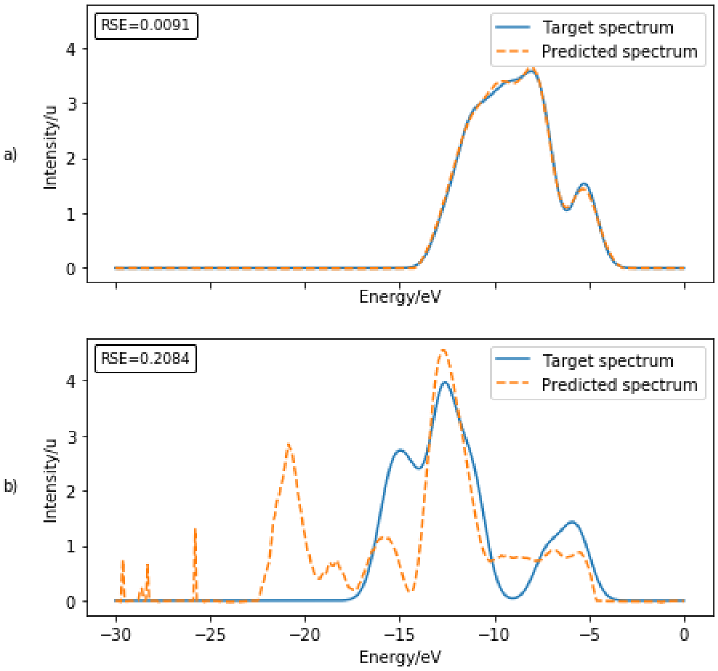


FIGURE 4.2: *RSE as a measure of quality of spectrum prediction. (a) An accurately predicted spectrum depicts a high level of agreement between the target and predicted spectra and has a low RSE value. (b) A high value of RSE implies the predicted spectrum has peaks that have mismatched intensities and as seen in the figure, some peaks that should not exist. Reprinted with permission from Singh et al. [103]. Copyright 2023 American Chemical Society.*

GNN architectures require defining outputs of fixed sizes, and therefore molecular spectra need to be standardized into fixed-length vectors for training and inference. To achieve this the energy range for the spectra is fixed and discretized into a fixed number of points typically into 0.1 eV between two consecutive points. The spectrum is then the value of the intensity at each of these energy values given by a fixed-length vector.

Since predicting spectra is a regression problem, we use the Mean Squared Error (MSE) loss function to train our models. In addition to minimizing training loss, the quality of predictions of spectra is reported using the Relative Spectral Error (RSE) metric [29], as defined in Equation 4.4. Let y^{tar} and y^{pred} denote the vectors for the target and the predicted intensities of the spectrum at an energy E. For a discrete spectrum with N points defined for the energy range $E_{max} - E_{min}$, the distance between two points is defined as ΔE, the unit distance between two points given by $\Delta E = \frac{E_{max} - E_{min}}{N}$ thereby obtaining the RSE as

$$RSE = \frac{\sqrt{\sum_i^N (y_i^{tar} - y_i^{pred})^2 \cdot \Delta E}}{\sum_i^N y_i^{tar} \cdot \Delta E}. \tag{4.4}$$

The RSE is thus a normalized form of the MSE, which is useful for interpreting the quality of spectra predictions. The RSE can provide a uniform metric to account for the differences in intensities of peaks across different molecules. An RSE value close to zero indicates a high-quality prediction and is desired for ML models. This is demonstrated through Figure 4.2 where (a) demonstrates a high-quality prediction with a low RSE value while in (b) a high RSE value arises out of a predicted spectrum that does not match effectively with the target spectrum.

Chapter 5

Explainable AI methods

Special techniques are required to understand the decision-making process of complex ML models, often termed black box models. Explainable Artificial Intelligence (XAI) methods focus on the development of such techniques. This chapter provides a brief description of the XAI method that was employed in Pub2. The content of this chapter follows some parts of the book on interpretability in ML by Molnar [105]. The interested reader is also referred to the book by Samek et al. [106] and the review article by Gunning et al. [107] for an overview of the state of XAI.

5.1 Attribution methods in XAI

A simple model whose predictions can be considered interpretable is a linear regression model. For such a model, it is easy to understand the contribution of each variable (or feature) of an input sample in the final prediction. The technique of assessing the contribution of different input features in a prediction is called feature attribution. It is a straightforward XAI technique that can be used to impart explainability to models that are otherwise not interpretable. Since feature attributions are used only after a model has been trained, they are termed *post-hoc* explainability methods. Additionally, feature attributions apply to individual data points, implying that these are locally interpretable methods. Despite the relatively recent rise of GNN models, feature attribution methods that explain graph model predictions have already borrowed from the techniques developed for interpreting ML models elsewhere. Popular feature attribution methods such as GradInput (GI) [108], class activation map (CAM) [109] and gradient class activation map (GradCAM) [110] have been shown to successfully explain predictions made by GNNs for molecular structure-property prediction models [42–45].

The CAM method is the simplest of these attribution methods in XAI and was employed in this dissertation to explain the importance of different molecular features for GNNs predicting spectra. Computing CAM attributions for a molecular graph involves calculating the node weights v_i for highlighting the contribution of various graph nodes to a spectrum prediction. As discussed in Chapter 4, GNNs use a global aggregation layer or a readout layer, before the output layer that predicts the final output value. The number of neurons in this layer depends on the length of the output vector. To obtain a CAM for such a GNN model, one assumes that the activation of each neuron can be treated as an independent class for which the model must make accurate predictions to get the correct peak value. Using this assumption one can obtain CAM weights for each of the different classes needed in a spectrum. CAM weights of each node (for each peak/class) can be depicted visually as a heatmap on the molecular graph. CAM operates on the aggregation layer before this final layer and obtains attributions for these different 'classes', giving an insight into atomic contributions at each point in the spectrum.

To compute CAM weights of a node for each class, let $F_{k(i)}$ be the activation of a neuron k in the GNN layer preceding the output layer. The CAM score $v_{c(i)}$ at node i of a molecular graph at a node for a class c is then defined [111, 112] as

$$v_{c(i)} = \sum_k \omega_k^c F_{k(i)}, \tag{5.1}$$

where ω_k^c denotes the weight of unit k for class c. Using this formulation, one can obtain CAM scores for each point in the spectrum of a given input molecular graph. CAM scores can also be aggregated for regions of a spectrum, giving an estimate of the most important nodes in that region according to the GNN model. The aggregated CAM visualization for a spectrum can then be used to analyze the degree to which an ML model attributes the correct contributing atoms for each peak in the predicted XAS spectrum. An example of a CAM obtained at one of the values on a spectrum predicted by a GNN model is given in Figure 5.1. The CAM plots in this figure illustrate the different CAM plots at different points and a peak in a spectrum.

5.2 Developing spectroscopic ground truths

CAMs provide both a visual and a quantitative measure of the activation of different atoms in a spectrum prediction. A heatmap from a CAM evaluation can be useful however only if one can figure out if an atom/region of the heatmap is correctly or incorrectly attributed for its contribution to the final prediction. For some property prediction tasks, this is fairly straightforward, as the ground truth to which the CAM is compared can be easily obtained using human expertise. Ground truth can be easy to obtain also when the ML task is elementary, such as identifying whether a molecule has a benzene ring. However, for many chemical property prediction tasks, such a ground truth benchmark is not easily obtainable, as in the case of prediction of XAS spectra.

The XAS of a molecule takes into account the various chemical environments surrounding the atoms that are being investigated. Since TDDFT-based approaches involve calculating excitations from individual orbitals, these can be used to obtain ground truth data for X-ray spectra systematically. Each excitation in an XAS originates from a core orbital and terminates at a final orbital, often delocalized over several atoms (Introduced in Sec 1.1.). Furthermore, excitations from the same atom can end up in different final states. Thus, for a particular peak region, there can be several contributing transitions originating from the same or similar environments. Figure 5.1 shows a visualization of this phenomenon, where excitations for a peak around 276 eV for a molecule are depicted. As seen in the figure, two similar line spectra around 276 eV originate from the same core atom. Since TDDFT spectra are peak-spectra (vertical lines in the plot, denoting individual excitations) that are broadened to mimic experimental spectra, to obtain a ground truth at any given point, one needs to incorporate information from different spectral lines underlying under a broadened peak.

To obtain a ground truth for an XAS spectrum one must first identify the peaks in the spectrum. For each peak then, the excitations that fall within the peak are arranged according to their oscillator strengths, and the five dominating contributing excitations in each region are selected. Each excitation is composed of transitions denoting the originating molecular orbital and the final orbital. Once contributing peaks are

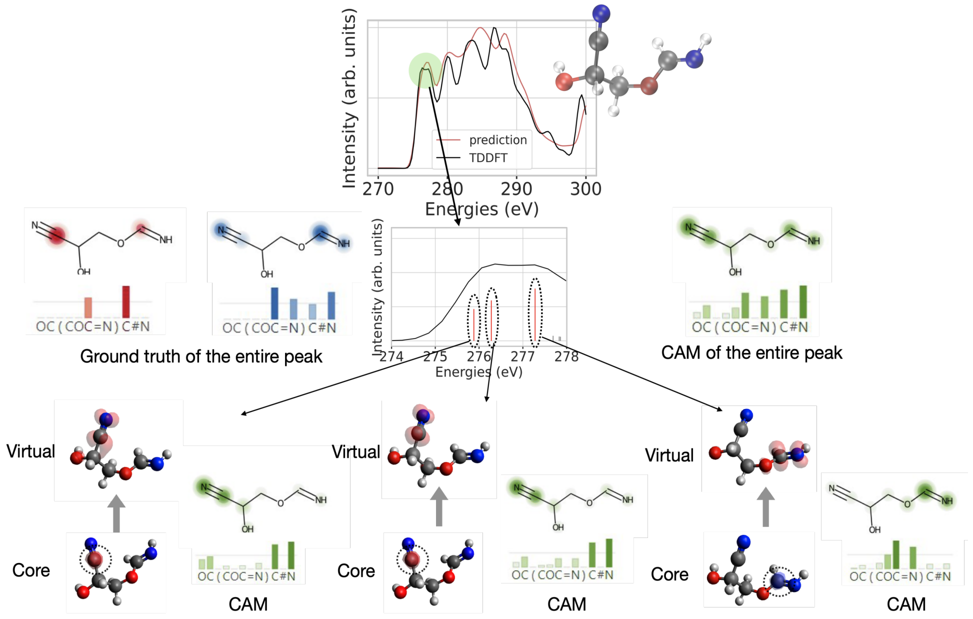


FIGURE 5.1: *Atom heatmaps depicting the ground truth evaluation and CAM for a GNN model for the XAS prediction of a molecule. Each vertical peak (central panel) is composed of different transitions, which all contribute to the common ground truth (top left). CAM plots of each peak, provide access to the atoms that the ML attributes most importance to (top right). Reprinted with permission from Kotobi et al. [113].*

obtained, these transitions and orbitals provide information on the atoms that contribute to the transition process. A contributing atom is assigned a score of 1, while a non-contributing atom gets a score of 0 for each transition. Finally using a weighted sum of the transition strengths of each peak we obtain a ground truth value for each atom in the molecule. Atoms that are a part of the orbitals where electronic transitions arise are labeled as 'core' atoms, while those that make up the virtual orbitals are labeled as 'virtual' atoms. These different contributions give a 'core' heatmap for the origins and 'virtual' heatmaps for the final states at each peak in the spectrum of a molecule. Each vertical line spectrum thus has its ground truth heatmap, which can be aggregated with other maps under the same broadened peak. Fig 5.1 depicts the ground truth core and virtual orbitals that contribute to transitions for the line spectra in the 276 eV region. The contributions of each of these line spectra are then visualized in the final ground truth for the peak, shown in the top left of the figure. This core ground truth shows the contributions from the carbon atoms attached to the two nitrogen atoms, while the excitations show the atoms in the virtual states that are composed of orbitals where electrons transition to. The ground truth creation approach in this work is based on quantum chemical principles and can be used for any software implementation of spectroscopy that provides access to the relevant quantities. Further, it can be fully automated and does not require any human intervention to label ground truth for the theoretical spectra of any kind.

5.3 Understanding spectroscopic predictions

Once a ground truth for the predictions has been calculated, the CAM obtained for each molecule is compared to this ground truth map, for each peak in the spectrum.

To do this, the Area Under the Curve (AUC) metric of the Receiver Operating Characteristic (ROC) [114] is employed. The AUC metric, which ranges from 0 to 1, has been extensively used to quantify the capacity of ML models to distinguish between different classes in a dataset. An AUC of 1.0 implies a perfect agreement between the CAM of a prediction and its ground truth, indicating that the model correctly attributes high activation values to the atoms from which excitation occurs (in the case of the core ground truth). On the other hand, a value of 0.5 can be achieved by a random classifier, which randomly assigns labels to the atoms showing activation values on a molecular graph. An average AUC of 0.7 or higher is generally considered indicative of a model that can correctly distinguish true and false label assignments for most of the data in a test set.

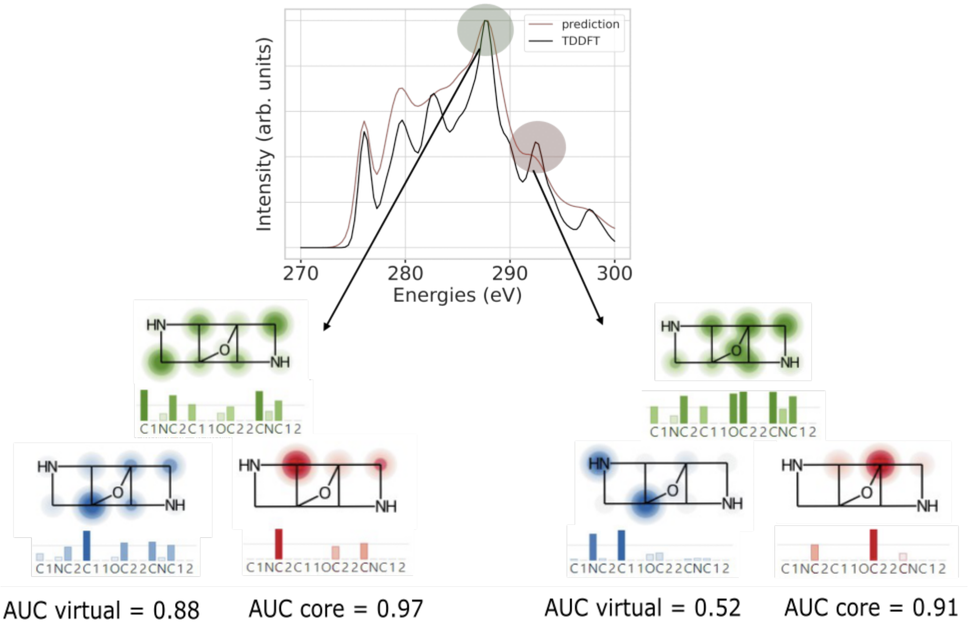


FIGURE 5.2: *AUC as an indicator of prediction quality. For the peak that is accurately depicted (around 288 eV) the AUC values are higher than the AUC values for the peak (around 292 eV), which is not reproduced as accurately. Reprinted with permission from Kotobi et al. [113].*

Figure 5.2 depicts how AUC values in the same molecular spectrum indicate differences in attributions, as well as reflect differences in prediction quality. For the peak at 288 eV, the core AUC score values are higher, indicating that while some atoms are not given the same importance in the CAM as in the ground truth, there are no incorrect attributions. For the peak at 293 eV however, the virtual AUC score is very low, due to the CAM not accounting for the transitions that end up in orbitals localized on orbitals which include amongst others, this nitrogen atom. This possibly is a reason for the relatively poor quality. The different AUC values for peaks in a spectrum can be averaged for the entire spectrum of a molecule, giving a value for an entire molecule. Similarly, the average AUC value for all molecules in the test set serves as an evaluation tool for the overall degree of interpretability of predictions made by different GNN models.

Chapter 6

Fourier Neural Operators for Quantum Dynamics

The final part of this dissertation explores the use of a data-driven approach for simulating wavepacket propagation in quantum dynamical systems. This requires the use of an ML framework suitable for solving the TDSE, which is a PDE in space and time. The Fourier Neural Operator (FNO) framework is one such approach that learns PDEs from data. This chapter provides an introduction to data-driven methods for PDE solving before discussing in detail the FNO framework and its implementation for simulating wavepacket propagation.

6.1 ML methods for solving partial differential equations

PDEs have been used to model physical phenomena in fluid dynamics, quantum chemistry, and electrodynamics for more than two centuries. The last century has been a testament to how scientific computing has helped accelerate numerical methods that are used to solve PDEs. However, several scientific phenomena are modeled using PDEs that are non-linear and high-dimensional. This makes solving and analyzing them using these computational methods expensive and often impossible. ML methods provide a novel way to analyze PDEs from data generated using experiments and simulations. While using ML as an approach for solving PDEs has been an area of research going back three decades [115], recent advances in ML have led to the development of algorithms that have helped accelerate these approaches. Data-driven approaches for PDE solving (and analysis) can be categorized into four approaches. The first approach uses ML techniques to enable the data-driven discovery of PDEs using data obtained from simulations and experiments. Most ML models, while adept at identifying patterns between input and output data, are not interpretable (in general) which limits their understanding, especially in modeling physical phenomena. Symbolic regression techniques, on the other hand, are interpretable ML techniques that can learn phenomena from data. Advanced symbolic regression applied to the task of PDE learning from data has helped in the discovery of PDEs for several non-linear phenomena [116, 117]. Coordinate transformations and dimensionality reduction are two common techniques that have been employed to reduce complex PDEs into simpler forms that are less complicated due to their lower dimensionality or analytic simplicity. The second approach of data-driven methods in PDE aims to bring about these transformations in an intelligent manner, thereby providing more stable and efficient reductions of PDEs describing physical phenomena [118, 119].

The third and fourth data-driven approaches are related to numerical methods of solving PDEs. Some of these approaches aim to improve the computational efficiency, accuracy, and scaling of the existing numerical methods in PDE solving. A deep learning approach, has for example, been used to accelerate and improve the computation of spatial derivatives on grids used to solve PDEs [120, 121]. Alternatively, interpolation methods based on ML have been used to accurately simulate phenomena on finer grids than is possible by conventional PDE solvers [122]. The final data-driven approach in PDE solving concerns itself with learning PDE solutions from data using ML methods. Within this category, there are two classes of methods, one of which uses neural networks to approximate solution functions and use physical constraints to improve the accuracy of neural network predictions [122, 123]. These methods require prior knowledge of the PDE under investigation but are also often limited to one set of parameters of the PDE. This means that simulating a new instance of PDE parameters requires retraining the neural network to model the new PDE. The operator learning approach [56, 124, 125], which is used in this work is the second class in this category and aims to learn PDEs purely from data. As a result, these approaches do not require any prior information on a PDE to model it from data. Additionally, these approaches are not limited to a single instance of PDE parameters and can be used to model several such instances in a single model. The next section describes this operator approach.

6.2 Fourier Neural Operators

Neural operator approaches [56, 124, 125] utilize the neural operator approximation theorem [126] to model relationships between data that originate from dynamical simulations or experiments. This theorem, which is a lesser-known counterpart of the popular universal approximation theorem [127] states that a neural network with a single layer can be used to approximate an operator that models the relationship between two function spaces. Mathematically, the neural operator $\mathbf{G}_\theta$ attempts to learn the relationship between inputs $\mathbf{I}$ and outputs $\mathbf{U}$ as

$$\mathbf{U} = \mathbf{G}_\theta(\mathbf{I}), \tag{6.1}$$

where θ denotes the neural network weights learned by training on PDE data. Each instance of the input data of a PDE is typically defined by a unique set of initial conditions and these give rise to different outputs. The operator aims to learn over several instances of such inputs, parameters of the PDE, and their outputs. This learning does not require an explicit form of the PDE but happens purely from data. Some physics-inspired neural operator approaches have been developed [128] that use governing equations and physical constraints relevant to a PDE for improving model training. However, the investigation in this project focuses mainly on the traditional FNO without incorporating any physics-inspired constraints.

Figure 6.1 depicts how the FNO architecture is implemented for the task of learning wavepacket propagation. Most neural operator approaches (of which FNO is one) share this same basic architecture. The input $\mathbf{I}$ for the FNO architecture is created by concatenating the wavepacket density, potential, and grid information into a tensor object (Panel (a)). The FNO model consists of three basic components. The input $\mathbf{I}$ is first transformed (or lifted into) a higher dimension using a linear neural layer. This layer is denoted as the lifting layer in Figure 6.1 (b). Transformation of inputs to higher dimensions allows the capturing of patterns and relationships in the input data. Such transforming layers are commonly used in neural network approaches in ML (it was also used in the GNN implementations for spectra, as an embedding layer). The hidden layers of the neural operator also serve this purpose, enabling the capture of relationships on a local as well as global level. In the case of wavepacket propagation, a global change is the movement of an entire wavepacket from one part of the grid to the other, while local information is the change that the wavepacket undergoes in between the initial and final states. The different operations in a neural operator framework are designed to learn these patterns from data using transformation operations that aggregate local/global information from different regions of the input tensor. In the case of dynamics induced due to PDEs, such information is not as easy to aggregate and therefore special layers are introduced for efficient learning in the hidden layers following the linear layer.

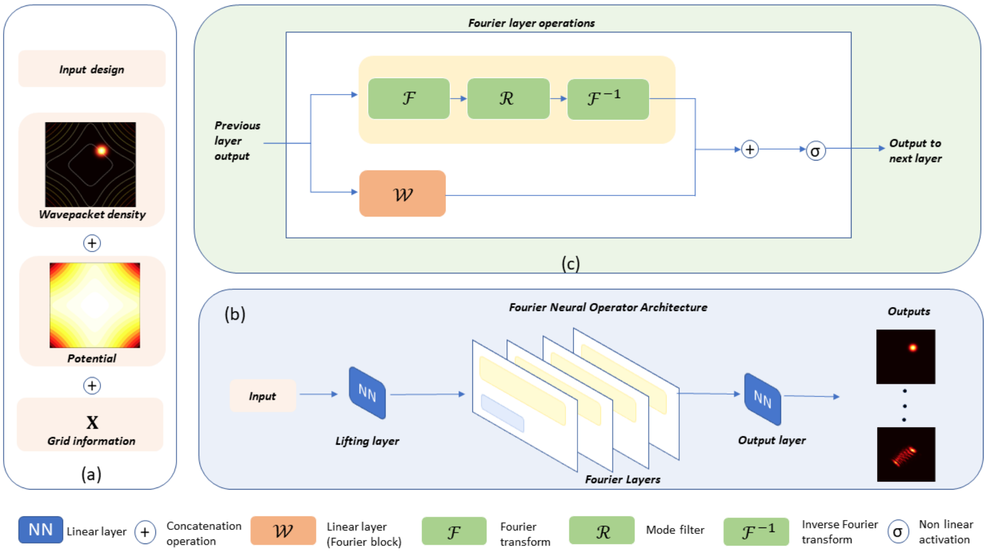


FIGURE 6.1: *Learning wavepacket propagation using the FNO ML model. Each input is a concatenation of the initial wavepacket density, along with the potential and grid information (a). An overview of the FNO architecture is depicted in (b). Fourier layers (and in general operator layers) and their operation as depicted in (c). Given an input (a) the FNO model is trained to predict the entire simulation in the output. Reprinted from the manuscript submitted at J.Chem.Comput.*

This learning of various features in any neural operator framework is enabled by the use of neural operator layers that follow the lifting layer. These layers are depicted as the Fourier layers in Panel (b) of Figure 6.1. The proof of how one arrives from a neural operator layer to a Fourier layer is beyond the scope of this dissertation. In the following lines, this transformation is presented. Operator approaches use kernel integration formalism to update the representations that form outputs of a neural layer. Such an output $a^{(l)}(x)$ of the l-th hidden neural layer is formally defined as [56]

$$a^{(l)}(x) = \sigma(\mathcal{W}_\theta^{(l)} a^{(l-1)}(x) + \mathcal{K}_\phi^{(l)} a^{(l-1)}(x)). \tag{6.2}$$

The output from a previous layer $a^{(l-1)}$ undergoes two transformations in such a layer. The linear layer $\mathcal{W}$ is used to keep information of the grid from the input data. The non-local integral layer $\mathcal{K}$ approximates an integral that helps approximate the input data (The other block of the Fourier layer shown in Panel (b) represents this operator). Finally, a non-linear activation σ is applied to the output of each layer. This non-linearity has several uses, as it helps the learning of hierarchical patterns, improves training, and enables the learning of complicated patterns from data. θ and

ϕ represent the adjustable parameters in $\mathcal{W}$ and $\mathcal{K}$ respectively since they are both neural networks. A non-local operator such as the kernel integrator is an operation or transformation that considers the local data point and the entire data through integration. The kernel integration operation for the kernel $\mathcal{K}$ over the coordinate space D can be defined as [56]

$$\mathcal{K}_{\phi}^{(l)} a^{(l-1)}(x) = \int_D \kappa_{\phi}^{(l)}(x,z) a^{(l-1)}(z) dz, \tag{6.3}$$

where $\kappa_{\phi}^{(l)}$ is a neural network parameterized by ϕ can be performed in the Fourier space. This integration operation in the Fourier space gives rise to the Fourier layer and subsequently, the FNO approach. Given that $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fourier and inverse Fourier transforms of a function $f(x)$ in the x and k domains, then the kernel integral operator from Equation 6.3 can then be written as

$$\mathcal{K}_{\phi}^{(l)} a^{(l-1)}(x) = \mathcal{F}^{-1}\left\{\mathcal{F}\{(\kappa_{\phi}^{(l)} * a^{(l-1)})(x)\}\right\} = \mathcal{F}^{-1}\left\{(\mathcal{F}\{\kappa_{\phi}^{(l)}(x)\} \cdot \mathcal{F}\{a^{(l-1)(x)}\})\right\}. \tag{6.4}$$

While a Fourier series expansion is mathematically infinite, a practical implementation of the integration requires terminating the Fourier series after a few terms. The number of terms of the Fourier series is defined as the modes of the Fourier series expansion. The maximum number of modes of each Fourier layer in the FNO model is a hyper-parameter defined prior to training the model. Practical implementations of the integration in the FNO model use the fast Fourier transform which includes terminating the Fourier series to these modes. The operations of the Fourier layer then become, the Fourier transform, filtering of relevant Fourier modes, and the inverse Fourier transform. These operations are depicted by the F, R, and F^{-1} blocks in Panel (c) of Figure 6.1. The Fourier layer integration operation can aggregate information but its periodic nature renders it insufficient for understanding the linear grids on which most input data operates. The linear part of this layer $\mathcal{W}$ ensures that the grid conditions of the model are also learned and added before the output is forwarded to the subsequent layer, which justifies its inclusion in Equation 6.2. Iterating through several FNO layers allows the model to learn local and global features from different instances of input data. To finally transform the output to the same dimensions as the desired output, a linear transformation layer, such as the output layer in Figure 6.1 Panel(c) is employed which gives the entire simulation as present in the output of the training samples.

The Fourier layers of the FNO allow for learning input-output relationships while keeping computational costs of integration relatively low [56], which is often an issue in neural operator frameworks. The FNO approach is also capable of transferring information learned on one grid resolution to another finer resolution without retraining the model. Another advantage of FNO models is that they provide one-shot inference, which means that entire output propagations are output in one go, in contrast to conventional ODE/PDE solvers. The FNO approach takes input instances in the range from $t = 0$ to $t = t_{\text{train}}$ timesteps and is capable of learning propagations from time $t = t_{\text{train}+1}$ to time $t = t_{\text{fin}}$. This means that even with one initial wavepacket density, the FNO model can output the entire propagation of a system when trained with sufficient data. For this project, the input consisted of wavepacket density plots concatenated with the information of potentials at all propagation timesteps. Grid information is also provided in the input in concatenation with the inputs defined previously.

6.3 Function optimization using Markov Chain Monte Carlo

PDEs have been extensively used to model physical phenomena. Researchers working in such domains, like quantum dynamics, are often interested in 'inverse problems'. These problems involve finding the initial state of a system (governed by a PDE), when the properties of the final state are known. Procedures to attain an initial state in such problems, often involve the optimization of a target function, that measures the similarity between the known desired state, and the result of propagating an initial state. Such function optimizations have typically been investigated using deterministic methods with great success. These methods involve using calculus to minimize (or maximize) the target function thereby searching for optimal parameters using mathematical principles. These optimization techniques are an active area of research in PDE-solving termed optimal control methods. Apart from optimal control, genetic algorithms can also be used to estimate the nature of optimal laser pulse shapes, as demonstrated by Thomas and Henriksen [129]. These methods, while effective are computationally expensive. As an alternative to both of these methods, in this project laser pulse optimization in quantum dynamics was explored using the Bayesian technique of Markov Chain Monte Carlo (MCMC).
Bayesian optimization methods [60] are a set of probabilistic methods that help optimize complex optimization functions. The Bayesian optimization approach relies on sampling methods to update its belief on optimal parameters as new data is sampled. Monte Carlo Sampling methods are an important subset of Bayesian optimization methods that can be used to sample optimal parameters from a distribution of input variables. MCMC sampling [130], a variation of Monte-Carlo sampling introduces dependencies in the sampling procedure using Markov chains, ensuring that the selection or rejection of a previous sample influences the samples that ultimately model the posterior distribution of variables. MCMC, when combined with a function evaluating target values, is a valuable tool for obtaining the desired optimal input parameters for the function [56, 61, 131]. Let $C(\mathrm{x})$ be the target cost function that depends on the set of parameters x specific to the problem, which needs to be optimized. The task of optimization then involves finding a set of optimal parameters $\mathrm{x}_{\mathrm{opt}}$ that maximizes C. The steps required to find optimal parameters using MCMC are as follows:

1. **Random Sample Selection:** To initialize the MCMC process a random sample of the initial variables is chosen from a pre-defined prior distribution of variables. A Gaussian distribution is a common choice for a prior for each variable that composes the target vector.

2. **Forward Step:** For the initial parameters, calculate the function value, $C(x)$, for the chosen parameters. For the first sample, this value is used to create a proposed distribution.

3. **Metropolis-Hastings (MH) Step [132]:** This step sets the criteria for selecting a new Markov state in the Markov chain. The decision to accept or reject the new sample, denoted as x' is based on a selection criterion defined below.

$$\text{Accept new sample } x' \text{ if } r \leq \min\left(1, \frac{C(\mathrm{x}')}{C(\mathrm{x})}\right) \text{ else reject}$$

The MH algorithm evaluates the likelihood of the new values of the cost function to the likelihood of the current Markov state. If the proposed new values of x improve this likelihood, then they are accepted as a new Markov state and added to the Markov chain. If the likelihood doesn't improve for the new set of points, there's still a chance of acceptance, which is determined using a probabilistic condition. If the new sample is rejected, the Markov chain stays at the current state, otherwise, the new sample is added to the Markov chain. If a new Markov state is added, the proposed distribution from step 2 is updated using this new state. The Metropolis-Hastings in this optimization step ensures that the Markov chain explores the parameter space not randomly, but by introducing criteria affecting the desired target distribution for the cost function. Different sampling methods such as Gibbs sampling [133] can be used to define the selection criterion. (Step 3 is not evaluated for the first sample, since it is the first point in the Markov chain.)

Optimization using MCMC involves repeating the cycle as described above, for several iterations, which leads to a distribution of target parameters for C for the input variables. Depending on the problem, the number of iterations required to converge to a narrow range of final overlap values can be different. Non-convergent samples at the beginning of the MCMC cycle can affect the quality of the target distribution, and eliminating them leads to better target distributions. Several tricks are used to eliminate such non-convergent samples. One such trick involves rejecting the initial non-convergent samples, which as termed as 'burn-in' samples (This is typically 10-20% of the initial points of the MCMC cycle are discarded, but the number is often problem-specific.). The final distribution after discarding the samples can then be used to infer knowledge about optimal parameters. A common technique to gain an idea of the final target distribution is to generate a smooth distribution using a process known as Kernel Density Estimation (KDE) [134], and then use the mean of the resulting distribution as the optimal set of parameters.

Chapter 7

Results

This dissertation explored the applications of ML in spectroscopy and quantum dynamics. The main results of this thesis appeared in the form of three peer-reviewed publications which are discussed in detail in this chapter. Each section of this chapter first provides a short outline of the main findings of each work, before treating each aspect of the investigations in detail. The final section of this chapter provides an outlook into the significance of the research activities done in the dissertation and research that can follow up on the work done in this dissertation.

7.1 GNNs for predicting excitation spectra

In Pub.1 a benchmarking of the performance of different GNN architectures was performed on a large spectroscopy dataset. Besides comparing GNN performance based on the quality of spectra, additional investigations were carried out to unravel the relationships between molecular structure and the quality of predictions made by different GNN architectures. To explore how the abundance of different functional groups in the training and test datasets affects the quality of their predictions, a statistical analysis was performed to show how easy or difficult it was for GNNs to predict spectra for molecules carrying certain functional groups. Finally, it was shown how unsupervised clustering methods can be used to understand the relationships between molecules with similar prediction accuracy.

7.1.1 Calculating spectra for the QM9* excitation spectra dataset

KS-DFT allows for the calculation of orbital energies of molecules and has been extensively employed to study ground and with some corrections excited state properties. Computing spectra with ground state DFT involves using techniques such as the core-hole approximation [135] to obtain transition states, which can then be employed to create spectra. Alternatively, TDDFT is employed in several spectroscopic calculations. The dataset used for calculations in Pub.1 contains excitation spectra that do not utilize any of these methods and were calculated by Ghosh et al. [29] in their work on exploring spectra prediction using ML. The spectrum calculations for the dataset in this work (termed QM9* henceforth) concern themselves only with the occupied states of molecules in the QM9 dataset [85], and excited states are not considered. Since the method for the calculation of these spectra is not one that is commonly encountered, it warrants further elaboration. The first step in the calculation of these spectra is to calculate the eigen-energies of the highest 16 occupied molecules using the PBE functional augmented with Tkatchenko–Scheffler van-der-Waals corrections (PBE+vdW) [136] as implemented in the Fritz Haber Institute ab initio molecular simulations (FHI-aims) code [137]. As a result of this design choice, molecules in the QM9 dataset with fewer than 16 occupied states are first discarded from the calculation. The spectra are then obtained by Gaussian broadening the energy values into the occupied density of states of the respective molecules. This broadening allows for the incorporation of the various vibrational, lifetime, and experimental effects into theoretical spectra, and is commonly used in computational simulations of various types of spectra.

This density of states incorporated spectrum, while not often encountered in chemistry, is nevertheless a qualitative spectrum that is easier to evaluate and not restricted to the different domains of spectra that depend on the excitation states that an electron is excited to. The notable difference is that it consists of the occupied states of a molecule, rather than its excited states (rather than transitions between occupied and excited states). However, the high quality of calculations in this dataset for a diverse set of molecules means that it can serve the purpose of testing the efficacy of various ML methods for the prediction of continuous chemical properties such as spectra. With this in mind, the dataset was used as the first point of investigation of the application of GNNs in spectroscopy.

7.1.2 Model selection and hyperparameter optimization

Four GNN frameworks, namely, GCN [138], SchNet [139], MPNN [91], and GIN [140] differ in their message-passing functions, leading them to generate different graph representations that are used to predict spectra were chosen for benchmarking. The performance of these GNNs was compared to the best results obtained by Ghosh et al. [29] using the Deep Tensor Neural Network (DTNN) [141] framework. The models were trained by minimizing the MSE loss on training data. The performance of the different models was compared by calculating their average RSE values on the test set.

Hyperparameter optimization, as discussed previously, is useful for training any ML model and the same applies to GNNs. In addition to the different parameters discussed previously, the number of message-passing cycles also affects the amount and quality of information encoded into the various feature vectors. Similarly, the dimension of the embedding into which different feature vectors are transformed for message-passing affects the quality of node embeddings obtained after message-passing. These additional hyperparameters were incorporated into the training process of the various GNN models. Optimal model hyperparameters for training these models were obtained through a grid search on a pre-defined space of hyperparameters. As with any model fitting exercise, steps need to be taken to ensure the models trained are not overfitted on the training data. For this task, the QM9* dataset, which contains 132k molecules was split into three sets: 80% for training, 10% for testing, and 10% for validation. The validation dataset (and the validation loss) was used to monitor the overfitting of the various models during the training process. To ensure the reproducibility of results, the data in each subset was kept consistent with the previous study. The Pytorch Geometric library [142] provides an efficient way to implement various GNN frameworks. All the GNN models in this work were implemented using this library.

7.1.3 Benchmarking model performance

The DTNN model trained by Ghosh et al. [29] required 10 days of training to obtain an average RSE value of 0.029 on the test set (as reported in their work). In contrast, all the GNN models trained in this work required between 3 to 6 hours of training (on a 32GB Nvidia Tesla GPU). The RSE values for these models on the same test data lay in the range of 0.023 to 0.039. The best-performing GNN model SchNet had an average RSE value of 0.023 which is a significant improvement over the DTNN value. Furthermore, analyzing the distribution of RSE values showed that the SchNet model had a high proportion of values lower than the mean value of 0.023, as shown in the left-skewed distribution in Figure 7.1. In contrast, the DTNN model can be seen to have higher average values, but fewer structures with extremely high RSE values (>0.1). Of the 13k molecules in the test dataset, 98% were found to have RSE values less than 0.05 for the SchNet model as compared to 87% for the DTNN model, further enforcing the superiority of the SchNet model.

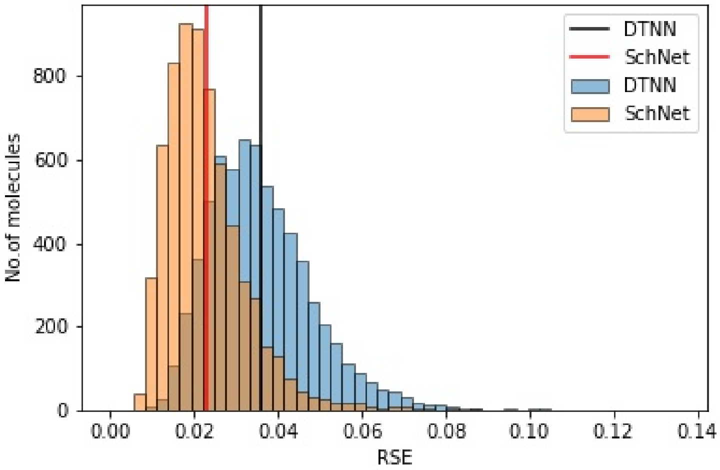


FIGURE 7.1: *Histogram demonstrating the RSE distribution for predicted test set spectra of the SchNet (orange bars) and DTNN (blue bars) models. The red and black lines mark the respective average RSE values of 0.029 and 0.033 respectively. Reprinted with permission from Singh et al. [103]. Copyright 2023 American Chemical Society.*

Besides benchmarking model performance, additional analysis was also performed to understand how molecular structure affects the RSE of spectrum prediction. To do this, the RSE values of the test set molecules were first obtained for models of each GNN architecture. To find whether the size of the molecules has an effect on prediction quality, a correlation between the number of atoms and the RSE values, as well as between the RSE values and the molecular weight, was investigated. No correlation was found between RSE values and molecular weight across all four GNN variants. While prediction quality improved with the number of heavy atoms (C, N, O, or F) in a molecule, this could be attributed to the relative abundance of molecules with 8 or 9 atoms compared to molecules with fewer than 8 atoms, rather than a correlation between the number of atoms in a molecule and predicted RSE values. These findings provide support to the claim that GNNs are capable of generating diverse molecular representations for molecules of different shapes and sizes.

7.1.4 Functional group based analysis of performance

Molecules are typically thought to be composed of various functional groups. Since the QM9* dataset consists of a selection of randomly generated molecules, the frequency of molecules containing these different functional groups also differs from one type of functional group to another. Additional analysis was therefore performed to investigate if GNN performance varies between molecules containing different functional groups. To accomplish this, the abundance of various functional groups in the dataset was first evaluated, and the average RSE values of the molecules in which these functional groups were then calculated for all four GNN models. The abundances of various functional groups were approximately found to be the same in the training and test sets ensuring that sampling bias did not affect prediction quality. In total 18 functional groups such as carboxylic acids, amides, alkenes, etc. were considered for this purpose. Groups such as ethers (44.21%) and alcohols (31.60%) were very abundant, while carboxylic acids (0.91%) and fluorides (1.63%) were rarely encountered in molecules of either the training or the test set. The RSE values of these functional groups also were found to be (mostly) reflective of their relative abundances. Molecules with alcohols (0.023) and ethers (0.022) were found to have lower RSE values on average than carboxylic acids (0.067) but not vastly different from fluorides(0.025). Despite high RSE values for poorly represented functionals, no clear correlation was found between the abundance of a functional group and RSE values. Further investigation was conducted, therefore, to identify structural relationships between poorly RSE values.

Another way of analyzing the relationship between different molecules and RSE values is by clustering molecular fingerprints. Molecular fingerprints use the SMILES representation of molecules to gather structural and functional information and encode them in the form of one hot encoding vectors. Morgan Fingerprints [143] of the test molecules along with the RSE values were therefore clustered together using the TMAP clustering algorithm [144] to unravel structural patterns between molecules of the test data and their RSE values. TMAP clustering of molecules resulted in the creation of a tree where different molecules appear as leaves/nodes. Molecules on the same branch have similar Morgan fingerprints and are similar in structure. The leaves of the TMAP were then colored with RSE values to provide a relationship between RSE values and molecular structure. Figure 7.2 depicts the TMAP of RSE values obtained for the test set (calculated for the SchNet model) with the blue nodes having low RSE values and red nodes having high RSE values. Most leaves of the TMAP were blue, due to the high prediction quality for most structures. Molecules with high RSE values were colored in bright neon green and red colors. Further, molecules that have high RSE values were found clustered together as highlighted in the figure.

Once the difference in prediction quality for functional groups for the SchNet model was established, further investigation was carried out to investigate whether similar patterns could be found for other GNN models. For each functional group, therefore, a relative ease of prediction metric was defined. This metric provides a statistical window into how the RSE of molecules containing a particular functional group compares to the average RSE of the rest of the molecules. Performing this for all functional groups across all models gives an idea of whether the prediction quality for different functional groups has a pattern across different GNNs. Across the four models, for example, the ease of prediction metric showed that carboxylic acids are difficult to predict. Some molecule types, such as those with fluorides are easier for a model like the GCN but difficult for other models. This metric thus, gives further insight into how feature design for functional groups could be improved to improve predictions.

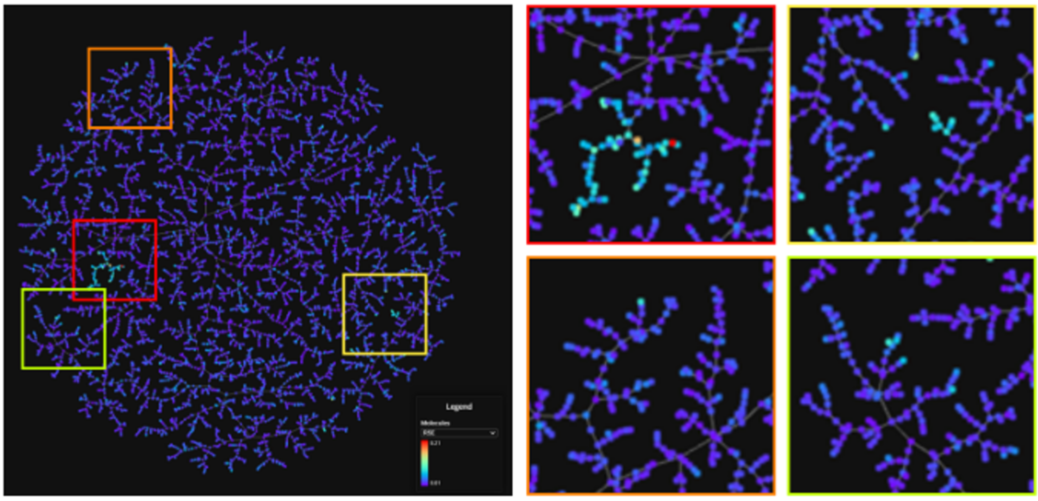


FIGURE 7.2: *TMAP clustering to understand the relationships between different test set molecules. The highlighted regions (marked on the left and enlarged on the right) depict regions of interest i.e. where molecules that have high RSE values are clustered together. Reprinted with permission from Singh et al. [103]. Copyright 2023 American Chemical Society.*

7.2 Incorporating explainability into GNNs for the prediction of XAS

Analysing the performance of GNNs, as in Pub.1, sheds light on the differences in the ability of different GNNs to predict spectra, as well as the differences in prediction quality for different functional groups, but does not provide information on how these differences arise. Complex ML models such as GNNs, are black-box by nature, and understanding how they arrive at predictions requires using a set of mathematical tools collectively known as Explainable Artificial Intelligence (XAI) methods. To understand how GNNs predict spectra, therefore, an investigation into the interpretability of different GNN models predict X-ray spectra was performed in Pub.2. Since X-ray absorption spectra (XAS) are more practical and used widely in materials research, a dataset containing X-ray spectra of QM9 molecules was chosen over the QM9* dataset of the previous work. Such an investigation requires, in addition to an XAI method, a ground truth to compare to the results of the XAI method. The development of such a ground truth for complex chemical properties such as XAS is a challenging task, and in Pub.2 an approach to automatically generate ground truths from quantum chemical calculations of XAS was developed. The XAI technique of Class Activation Map (CAM) was then used to provide an insight into which atoms of a molecular graph a GNN assigned importance to, across different regions of a spectrum. An area under the curve (AUC) metric was then used to quantify the similarity between the results of the CAM and the ground truth for each molecule in the test data. Average AUC values were then used to infer the degree of interpretability of different models and investigate the effect of various feature changes on the interpretability of a GNN model.

7.2.1 X-ray Absorption Spectra for QM9 molecules

To investigate GNN performance and interpretability a dataset containing XAS spectra of molecules was first created by coworkers of Pub.2. This dataset, termed QM9-XAS contains the XAS spectra of 56k molecules chosen randomly from the QM9 dataset. Carbon K-edge XAS of these molecules were calculated by exciting electrons from the C 1s orbitals of different carbon atoms in the molecules. These spectra typically show interesting features in the range of 276 eV to 300 eV, hence excitations in this range were considered for creating XAS spectra. Spectra were calculated with the ORCA quantum chemistry [145] package using TDDFT and the TDA approximation using the B3LYP functional. A TZVP basis set was chosen to ensure high-quality calculation of the excitation energies of various molecules. To further ensure the accurate calculation of spectra, a sufficiently large amount of excited states was chosen for the calculation of transitions.

7.2.2 Model training and performance

Three different GNN architectures were chosen to understand the differences in the interpretability of various GNNs. The GCN [138] is one of the first GNNs applied to molecules and uses a simple convolution approach to update node embeddings in its message-passing process. Graph Attention Networks (GATv2) [146] employ an attention mechanism [147] to update node embeddings, giving them the ability to dynamically select important nodes from the neighborhood of a node. The third GNN chosen for this work was the GraphNets [148] framework which updates node representations using the total graph representation in addition to updating representations using neighborhood nodes and edges during the message-passing process. Incorporating this information can arguably improve the understanding of regions away from the neighborhood of the node, which can have an impact on the final node representation of a graph. Training and hyperparameter optimization was performed in an approach similar to the protocol described for Pub.1. The training data consisted of 40k molecules, the validation set of 10k molecules and the test set contained 6k molecules. Models were trained to minimize MSE loss, and prediction quality was reported using average RSE values on the test set.

The three GNNs had variations in their average performance, due to the differences in their message-passing architecture. The average RSE values on the test set for the GCN, GATv2, and GraphNet models were reported to be 0.047, 0.031, and 0.042 respectively, showing that the GATv2 model performs better than the other two models. To understand how these values translate into actual spectra, we refer to Figure 7.3 which shows the best, worst, and average predictions of the different models for molecules of the test set. Across all three models, the best predictions accurately replicate the TDDFT spectra in peak shape and intensity. Average predictions are capable of replicating most features of the spectra but often do not accurately depict intensities. Poor spectra show a mismatch in both intensities and peak shapes.

7.2.3 Interpretability analysis

The main goal of this work was to investigate the interpretability of these predictions and for this, the feature attribution technique of CAMs was applied to the trained GNN models. Along with this, a ground truth benchmark was developed to compare the CAM heatmaps for different molecules using the explainability approach discussed earlier. An XAS spectrum is composed of various transitions, which originate in one of the core orbitals of an atom, and end up in an unoccupied orbital often comprising molecular orbitals delocalized over several atoms. The ground truth benchmark developed in this work gives two heatmaps for each spectrum: the core ground truth showing the core atom contributions, and the virtual ground truth which shows the atoms involved in orbitals where the excited orbitals in an XAS

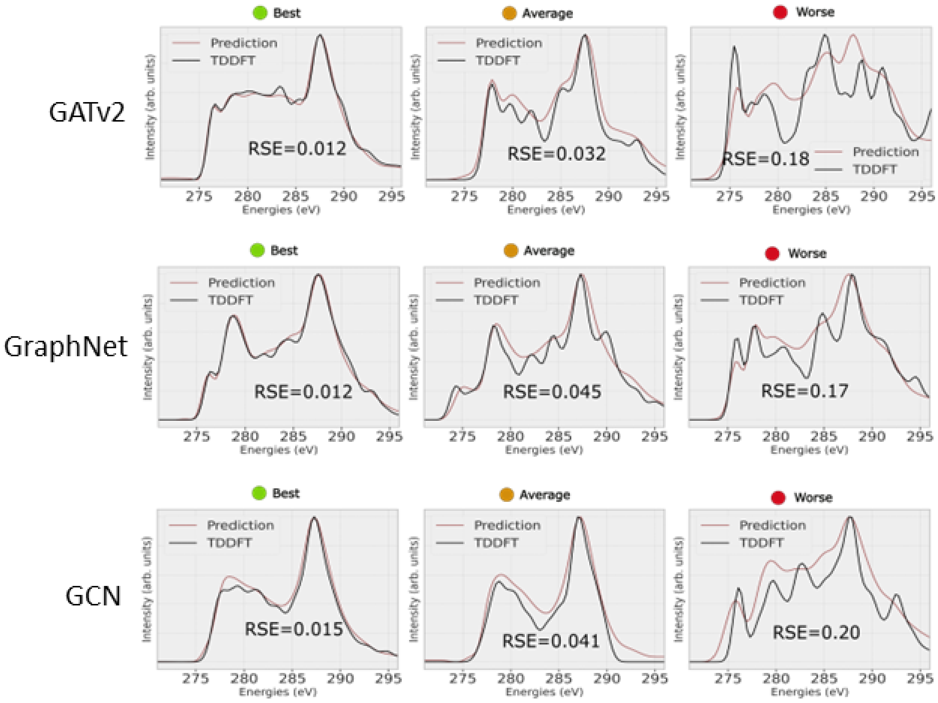


FIGURE 7.3: *Best, worst, and average predictions of GCN, GATv2 and Graphnets models. The good predictions for all models replicate the TDDFT spectra (left column), the middle column shows how an average prediction covers most features of the spectrum accurately, while the last column shows the worst predictions with mismatching intensities and shapes. Reprinted with permission from Kotobi et al. [113].*

transition end up. To compare the ground truth to the CAM obtained for various spectra, the AUC metric was employed (A visual representation of this comparison is provided in Chapter 4.). This approach lends two AUC values comparing CAMs and ground truths i.e., the CAM for the core and virtual atoms. Once the average AUC metric is obtained for all the models, it can be used to compare the degree to which the predictions of the models are interpretable for XAS prediction.

The average AUC for virtual and core comparisons for the GCN model were both found to be around 0.5. This value is comparable to a random classifier that assigns labels of contributing atoms for an XAS prediction randomly in a molecule implying that the predictions made by a GCN model are not interpretable per the ground truth logic. In contrast, the GATv2 and GraphNet models were both found to have superior average attribution scores with median values for core and virtual orbitals being greater than 0.7, which lends them a higher interpretability than the GCN model. The higher interpretability of the GNN architectures can be attributed to their superior message-passing architectures compared to the GCN. While the attention mechanism of the GATv2 allows for the incorporation of neighborhood information dynamically, GraphNets have higher interpretability due to the incorporation of graph-level information. XAS arises out of a complex interplay of different structural and functional factors in a molecule, which means that the patterns to be learned by a GNN from XAS data are complex. In such a situation, therefore, enriching node information via the attention mechanism or by incorporating graph-level information could be crucial in improving performance and interpretability.

To understand the relationship between the quality of predicted spectra and the interpretability across different models, for each model, the test set was divided into deciles according to RSE values, and the variation of AUC values for each decile was studied. While the AUC did not show any variation across the different deciles for the GCN model, a pattern was found for the other models. For both the GATv2 and GraphNet models, the interpretability was found to be high for higher-quality spectra and declined with a fall in prediction quality.

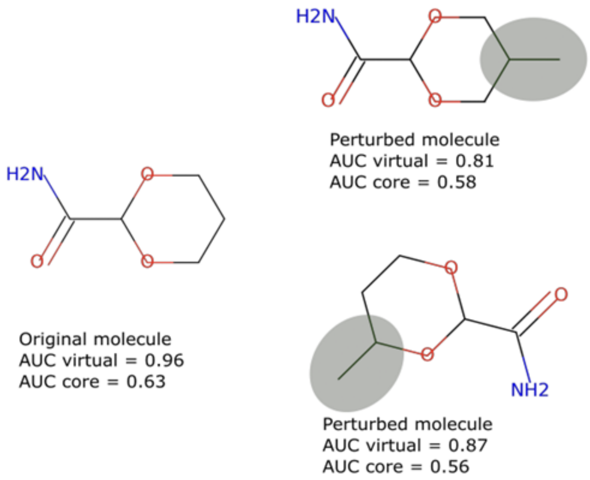


FIGURE 7.4: *Depicting the change in AUC values due to the addition of a methyl group. The AUC values for the core and virtual ground truth are higher for the original molecule (0.96 and 0.63). Adding a methyl group to random positions in the molecule (top and bottom) causes a decline in both AUC values for both structures. Reprinted and adapted with permission from Kotobi et al. [113].*

Biases in a dataset arise due to differences in the frequency of different types of molecules, which in turn can bias the model to learn spurious correlations. Interpretability can also help identify dataset biases and be used to test the robustness of models. To study how robust a model is to such spurious correlations, a few structures from the QM9 dataset were chosen and perturbed by adding a methyl group randomly to the molecule. Altogether 40 molecules that do not appear at all in the QM9 dataset were obtained using this perturbation. Adding such a perturbation alters the TDDFT spectrum, as well as the predicted spectrum of the original molecule. For all three GNN models, the perturbation unanimously was found to cause a drop in prediction quality and interpretability. The average RSE value for the unperturbed structures for the GraphNets model was 0.03. For the perturbed structures, the average RSE dropped to 0.13 for the same model, representing a significant decline in performance for these new structures. A purported reason for this drop could be the increase in the size of molecules upon the addition of the methyl group. Most QM9 molecules have a maximum of 9 heavy atoms (C, N, O, F), while the perturbed molecules, on average, were found to have more than 9 atoms, indicating the difficulty in generalizing to molecules of larger size than the model has been trained on. The change in RSE values for these structures was found to be even higher for the GATv2 and GCN models. Figure 7.4 depicts how the addition of a methyl group causes a fall in the AUC values for a structure from the test set. As seen from the figure, the AUC values of both core and virtual ground truths drop when the functional is added, irrespective of the site at which the addition is performed. The % drop in values for the virtual ground truth in this case was found to be higher than the corresponding drop for the core ground truth. This behavior was also representative of the general trend observed across all molecules for all models, where the drop in AUC core values was much smaller than the corresponding values for the virtual ground truth.

7.3 Machine Learning wavepacket propagation

Pub.3 explored how ML can be used to predict wavepacket propagation for quantum dynamical systems. To accomplish this, Fourier Neural Operator (FNO) models were trained on simulation data created by propagating wavepackets for two different quantum dynamical systems. FNO models were found to accurately and faithfully reproduce the propagation of wavepackets for both systems. The dynamics of one of these systems involved a wavepacket trapped in a double well, propagating under the influence of a laser-induced electric field. Having trained an FNO model to simulate the propagation of a wavepacket under different electric fields, Pub.3 demonstrated how the Bayesian technique of Markov Chain Monte Carlo (MCMC) could be employed to find optimal laser pulses that can successfully induce the wavepacket to propagate from one part of the double well potential to another. Such an MCMC-coupled optimization typically requires several thousand iterations over different laser parameters, which was made possible by the high inference speed and single-shot propagation capabilities of the FNO models, thereby demonstrating the viability of this pulse optimization approach as an alternative to methods such as optimal control.

7.3.1 Test systems

To demonstrate the ability of ML models to learn the dynamics of wavepacket propagation, two test systems were chosen. One of these systems exists in one spatial and one temporal dimension, while the other (2D) has two spatial dimensions and one temporal dimension. The type of potential in each system was also different, as in the 1D system the potential in the Hamiltonian is time-dependent, while the 2D system has a static potential. This variability between systems allows for the demonstration of the versatility of the ML methods, as well as their applicability to systems under different conditions.

Two-dimensional anharmonic potential system

It is common practice to use polynomial functions to approximate the potential surface of various quantum systems. Anharmonic potentials are one such form of polynomial potentials that have been extensively used to model systems involving small molecules, or their vibrational modes [149–151]. The propagation of a wavepacket under an anharmonic potential in two dimensions was therefore chosen as one of the systems for investigation in this project. The wave packet in such a system is a function of two spatial coordinates and time. Given an anharmonic potential of the form

$$V(q_1, q_2) = \frac{1}{2}(q_1^2 + q_2^2) + \lambda q_1^2 q_2^2 \tag{7.1}$$

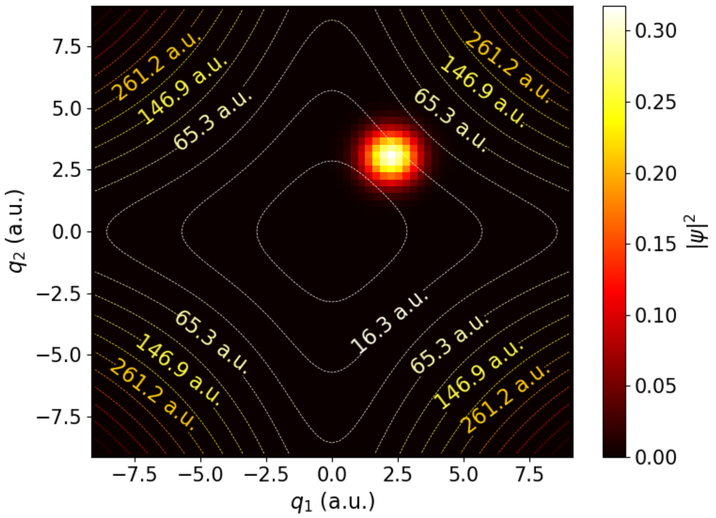


FIGURE 7.5: *The density $|\Psi|^2$ of a randomly chosen initial wavepacket in the two-dimensional anharmonic potential. Contour lines show the potential value (in a.u.). Reprinted from the manuscript submitted at J.Chem.Comput.*

the Hamiltonian of the system is a function of two spatial dimensions q_1 and q_2 thereby leading to the TDSE for the system (in a.u.) as

$$i\frac{\partial}{\partial t}\Psi = -\frac{1}{2}\left(\frac{\partial^2}{\partial q_1^2} + \frac{\partial^2}{\partial q_2^2}\right)\Psi + \frac{1}{2}(q_1^2 + q_2^2)\Psi + \lambda q_1^2 q_2^2 \Psi. \tag{7.2}$$

The parameter λ can be varied to give access to different Hamiltonians under which a system can evolve. The initial wavepacket that was used for studying this system was a product of two Gaussians centered at random points in the spatial grid. Randomly varying the initial position coordinates allows for placing the wavepacket at different points in the spatial grid, thereby giving access to a variety of initial conditions. A combination of the initial wavepackets and their different potentials thus allows for creating a dataset to train ML models for this system.

Laser driven tunneling in a 1D double well potential

The second system is a 1D double well potential that has been extensively used to approximate the rotation of the molecule $F_2H_3C_6 - C_6H_3Br_2$ along the torsional bond connecting the two phenyl rings. Figure 7.6 (a) shows the molecule and the torsional bond whose rotation is measured along the dihedral coordinate (ϕ_d). This system has been extensively investigated [129, 152], both theoretically and experimentally. The rotation of the molecule along the bond can be approximately modeled as the propagation of a 1D wavepacket along the dihedral angle coordinate. The time variation in the Hamiltonian (through the potential) occurs due to the influence of a laser, which carries out this rotation. In the 1D system, this amounts to the tunneling of the wave packet initially localized on the left side through the barrier and into the right side of the double well, as depicted in Figure 7.6 (b).
The steady-state Hamiltonian of this system, without the influence of an external field, is given by Equation 7.3 as below:

$$\hat{H} = \frac{1}{2I_{rel}}\left(-\hbar^2\frac{\partial^2}{\partial \phi_d^2}\right) + V(\phi_d) \tag{7.3}$$

The double-well potential $V(\phi_d)$ in which the wavepacket is initially confined is given by Equation 7.4. This potential has been verified experimentally in previous studies [152, 153] and its functional form was obtained using DFT calculations for various molecular configurations of the system in the aforementioned studies. I_{rel} in this equation is the relative moment of inertia of the molecule defined as $I_{rel} = I_{Br}I_F/(I_{Br} + I_F)$.

$$V(\phi_d) = \sum_{n=0}^{6} A_n \cos(n\phi_d) \tag{7.4}$$

The time-varying laser pulse applied to the system induces an electric field ϵ which alters the potential of the system. This induced electric field changes the Hamiltonian of the system to

$$\hat{H}' = \hat{H} - \frac{1}{4}\epsilon^2(t)\alpha(\phi_d), \tag{7.5}$$

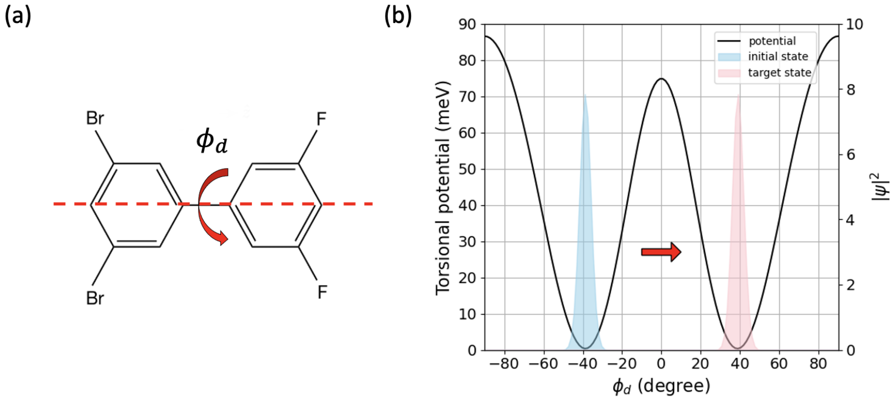


FIGURE 7.6: *(a) The molecule $F_2H_3C_6 - C_6H_3Br_2$ along with its dihedral angle ϕ_d, along which the rotation occurs. (b) The double well potential depicting the confined wavepacket on the left side (blue) and the desired final state of the rotated molecule(pink). Reprinted from the manuscript submitted at J.Chem.Comput.*

where $\alpha(\phi_d)$ is the molecular polarizability as calculated in [152]. Since molecules are not in a steady-state under room temperature, but in vibrational motion, the double well potential is itself not static, and slightly varies in time, leading to changes in both $V(\phi_d)$ and α. These terms are therefore additionally modified to include perturbations calculated using the technique elaborated in [129]. The complete Hamiltonian of the system considers effects from the vibration of the molecule and the electric field that results from applying a laser field. For this system, an initial state for the propagation can be chosen to be the initial state is an eigenstate of the ground state Hamiltonian which is localized on the left side of the double well (The vibrations induce asymmetry in the double well, which allows for such a localization). This initial state can be calculated for the system using a method such as the Fourier Grid Hamiltonian Method [154]. Different time-varying pulses can then be applied to the system, leading to different outcomes in the propagation of this wavepacket, helping the creation of a varied dataset to enable learning of the propagation for different conditions using ML.

Not all laser pulse-induced electric fields are capable of performing the tunneling of the wavepacket for this system. Laser pulses that are too low in intensity will fail to induce any tunneling and only result in oscillations on the left side of the wavepacket. Too strong lasers could ionize the molecule by breaking the bond, which is not desirable. Inducing a change as in the 1D test system using a laser can therefore be considered an optimization problem. Such optimization involves finding the initial conditions of the system to attain a final target system and can be referred to as an inverse problem. To do so, it is first necessary to define a reference target state that the laser pulse must be able to translate the initial wavepacket into. The optimization procedure then tries to maximize the degree of overlap between a final state at the end of the wavepacket propagation with this reference state. Let $C(\mathrm{x})$ be such a function that depends on the vector of laser pulse parameters denoted by x. Then, the task of pulse tuning for the 1D system translates to finding the optimal set of parameters $\mathrm{x}_{\mathrm{opt}}$ that maximizes C. The use of MCMC optimization to maximize this function C using FNO to evaluate final states for different laser pulses was explored in Pub.3.

7.3.2 Dataset generation and model training

To demonstrate wavepacket propagation using ML datasets for the two test systems were generated. For the 1D system, the split operator method was employed for propagating the initial wavepacket density described earlier. Different induced potentials owing to the varied laser pulse parameters, lead to a variety of Hamiltonians that influence the simulation leading to different final states at the end of the propagation. Thus, using the fixed initial wavepacket and different laser pulses a total of 4,000 different simulations were generated using the split operator method. For the 2D system, the parameter λ and the initial wavepacket provide access to the creation of a dataset for this system. In total using MCTDH to propagate wavepackets for this system, a dataset of 1,000 propagations was generated. For each wavepacket (in both systems), the corresponding wavepacket density was used in the input data for the FNO model.

Data generated from PDEs is different from usual output data in other applications, as the subsequent instances in a trajectory (propagation) are continuous (and thus, dependent on one another) since most phenomena modeled by PDEs are continuous. Hence, while MSE loss can be used to train a model like the FNO, it is not an optimal choice. The models trained in this work utilize therefore a special loss function, named the Sobolev loss function [155, 156]. The Sobolev loss function considers the transition between two consecutive time instances in propagation by introducing gradients in the loss function, thereby penalizing discontinuous outputs. This loss function hence is used where a smooth approximation of the output is necessary, e.g. in image denoising and PDE solving [155].

Two different FNO model architectures were used to train the two different systems since these differ in dimensionality. Hence the FNO 2D model which models systems varying in two coordinates was used for the 1D test system (which has one spatial and one temporal coordinate), while the FNO 3D model was used for the 2D system, which has two spatial coordinates and one temporal coordinate. Hyperparameter optimization for the FNO model involves using hyperparameters that are specific to the FNO model are the number of Fourier modes the number of Fourier layers, and the width of each linear layer in the FNO model, in addition to the general hyperparameters previously discussed. The FNO model for the 1D system required 4h 51 mins to train while the 2D system required 26 mins of training. The FNO model for the 1D system has 629 million trainable parameters compared to 205 million for the model for the 2D system and requires more epochs to train leading to a longer training time for the former.

7.3.3 Learning wavepacket propagation with ML

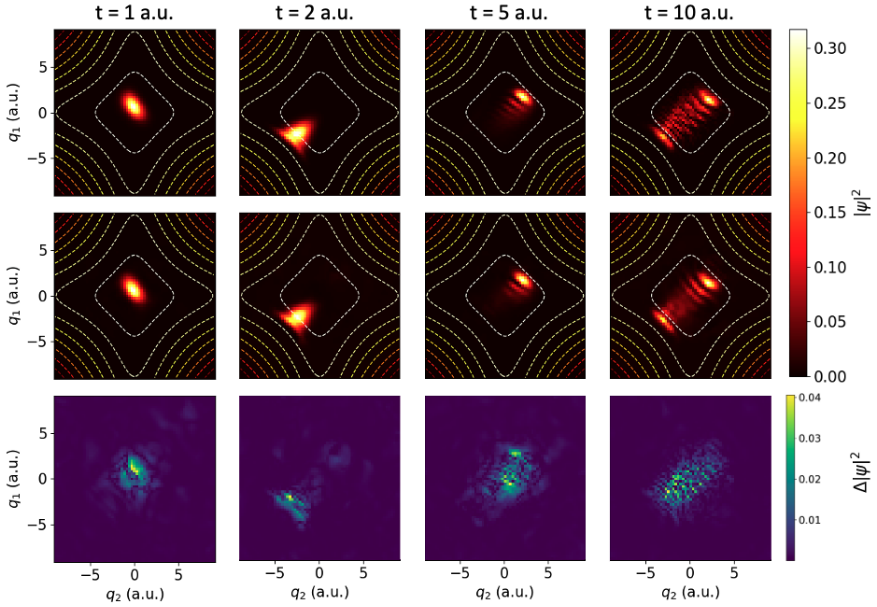


FIGURE 7.7: *Comparison of wavepacket densities generated by the MCTDH and FNO methods at various timesteps for a randomly chosen test sample. The top panel displays the densities at t=1 t=2, t=5, and t=10 a.u. generated from their corresponding wavepackets with MCTDH. The middle panel shows corresponding values at these timesteps for the same initial density predicted by the trained FNO model. The bottom panel illustrates the absolute difference error between the wavepacket densities generated by the two methods. Reprinted from the manuscript submitted at J.Chem.Comput.*

Figure 7.7 provides a comparison between an FNO-predicted propagation and its MCTDH counterpart for a randomly chosen test sample at four different timesteps during the propagation. The initial wavepacket density for this sample is the same as the one depicted in Figure 7.5. Wavepacket densities obtained from wavepackets at different timesteps used for the MCTDH propagation are shown in the top panel of the figure. As can be inferred from a visual comparison of the two panels, there is very little difference in the MCTDH and FNO results, demonstrating the ability of FNO models to accurately predict trajectories at different timesteps for this system. The bottom panel shows the absolute difference between the densities obtained from MCTDH and FNO methods. The absolute difference also shows some 'residual' density at various points in the grid for the FNO model, which is the source of error between the two models. The error is itself more concentrated in regions where the wavepacket propagation takes place, but besides this, there is no discernable pattern to the error propagation in the sample.

7.3.4 Pulse tuning using Bayesian optimization

A major advantage of ML methods like the FNO model is their speed of inference. This means that while training an FNO model is expensive, once trained the model is capable of accurately generating propagation for input data at a very high speed compared to conventional PDE solvers (for the 2D system the speedup was by a factor of 10^3). This speedup can be useful in downstream procedures that require generating a large number of outputs and iterating over them. To test whether such gains in speed can be useful in applications a laser pulse optimization using the FNO model and the MCMC approach was carried out. To accomplish this, a cost function was first defined to compare how close the target state (defined for the system) is with the final state obtained from propagation. The MCMC-FNO optimization then was used to find the set of laser pulse parameters that maximize this cost function. For this purpose, a functional form of the laser pulse was first defined. The MCMC approach then involved using randomly generated laser pulses (using random parameters in the functional form) to propagate wavepackets in the FNO model. The laser pulse chosen for the process was a chirped pulse which has a functional form with 6 variables given by

$$\epsilon(t) = Ae^{-\frac{1}{2}(\frac{t-t_c}{\sigma})^2} cos(2\pi(f+\beta(t-t_c))(t-t_c)+\theta). \tag{7.6}$$

Of these, three variables t_c, β, θ were kept fixed for the MCMC process, while the other three were varied during the optimization. The MCMC process case was repeated for 20 different seed values, to demonstrate the stability of this approach for designing an optimal laser pulse shape. For each seed, the pulse optimization (which includes predicting the propagation using FNO) took approximately 5 minutes to complete. Given that there were 5,000 different samples in each cycle the computational cost for each MCMC cycle is quite low. While the optimization was constrained only to three variables, it is possible to extend the process to optimizing all variables of the laser pulse. This would require more samples to train the model and more training time. The number of samples required to estimate a distribution using MCMC would also in this case be greater than the 5,000 iterations performed in this work.

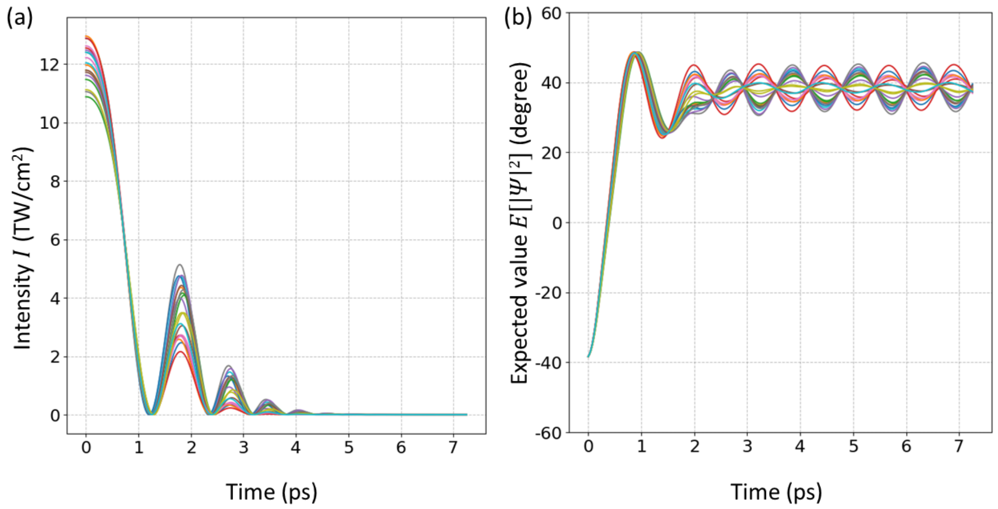


FIGURE 7.8: *Plots depicting the outcomes of 20 MCMC pulse optimization tasks utilizing the FNO model. The left figure exhibits 20 pulse shapes, all exhibiting a consistent pattern of peaking at the simulation's outset and subsequently decreasing over the next cycles. The right plot portrays the change in the expected values of the wavepacket density over time. Initially, all wavepackets are concentrated on the left well, denoted by negative values. Applying the pulse transports them to the right, reflected in the positive values of the expected density. Reprinted from the manuscript submitted at J.Chem.Comput.*

Figure 7.8 demonstrates the results of the MCMC-FNO pulse optimization process for these 20 seed values. The left side of the figure shows the pulse shapes obtained for the different MCMC cycles. As seen in the figure, all the pulses have similar profiles, with the first pulse being the highest in amplitude, followed by shorter subsequent pulses that decrease in time. The right-hand side of figure 7.8 shows the expected value of the wavepacket density on the coordinate grid. These expected value plots show, that the laser pulses indeed bring about the tunneling with the first large pulse, while the subsequent pulses keep the wavepacket confined (and oscillating) in the right-hand side of the well, as indicated by the change from negative expected values to positive after the application of the pulse. The physical reason for the laser pulse profile can be justified by using the following logic: The first pulse has to be the largest in amplitude as it seeks to immediately drive the tunneling. The subsequent pulses, on the other hand, act as stabilizers, preventing the wavepacket momentum from carrying it back to the other side of the well. The degree of overlap between the desired target state and the states obtained from the MCMC processes ranged between $0.7 - 0.95$ indicating a high degree of similarity between the final states and the desired target, thereby demonstrating the viability of this approach for laser pulse optimization at low compute costs.

7.4 Conclusion and research outlook

Graph Neural Networks (GNNs) are the latest in a long line of ML algorithms used for chemical property prediction that have already demonstrated significant promise in property prediction and structure generation tasks. The work on GNNs for predicting spectra in this dissertation has provided a solid platform to explore the applicability of GNNs to larger structures. A novel GNN approach is being investigated by Prof. Bande and co-workers for the prediction of XAS spectra of functionalized coronene and circumcoronene molecules. I have contributed to this through the creation of a dataset of XAS spectra for randomly generated molecules using an in-house algorithm. A GNN framework capable of predicting site-specific XAS at different atoms in these molecules is also being investigated for this purpose. Such an approach will enable the usage of GNNs to predict spectra for macromolecules such as graphene oxides as an alternative to computing expensive quantum chemical spectra using DFT or TDDFT based methods.

Chemical property prediction using ML is expected to be a major part of several research pipelines in materials and drug research. A large number of these chemical properties arise due to complex electronic structure interactions within different atoms of a molecule. While ML models can discern various patterns from large amounts of data, these might not always correlate to scientific logic. Pub.2 was an attempt to understand the decision-making of ML models used for complex chemical property prediction tasks. It demonstrated how novel ground truth benchmarks can be generated using quantum chemical properties, thereby reducing the reliance on expert opinions and empirical evidence for investigating ML model interpretability. This approach can be extended to various chemical property prediction tasks as an added tool to measure model and dataset suitability for the task at hand. Once again, applying such an approach to large molecules in tasks such as prediction of molecular spectra can provide valuable insights into structure-property relationships, that are difficult to discern using other methods.

The work on FNOs in this dissertation demonstrates their ability to learn quantum dynamics propagations in different systems. The speed of inference of these models gives them the ability to act as low-cost surrogates for exploring parameter spaces in dynamical systems before a high-accuracy method is used. A natural extension of the work done in this dissertation is to use FNOs for systems with three spatial coordinates and one temporal coordinate. Another avenue for exploration is the use of physically inspired loss functions to train FNOs, which can improve the prediction quality of these models. Data generation for quantum dynamical systems in large dimensions can be computationally expensive and even here ML may be useful. For example, dimensionality reduction using ML can help find efficient approximations for quantum dynamical models with large dimensions. By reducing the number of dimensions, large systems can be broken down into smaller ones, making it easier to generate data. This can extend the applicability of ML methods that predict solution operators such as the FNO models to systems larger than four dimensions. Furthermore, the discovery of PDEs using symbolic regression (enhanced with ML) might be useful to improve the ability to model PDEs for quantum dynamical systems.

Applicability to experimental data remains a critical bottleneck in the usage of ML for chemistry. The work done in this project has largely relied on data from quantum chemical calculations. As such, the spectrum predictions of these ML models do not take into account the various experimental effects arising from solvent interaction, lifetime broadening, molecular vibrations, etc. Where experimental data are available, the ML architectures used in this work can be trained to predict results, but it is difficult to determine the accuracy of the predictions without having performed this exercise. When experimental data is unavailable, however, it is recommended to perform investigations that allow for incorporating experimental corrections in these models, either during data creation and collection or in a post-hoc manner by cascading different ML approaches together. ML applications in chemistry are at a nascent stage and we have barely scratched the surface in terms of how ML can revolutionize chemistry. The work presented in this dissertation will help to further the application of ML in different fields of quantum chemical research.

Bibliography

(1) Schleder, G. R. et al. *Journal of Physics: Materials* **2019**, *2*, 032001.

(2) Hey, A. J. G., *The fourth paradigm: data-intensive scientific discovery*; Microsoft Research: 2009.

(3) Le, T. et al. *Chemical Reviews* **2012**, *112*, 2889–2919.

(4) Schrödinger, E. *Annalen der Physik* **1926**, *384*, 361–376.

(5) Keith, J. A. et al. *Chemical Reviews* **2021**, *121*, 9816–9872.

(6) Hansen, K. et al. *The Journal of Physical Chemistry Letters* **2015**, *6*, 2326–2331.

(7) Cova, T. F. G. G.; Pais, A. A. C. C. *Frontiers in Chemistry* **2019**, *7*.

(8) Fooshee, D. et al. *Molecular Systems Design & Engineering* **2018**, *3*, 442–452.

(9) Meuwly, M. *Chemical Reviews* **2021**, *121*, 10218–10239.

(10) Schmitz, G.; Godtliebsen, I. H.; Christiansen, O. *The Journal of Chemical Physics* **2019**, *150*, 244113.

(11) Noé, F. et al. *Annual Review of Physical Chemistry* **2020**, *71*, 361–390.

(12) Wang, Y.; Lamim Ribeiro, J. M.; Tiwary, P. *Current Opinion in Structural Biology* **2020**, *61*, 139–145.

(13) Rydberg, J. *Kongliga Svenska Vetenskaps-Akademiens Handlingar* **1889**, *23*, 1–177.

(14) Whittaker, E. T., *A history of the theories of aether & electricity*; Dover Publications: New York, 1989.

(15) Ratcliff, L. E. et al. *WIREs Computational Molecular Science* **2017**, *7*, e1290.

(16) Extance, A. Quantum Chemistry Simulations Offer Beguiling Possibility of Solving Chemistry.

(17) Einstein, A. *Annalen Der Physik* **1905**, *322*, 132–148.

(18) Nordling, C.; Sokolowski, E.; Siegbahn, K. *Physical Review* **1957**, *105*, 1676–1677.

(19) Stöhr, J., *NEXAFS Spectroscopy*; Springer Series in Surface Sciences, Vol. 25; Springer Berlin Heidelberg: Berlin, Heidelberg, 1992.

(20) Westermayr, J.; Marquetand, P. *The Journal of Chemical Physics* **2020**, *153*, 154112.

(21) McNaughton, A. D. et al. *Journal of Chemical Information and Modeling* **2023**, *63*, 1462–1471.

(22) Mamede, R.; Pereira, F.; Aires-de Sousa, J. *Scientific Reports* **2021**, *11*, 23720.

(23) Gastegger, M.; Behler, J.; Marquetand, P. *Chemical Science* **2017**, *8*, 6924–6935.

(24) Jung, G.; Jung, S. G.; Cole, J. M. *Chemical Science* **2023**, *14*, 3600–3609.

(25) Zhang, W. et al. *Sensors* **2022**, *22*, 9764.

(26) Guo, S.; Popp, J.; Bocklitz, T. *Nature Protocols* **2021**, *16*, 5426–5459.

(27) Ralbovsky, N. M.; Lednev, I. K. *Chemical Society Reviews* **2020**, *49*, 7428–7453.

(28) Madden, M. G.; Ryder, A. G. **2003**, *4876*, 1130–1139.

(29) Ghosh, K. et al. *Advanced Science* **2019**, *6*, 1970053.

(30) Aarva, A. et al. *Chemistry of Materials* **2019**, *31*, 9243–9255.

(31) Aarva, A. et al. *The Journal of Physical Chemistry C* **2021**, *125*, 18234–18246.

(32) Golze, D. et al. *Chemistry of Materials* **2022**, *34*, 6240–6254.

(33) Guda, A. A. et al. *npj Computational Materials* **2021**, *7*, 203.

(34) Rankine, C. D.; Madkhali, M. M. M.; Penfold, T. J. *The Journal of Physical Chemistry A* **2020**, *124*, 4263–4270.

(35) Rankine, C. D.; Penfold, T. J. *The Journal of Chemical Physics* **2022**, *156*.

(36) Carbone, M. R. et al. *Phys. Rev. Lett.* **2020**, *124*, 156401.

(37) Doshi-Velez, F.; Kim, B. *arXiv:1702.08608 [cs, stat]* **2017**.

(38) Danilevsky, M. et al. *arXiv:2010.00711 [cs]* **2020**.

(39) Abhishek, K.; Kamath, D. *arXiv:2211.14736 [cs]* **2022**.

(40) Wellawatte, G. P. et al. *Journal of Chemical Theory and Computation* **2023**, *19*, 2149–2160.

(41) Seshadri, A. et al. *ChemRxiv* **2022**.

(42) Sanchez-Lengeling, B. et al. **2020**, *33*, ed. by Larochelle, H. et al., 5898–5910.

(43) McCloskey, K. et al. *Proceedings of the National Academy of Sciences* **2019**, *116*, 11624–11629.

(44) Jiménez-Luna, J.; Skalic, M.; Weskamp, N. *Journal of Chemical Information and Modeling* **2022**, *62*, 274–283.

(45) Oviedo, F. et al. *Accounts of Materials Research* **2022**, *3*, 597–607.

(46) Gandhi, H. A.; White, A. D. *ChemRxiv* **2022**.

(47) Ambegaokar, V.; Eckern, U.; Schön, G. *Physical Review Letters* **1982**, *48*, 1745.

(48) Clary, D. C. *Science* **2008**, *321*, 789–791.

(49) Giese, K. et al. *Physics Reports* **2006**, *430*, 211–276.

(50) Meyer, H. D.; Manthe, U.; Cederbaum, L. S. *Chemical Physics Letters* **1990**, *165*, 73–78.

(51) Manthe, U.; Meyer, H.-D.; Cederbaum, L. S. *The Journal of Chemical Physics* **1992**, *97*, 3199–3213.

(52) Beck, M. *Physics Reports* **2000**, *324*, 1–105.

(53) Eivazi, H. et al. *Physics of Fluids* **2022**, *34*, 075117.

(54) Wen, G. et al. *Energy & Environmental Science* **2023**, *16*, 1732–1741.

(55) Pathak, J. et al. *arXiv:2202.11214 [physics]* **2022**.

(56) Li, Z. et al. *arXiv:2010.08895 [cs, math]* **2021**.

(57) Zhu, W.; Botina, J.; Rabitz, H. *The Journal of Chemical Physics* **1998**, *108*, 1953–1963.

(58) Schröder, M.; Carreón-Macedo, J.-L.; Brown, A. *Physical Chemistry Chemical Physics* **2008**, *10*, 850–856.

(59) Peirce, A. P.; Dahleh, M. A.; Rabitz, H. *Physical Review A* **1988**, *37*, 4950–4964.

(60) Mockus, J., *Bayesian approach to global optimization: theory and applications*; Mathematics and its applications. Soviet series; Kluwer Academic: 1989.

(61) Cotter, S. L. et al. *Statistical Science* **2013**, *28*, 424–446.

(62) Koch, W.; Holthausen, M. C., *A Chemist's Guide to Density Functional Theory*, 1st ed.; Wiley: 2001.

(63) Parr, R. G.; Yang, W., *Density-functional theory of atoms and molecules*, 1st ed.; International series of monographs on chemistry 16; Oxford Univ. Press [u.a.]: New York, NY, 1994.

(64) Gross, E. K. U.; Maitra, N. T., *Introduction to TDDFT*; Marques, M. A. et al., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2012; Vol. 837, pp 53–99.

(65) Fiolhais, C. et al., *A Primer in Density Functional Theory*; Lecture Notes in Physics, Vol. 620; Springer Berlin Heidelberg: Berlin, Heidelberg, 2003.

(66) Hohenberg, P.; Kohn, W. *Physical Review* **1964**, *136*, B864–B871.

(67) Kohn, W.; Sham, L. J. *Physical Review* **1965**, *140*, A1133–A1138.

(68) Mardirossian, N.; Head-Gordon, M. *Molecular Physics* **2017**, *115*, 2315–2372.

(69) Perdew, J. P. *AIP Conference Proceedings* **2001**, *577*, 1–20.

(70) Perdew, J. P.; Burke, K.; Ernzerhof, M. *Physical Review Letters* **1996**, *77*, 3865–3868.

(71) Becke, A. D. *The Journal of Chemical Physics* **1993**, *98*, 5648–5652.

(72) Lee, C.; Yang, W.; Parr, R. G. *Physical Review B* **1988**, *37*, 785–789.

(73) Stephens, P. J. et al. *The Journal of Physical Chemistry* **1994**, *98*, 11623–11627.

(74) Vosko, S. H.; Wilk, L.; Nusair, M. *Canadian Journal of Physics* **1980**, *58*, 1200–1211.

(75) Casida, M. E. *Physical Review A* **1995**, *51*, 2005–2013.

(76) Casida, M.; Huix-Rotllant, M. *Annual Review of Physical Chemistry* **2012**, *63*, 287–323.

(77) Hirata, S.; Head-Gordon, M. *Chemical Physics Letters* **1999**, *314*, 291–299.

(78) Tal-Ezer, H.; Kosloff, R. *The Journal of Chemical Physics* **1984**, *81*, 3967–3971.

(79) Leforestier, C et al. *Journal of Computational Physics* **1991**, *94*, 59–80.

(80) Feit, M.; Fleck, J.; Steiger, A *Journal of Computational Physics* **1982**, *47*, 412–433.

(81) Richings, G. et al. *International Reviews in Physical Chemistry* **2015**, *34*, 269–308.

(82) Achilles, R.; Bonfiglioli, A. *Archive for History of Exact Sciences* **2012**, *66*, 295–358.

(83) Murphy, K. P., *Probabilistic machine learning: an introduction*; Adaptive computation and machine learning series; The MIT Press: Cambridge, Massachusetts, 2022.

(84) Goodfellow, I.; Bengio, Y.; Courville, A., *Deep learning*; Adaptive computation and machine learning; The MIT Press: Cambridge, Massachusetts, 2016.

(85) Ramakrishnan, R. et al. *Scientific Data* **2014**, *1*, 140022.

(86) Himanen, L. et al. *arXiv:1904.08875 [cond-mat.mtrl-sci]* **2019**.

(87) Weininger, D. *Journal of Chemical Information & Computer Sciences* **1988**, *28*, 31–36.

(88) Montavon, G. et al. *Advances in Neural Information Processing Systems* **2012**, *25*.

(89) Bartók, A. P.; Kondor, R.; Csányi, G. *Physical Review B* **2013**, *87*, 184115.

(90) Scarselli, F. et al. *IEEE Transactions on Neural Networks* **2009**, *20*, 61–80.

(91) Gilmer, J. et al. *arXiv:1704.01212 [cs]* **2017**.

(92) David, L. et al. *Journal of Cheminformatics* **2020**, *12*, 56.

(93) Wei, J. N.; Duvenaud, D.; Aspuru-Guzik, A. *ACS Central Science* **2016**, *2*, 725–732.

(94) Casey, A. D. et al. *Journal of Chemical Information and Modeling* **2020**, *60*, 4457–4473.

(95) Ertl, P. et al. *arXiv:1712.07449 [cs, q-bio]* **2018**.

(96) Li, X. et al. *Journal of Cheminformatics* **2020**, *12*, 42.

(97) Hu, D. et al. *The Journal of Physical Chemistry Letters* **2018**, *9*, 2725–2732.

(98) Deringer, V. L. et al. *Chemical Reviews* **2021**, *121*, 10073–10141.

(99) Feurer, M.; Hutter, F. In *Automated Machine Learning: Methods, Systems, Challenges*, Hutter, F., Kotthoff, L., Vanschoren, J., Eds.; The Springer Series on Challenges in Machine Learning; Springer International Publishing: Cham, 2019, pp 3–33.

(100) LaValle, S. M.; Branicky, M. S.; Lindemann, S. R. *The International Journal of Robotics Research* **2004**, *23*, 673–692.

(101) Bergstra, J.; Bengio, Y. *Journal of Machine Learning Research* **2012**, *13*, 281–305.

(102) Snoek, J.; Larochelle, H.; Adams, R. P. *arXiv:1206.2944 [cs, stat]* **2012**.

(103) Singh, K. et al. *Journal of Chemical Theory and Computation* **2022**, *18*, 4408–4417.

(104) Landrum, G. **2020**.

(105) Molnar, C., *Interpretable machine learning: a guide for making Black Box Models interpretable*; Lulu: Morisville, North Carolina, 2019.

(106) Samek, W. et al., *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*; Springer International Publishing: 2019; Vol. 11700.

(107) Gunning, D. et al. *Science Robotics* **2019**, *4*, eaay7120.

(108) Shrikumar, A.; Greenside, P.; Kundaje, A. **2017**, 3145–3153.

(109) Zhou, B. et al. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **2016**, 2921–2929.

(110) Selvaraju, R. R. et al. *2017 IEEE International Conference on Computer Vision (ICCV)* **2017**, 618–626.

(111) Sharma, A. et al. *Briefings in Bioinformatics* **2021**, *22*, 1–12.

(112) Agarwal, C. et al. *Scientific Data* **2023**, *10*, 144.

(113) Kotobi, A. et al. *Journal of the American Chemical Society* **2023**, *145*, 22584–22598.

(114) Bradley, A. P. *Pattern Recognition* **1997**, *30*, 1145–1159.

(115) Dissanayake, M. W. M. G.; Phan-Thien, N. *Communications in Numerical Methods in Engineering* **1994**, *10*, 195–201.

(116) Schaeffer, H. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2017**, *473*, 20160446.

(117) Rudy, S. H. et al. *Science Advances* **2017**, *3*, e1602614.

(118) Lee, K.; Carlberg, K. *arXiv:1812.08373 [cs]* **2019**.

(119) Kutz, J. N., *Dynamic mode decomposition: data-driven modeling of complex systems*; Society for Industrial and Applied Mathematics: Philadelphia, 2016.

(120) Bar-Sinai, Y. et al. *arXiv:1808.04930 [cond-mat, physics:physics]* **2019**, *116*, 15344–15349.

(121) Greenfeld, D. et al. *Proceedings of the 36th International Conference on Machine Learning* **2019**, 2415–2423.

(122) Kochkov, D. et al. *Proceedings of the National Academy of Sciences* **2021**, *118*, e2101784118.

(123) Raissi, M.; Perdikaris, P.; Karniadakis, G. E. *Journal of Computational Physics* **2019**, *378*, 686–707.

(124) Lu, L. et al. *Nature Machine Intelligence* **2021**, *3*, 218–229.

(125) Li, Z. et al. *arXiv:2003.03485 [cs, math, stat]* **2020**.

(126) Chen, T.; Chen, H. *IEEE Transactions on Neural Networks* **1995**, *6*, 911–917.

(127) Hornik, K.; Stinchcombe, M.; White, H. *Neural Networks* **1989**, *2*, 359–366.

(128) Rosofsky, S. G.; Majed, H. A.; Huerta, E. A. *Machine Learning: Science and Technology* **2023**, *4*, 025022.

(129) Thomas, E. F.; Henriksen, N. E. *Physical Review A* **2019**, *99*, 023422.

(130) Brooks, S. *Journal of the Royal Statistical Society: Series D (The Statistician)* **1998**, *47*, 69–100.

(131) Jensen, H. A.; Jerez, D. J.; Valdebenito, M. *Mechanical Systems and Signal Processing* **2020**, *143*, 106836.

(132) Robert, C. *arXiv:1504.01896 [stat]* **2016**.

(133) Casella, G.; George, E. I. *The American Statistician* **1992**, *46*, 167–174.

(134) Parzen, E. *The Annals of Mathematical Statistics* **1962**, *33*, 1065–1076.

(135) Klein, B. P.; Hall, S. J.; Maurer, R. J. *Journal of Physics: Condensed Matter* **2021**, *33*, 154005.

(136) Tkatchenko, A.; Scheffler, M. *Physical Review Letters* **2009**, *102*, 073005.

(137) Blum, V. et al. *Computer Physics Communications* **2009**, *180*, 2175–2196.

(138) Kipf, T. N.; Welling, M. *arXiv:1609.02907 [cs, stat]* **2017**.

(139) Schütt, K. T. et al. *arXiv:1706.08566 [physics, stat]* **2017**.

(140) Xu, K. et al. *arXiv:1810.00826 [cs, stat]* **2019**.

(141) Schütt, K. T. et al. *Nature Communications* **2017**, *8*, 13890.

(142) Fey, M.; Lenssen, J. E. *arXiv:1903.02428 [cs, stat]* **2019**.

(143) Rogers, D.; Hahn, M. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754.

(144) Probst, D.; Reymond, J.-L. *Journal of Cheminformatics* **2020**, *12*, 12.

(145) Neese, F. et al. *The Journal of Chemical Physics* **2020**, *152*, 224108.

(146) Veličković, P. et al. *International Conference on Learning Representations* **2017**.

(147) Vaswani, A. et al. *Advances in Neural Information Processing Systems* **2017**, *30*.

(148) Battaglia, P. W. et al. *arXiv preprint arXiv:1806.01261* **2018**.

(149) Tesch, C. M.; Kompa, K.-L.; Vivie-Riedle, R. d. *Chemical Physics* **2001**, *267*, 173–185.

(150) Carreira, L. A.; Mills, I. M.; Person, W. B. *The Journal of Chemical Physics* **1972**, *56*, 1444–1448.

(151) Conte, R. et al. *The Journal of Chemical Physics* **2019**, *151*, 214107.

(152) Thomas, E. F.; Henriksen, N. E. *The Journal of Physical Chemistry Letters* **2017**, *8*, 2212–2219.

(153) Madsen, C. B. et al. *The Journal of Chemical Physics* **2009**, *130*, 234310.

(154) Marston, C. C.; Balint-Kurti, G. G. *The Journal of Chemical Physics* **1989**, *91*, 3571–3576.

(155) Son, H. et al. *Communications in Mathematical Sciences* **2023**, *21*, 1679–1705.

(156) Czarnecki, W. M. et al. *arXiv:1706.04859 [cs]* **2017**.

Appendix A

Publications

A.1 Graph Neural Networks for learning molecular excitation spectra

Kanishka Singh, Jannes Münchmeyer, Leon Weber, Ulf Leser, Annika Bande

J. Chem. Theory Comput. 2022, 18, 7, 4408–4417
DOI: 10.1021/acs.jctc.2c00255
URL: https://doi.org/10.1021/acs.jctc.2c00255

Contributions: The project was conceptualized primarily by AB and KS. KS implemented the Python code for the project in PyTorch Geometric. JM and LW contributed extensively to the debugging of the code, and guiding the implementation of the hyperparameter process. UL and AB were instrumental in the formulation and ideation of various analysis techniques beyond performative comparison of the various GNN models. KS contributed to writing the main text of the manuscript. AB and UL contributed extensively to corrections and feedback, as well as writing some sections of the text. UW and JM helped streamline the text and contributed to certain figures.

A.2 Integrating Explainability into Graph Neural Network Models for the Prediction of X-ray Absorption Spectra

Amir Kotobi, **Kanishka Singh**, Daniel Höche, Sadia Bari, Robert Meißner, Annika Bande

J. Am. Chem. Soc. 2023, 145, 41, 22584–22598
DOI: 10.1021/jacs.3c07513
URL: https://doi.org/10.1021/jacs.3c07513

Contributions: The idea of using GNNs to predict XAS was conceived as part of a collaboration involving KS, AB, AK, RM, and DH. In line with this idea, AK created the QM9-XAS dataset which was used to train and test various GNNs. KS had previously used CAMs to gain insight into ML-predicted spectra in a previous undocumented work. The results of this exercise helped KS and AB propose investigating the interpretability of GNNs used for XAS prediction, which became the core idea of this work. While AK was the main developer of the code for this project, KS also helped with the implementation of GNNs and the subsequent analysis in Python. KS contributed extensively to the ideation and development of the ground truth benchmark from quantum chemical data and deriving a metric for comparison between CAMs and the ground truth. KS also helped design and implement the tasks on perturbation and visualization of the explanations in the manuscript. AK and KS were the main contributors to the manuscript. AB regularly supervised AK and KS and provided ideas that helped move the project forward. AB and RM were instrumental in making extensive corrections to the manuscript. SB provided valuable input that helped in the understanding of the interpretations of XAS, and DH provided valuable feedback on understanding the impact of the effect of input feature perturbations on interpretability during the writing and review phase of the manuscript.

JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/JACS

Article

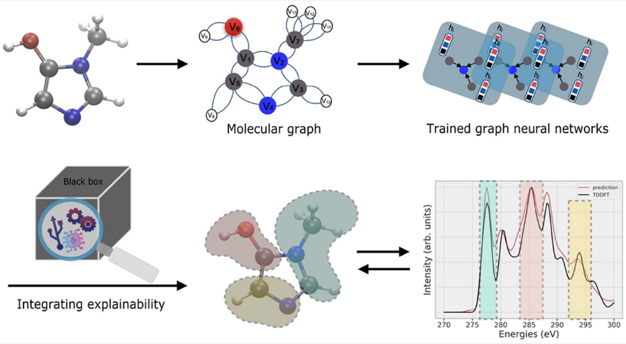
Integrating Explainability into Graph Neural Network Models for the Prediction of X-ray Absorption Spectra

Amir Kotobi, Kanishka Singh, Daniel Höche, Sadia Bari, Robert H. Meißner, and Annika Bande*

Cite This: *J. Am. Chem. Soc.* 2023, 145, 22584−22598

Read Online

ACCESS | Metrics & More | Article Recommendations



ABSTRACT: The use of sophisticated machine learning (ML) models, such as graph neural networks (GNNs), to predict complex molecular properties or all kinds of spectra has grown rapidly. However, ensuring the interpretability of these models' predictions remains a challenge. For example, a rigorous understanding of the predicted X-ray absorption spectrum (XAS) generated by such ML models requires an in-depth investigation of the respective black-box ML model used. Here, this is done for different GNNs based on a comprehensive, custom-generated XAS data set for small organic molecules. We show that a thorough analysis of the different ML models with respect to the local and global environments considered in each ML model is essential for the selection of an appropriate ML model that allows a robust XAS prediction. Moreover, we employ feature attribution to determine the respective contributions of various atoms in the molecules to the peaks observed in the XAS spectrum. By comparing this peak assignment to the core and virtual orbitals from the quantum chemical calculations underlying our data set, we demonstrate that it is possible to relate the atomic contributions via these orbitals to the XAS spectrum.

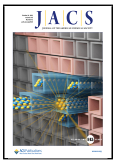
■ INTRODUCTION

X-ray absorption spectroscopy (XAS) is an important characterization technique in chemical analysis to unveil the atomic structure of matter, having a broad range of applications in material science,[1] biomedical research,[2] and identification of metals and solids.[3] XAS is particularly useful in the investigation of the electronic and geometric structure of biomolecules, nanoparticles, and metal complexes.[4−6] The interpretation of experimentally obtained XAS spectra is, however, complicated due to the intricate interplay between the complex electronic structure of the material and the adsorption of X-ray photons. Several factors, including the chemical environment of the atom, the presence of solvents, and the energy of the incident X-rays, influence this complexity.[7] Therefore, sophisticated—but computationally also expensive—theoretical methods from *ab initio* quantum chemistry can accurately predict XAS and are a necessary complement to interpret experimental results.[8]

Machine learning (ML) techniques are being increasingly applied to various areas of theoretical and computational chemistry given their ability to infer structure−property relationships on the basis of large amounts of data.[9−11] Among those ML techniques, graph neural networks (GNN) and deep neural networks (DNN) are promising candidates to predict the properties of matter, such as the electronic structure,[12] at a higher computational speed, already making them favorable for high-throughput calculations in materials design and drug discovery.[13,14] Thus, the ability to perform efficient computations with high accuracy has demonstrated that ML techniques are advantageous in domains such as various types of spectroscopy, including vibrational and optical.[12,15−24]

Several studies have focused on X-ray spectroscopy using ML methods with the additional aim to improve the understanding of the contribution of different atomic environments to the peaks occurring in the spectra.[22−24] Accurate prediction of XAS spectra has been accomplished by employing some of the more sophisticated ML models, such as GNNs and DNNs.[17,25,26] However, a large number of layers in the underlying neural network, as well as a high parameter count, implies such models are black-box,[27] which means understanding the rationale behind predictions is a challenging task. On the other hand, ML models designed to predict XAS spectra must provide clear peak assignments, as this option for

Received: July 17, 2023
Published: October 9, 2023



https://doi.org/10.1021/jacs.3c07513

interpretation is typically required in spectroscopy experiments and often necessitates theoretical calculations. The comprehensibility of why ML models can achieve this peak assignment capability must be transparent to users to ensure trust in the predictions, given the diverse range of applications of XAS in material and biochemical sciences.[7,28,29] It is therefore imperative to develop an understanding of the XAS predictions made by complex ML models and ascertain whether the predictions align with human logic and decision-making, as incorporated in the quantum-mechanical equations. This can be achieved using explainable artificial intelligence (XAI) methods, which provide a window into the ML model's decision-making process and correlations uncovered by the model through data analysis.[30] Justification and interpretability offered by XAI methods not only provide evidence defending why a prediction is trustworthy with quantitative metrics but also refer to the degree of human understanding intrinsic within the model.[10,31,32]

Numerous techniques are available to incorporate explainability in GNN and DNN models.[33,34] Our emphasis in this work lies in using a method known as attribution.[35] Attribution methods have found widespread use in applications where the input data consists of images or text, composed of features such as pixels, characters, and words.[36,37] In these cases, attribution scores highlight particular regions via pixels of the image or certain characters or words in a text that affect, in this case, the decision-making of the ML model used in the task. Therefore, it is relatively easy to validate such explanations in image- or text-based tasks. However, validating explanations for chemical property prediction is challenging since a property is often the result of a complex interplay between the geometric and electronic structure of the atoms in a molecule. This gives rise to intricate structure−property connections within molecules, especially complex properties such as X-ray absorption spectra, which only find interpretation by the examination of each individual peak detected through a combination of experiments and simulations.[21] Therefore, the validation of explanations generated using attributions also requires the creation of a robust "ground-truth" benchmark using such domain-specific knowledge, which is often a challenging task in molecular-property prediction. Examining the robustness of GNN models to predictions on unseen data, being possibly biased toward specific chemical structures, is yet another challenge in understanding the overall performance of different models.[38,39]

In this study, we introduce a framework that uses a combination of graph attributions and ground-truth data generated from linear-response time-dependent density functional theory (TDDFT),[40] to provide explainability on GNN models trained to predict carbon K-edge XAS spectra of organic molecules. Carbon K-edge spectroscopy was used for XAS for various reasons. First, carbon atoms play a central role in the structure and function of a wide range of organic molecules as well as inorganic materials. Second, carbon K-edge XAS offers a unique perspective, providing valuable insights into the structure, function, and reactivity of these molecules.[41,42] Finally, among the XAS calculations, K-edge spectroscopy on a main group element is less complicated than, for example, the spectroscopy on the transition metal L edge, and can be computed via TDDFT on a time scale that allows the creation of a large data set. To train the different GNN architectures, an in-house QM9-XAS data set, based on a subset of the QM9 data set or organic molecules,[43] was set up (see Data Availability Statement). We compare the performance of the trained models in predicting XAS spectra on the test data set. In order to evaluate the explainability of GNN models, we analyze the ability of these models to identify the contribution of atoms and their surrounding environment toward the distinct peaks in the XAS spectrum. For creating the "chemical" ground truth pertaining to XAS, we created a data pipeline that inputs the output of TDDFT calculations and renders the labels to atoms, indicating whether or not an atom contributes to a specific excited state in XAS. These ground-truth values are then finally quantitatively compared with the attribution scores obtained from GNNs. Applying this method to different GNN models, we find that specific GNN architectures, which incorporate both global and local information on atoms, offer superior explanations for the peaks observed in the carbon K-edge XAS spectra. Additionally, we investigate the robustness of the GNN models by randomly perturbing molecules in the test data set, to rationalize the difference in the explainability power of various used GNN architectures.

■ METHODS

The QM9-XAS Data Set. While X-ray absorption spectroscopy is a popular technique in chemistry, to the best of our knowledge there is no organic molecule XAS data set that is large enough and available for training ML models. Therefore, we used the QM9 data set[43] containing 132,531 organic molecules composed of the first and second row of main group elements H, C, N, O, and F. We choose a random subset of the QM9 data set, containing 56,000 molecules, which we term QM9-XAS for the purpose of our data set. We use these structures to calculate carbon K-edge XAS spectra with the time-dependent density functional theory (TDDFT)[44] method, which is in general a useful complement to experiments and allows for the interpretation of spectral peaks. More specifically we used the ORCA electronic structure package[45] to calculate TDDFT at the B3LYP/TZVP[46,47] level of theory. All calculated XAS spectra were obtained in the energy ranges E_{min} = 270 eV and E_{max} = 300 eV and peaks broadened using Gaussians of widths 0.8 eV. The resulting curves were discretized into N_{grid} = 100 points between. This step ensures that the length of the target output to be learned for ML applications is consistent across all spectra. Further processing is then performed to generate tuples of molecular graphs and their spectra to convert them into a format optimal for training GNN models. Molecular graphs were generated from the SMILES strings of the molecules, which were available in the original QM9 data set using the RDKit[48] python library. Since our models are implemented using the Pytorch Geometric[49] library, the graph and spectrum tuples were converted into the native data set class of this library.

Graph Neural Networks. GNNs are neural networks specifically designed to treat unstructured molecular data.[50] A graph is formally defined as a tuple of $G = (V, E)$ of a set of nodes $v \in V$ and a set of edges $e_{v,w} = (v, w) \in E$, which defines the connection between nodes. It is intuitive to represent molecules as graphs, in which atoms and the bonds between them are represented as nodes and edges, respectively. Further information about each atom and bond in a molecular graph is incorporated in the form of node and edge feature vectors added to the tuple G of each graph in the data set. A node (atomic) feature vector represents information such as the atom type (e.g., C, H, N, O, or F) or the number of hydrogen atoms attached to it. Similarly, edge (bond) feature vectors are representatives of properties such as the bond lengths between two atoms or the bond multiplicity. We employ one-hot encoding to convert most of the node and edge features, including categorical attributes, such as atom type, into numeric vectors. All encodings used in this work are summarized in Table 1.

A GNN layer takes as input a graph with node and edge features and outputs a graph with the same topology where the node, edge, and global graph information is updated. To achieve this, the node

Table 1. Features of Nodes and Edges (Atoms and Bonds) as Represented in the Encoded Vector in Conjunction with Their Respective Type of Encoding

Node feature	Encoding
Atomic number	One hot
Hybridization	One hot
Aromaticity	One hot
Number of H atoms	Integer
Edge feature	**Encoding**
Bond distance	Real
Bond type	One hot

and edge information represented as feature vectors are first converted into vectors in higher dimensional space (feature space) referred to as node and edge states, respectively, using a transformation function. Transformation functions can be fully connected layers, convolutional layers, or recurrent layers depending on the GNN architecture. A fundamental part of GNNs is the so-called propagation (or message-passing) process used to update these node (or edge) states. Message passing occurs in two steps: The first step involves gathering the information on the nodes (or edges) surrounding a target node by collecting their node states. In the second step, these states, along with the state of the target node, are aggregated using an aggregation function such as sum or average. If the final task is to predict the property of a graph, then these updated node states are further aggregated using a graph-level aggregation function, termed readout.

Different GNN architectures have different message propagation and readout functions that affect the node, edge, and graph states obtained at the end of a message-passing process. In this work, we trained ML models on three GNN algorithms. The first architecture is the graph convolutional neural network (GCN),[51] which employs only node states to aggregate information in the message-passing process. The second GNN model is GraphNet,[52] in which a global state vector, including node and edge states, is used in the message function. The third is the multihead graph attention network (GATv2)[53] in which an attention mechanism[54] is used to aggregate node information. The attention mechanism in GATv2 allows for the calculation of edge weights to each node in the neighborhood of a target node, which assigns an importance value to the message passed from each node to the target node. Training a multihead GATv2 converges faster at a moderately higher computational cost, while also increasing the robustness of the final model since it is in principle trained on multiple attention instances in parallel.

Training. In order to assess various trained models, the QM9-XAS data set was shuffled and divided into a training set of 50k samples and a test set of 6k samples, with the training data further partitioned into an 80:20 ratio for training and validation. The GNNs and all fully connected layers were trained for 1000 epochs, at a learning rate of 1 × 10^{-3}, and a batch size of 100 samples. A learning rate scheduler was implemented to reduce the learning rate by a factor of 0.8 every 100 epochs. For all the models, three GNN hidden layers with sizes of 128, 256, and 512 were used for node updates, and a fully connected layer was used as the output layer for predictions. We used the AdamW optimizer[55] and the root mean squared error (RMSE) as the loss function to train the models. In order to keep track of overfitting, we monitor the RMSE loss on the validation set after every 50 epochs. All models were trained on a single NVIDIA Tesla A100 64GB GPU. We select the model which has the best RMSE loss and relative spectral error (RSE)[19] on the validation data set.

RSE is obtained by dividing the RMSE among the target y^{tar} and the predicted y^{pred} intensities of the signal at energy E, by the total spectral energy of the target. In the discretized spectrum in steps of $\Delta E = (E_{\max} - E_{\min})/N_{\text{grid}}$, the RSE is approximated as

$$\text{RSE} = \frac{\sqrt{\sum_i^{N_{\text{grid}}} (y_i^{\text{tar}} - y_i^{\text{pred}})^2 \cdot \Delta E}}{\sum_i^{N_{\text{grid}}} y_i^{\text{tar}} \cdot \Delta E} \quad (1)$$

A small relative spectral error indicates that the predicted spectrum is a good prediction of the original spectrum. The quality of XAS spectra predictions made by different GNN architectures was compared by calculating the average RSE on the test data set.

Graph Attribution. Attributions or feature attributions are one of the most popular techniques used to explain the model's predictions.[56] The attribution method assigns scores to each input feature that reflects the contribution of that feature to an ML model's prediction, thereby explaining the role played by that feature in the prediction.[38,57,58] In the case of GNNs, attribution methods assign attribution scores to graph nodes and edges based on their contributions to the final prediction of the model. One way to visualize the attribution scores obtained is by overlaying a heat map on top of a graph, highlighting the importance of individual atoms to the target property in the case of a molecular graph. From these heatmaps, one can deduce structural correlations between the model's rationale for good or bad predictions and compare them to existing knowledge of why the prediction should be so. GradInput (GI),[59] class activation map (CAM),[60] and gradient class activation map (GradCAM)[61] have been shown to successfully explain predictions made by GNNs for molecular structure−property prediction models;[35] that is, they can reveal the contribution of individual atoms or atom pairs to the model's decision. Although GNNs and their interpretation through attribution techniques have proven successful in decoding binding mechanisms and performing materials discovery,[38,58,62] to the best of our knowledge, these explainability techniques have not been employed in XAS analysis. The scoring attribution of atoms arising from CAM is intuitively well-suited to the phenomenon of XAS, where peaks in the spectrum arise from the local and global environments of atoms in a molecule.[63,64]

We, therefore, use CAM to obtain atomic contributions to the XAS spectra of molecules in the QM9-XAS data set to explain the spectrum predictions of GNN models. CAM attributions calculate the node weights v_i for highlighting the contribution of various nodes of the graph to the prediction. As discussed above, GNNs that perform property prediction on graphs use a global aggregation layer or a readout layer prior to the output layer. In our case, the model generates 100 values for the final spectrum by utilizing a layer consisting of 100 units (neurons). For the purpose of evaluating the attributions, each of these values can be treated as an independent class. CAM operates on the aggregation layer prior to this final layer and obtains attributions for these different "classes", giving an insight into atomic contributions at each point in the spectrum. To compute CAM weights of a node for each class, let $F_{k(i)}$ be the activation of a unit k in the last GNN convolutional layer, preceding the output layer, at node i. The CAM score at a node for a class c then is defined as[65,66]

$$v_{c(i)} = \sum_k \omega_k^c F_{k(i)} \quad (2)$$

where ω_k^c denotes the weight of unit k for class c. Using this formulation, one can obtain CAM scores for each point in the spectrum of a given input molecular graph.

Ground-Truth Evaluation. In addition to the evaluation of attributions, it is crucial to establish a ground-truth logic that enables the assessment of attribution quality. Hence, the agreement between CAM weights of the model's prediction and ground-truth logic should be quantified. To this end, a definition for a numerically measurable ground truth for the excitations underlying the spectra is needed. In other instances of XAI in chemistry, a suitable ground truth was developed by directly considering the molecular fragments or functional moieties that experts knew to be important for decision making,[67] such as binding mechanism learned by DNNs.[38] Nevertheless, when it comes to predicting XAS, comparing attribution scores to ground truth becomes more complex, since it necessitates careful examination of all atoms in the molecule and a comprehensive understanding of the quantum mechanics behind X-ray excitations. Furthermore, delocalized molecular orbitals present yet another challenge for understanding the precise contribution of atoms to virtual orbitals in excitation states of XAS.[68] Therefore, we have developed a method that assigns the ground-truth contributions of

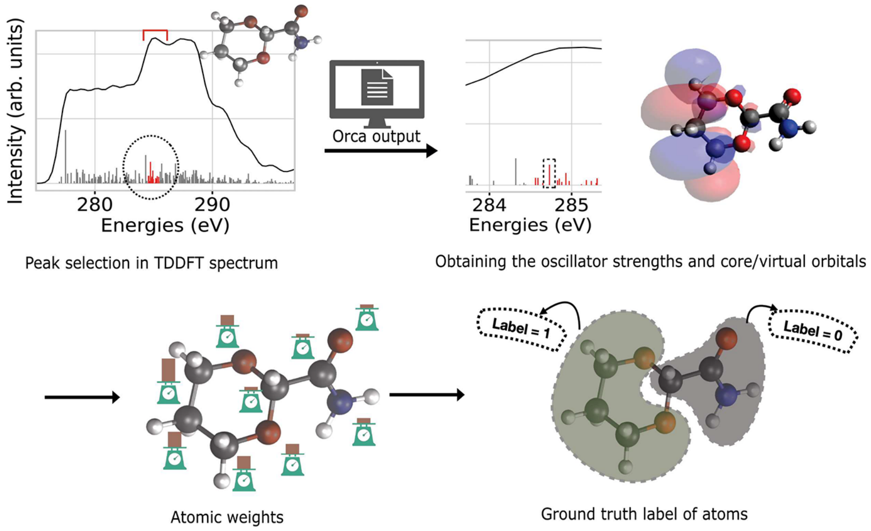

Intensity (arb. units)
280
290
Energies (eV)
Peak selection in TDDFT spectrum
Orca output
284
285
Energies (eV)
Obtaining the oscillator strengths and core/virtual orbitals
Atomic weights
Label = 1
Label = 0
Ground truth label of atoms

Figure 1. Ground-truth evaluation is based on TDDFT data. The process of evaluating the TDDFT data starts with selecting a specific peak in the XAS spectrum. The oscillator strength and orbital contributions for each excitation state in the peak are used to determine the final atomic contributions to the peak. Atoms in the molecule are then labeled based on the calculated weights, i.e., 1 for atoms contributing to the peak and 0 otherwise.

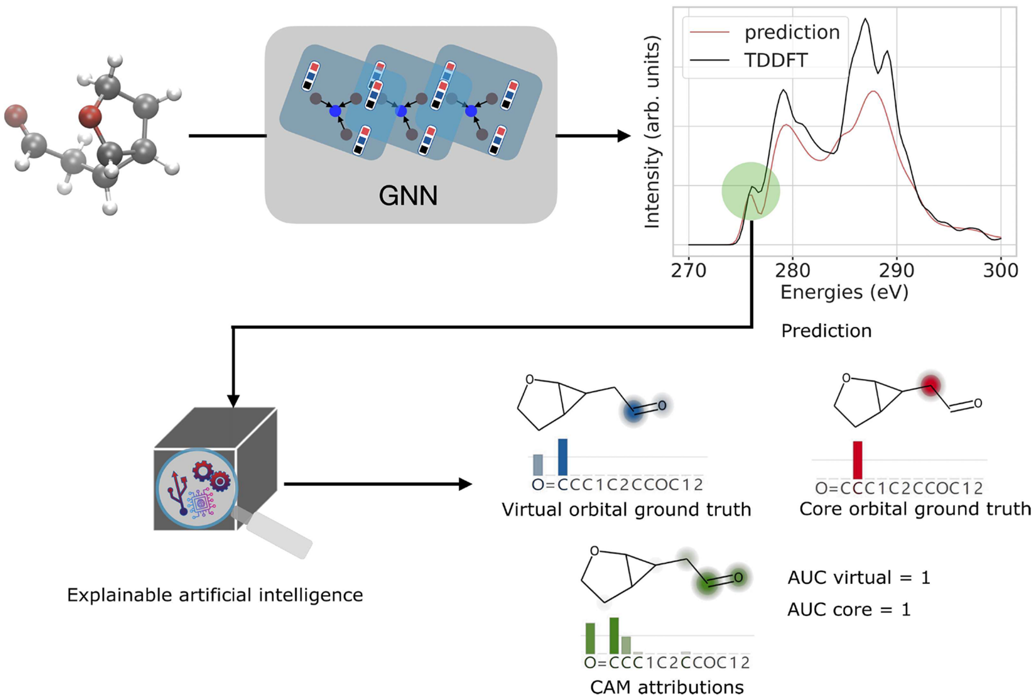


Figure 2. Workflow of the ML and explainability of the XAS spectrum. This process consists of converting a molecule to a molecular graph, training a GNN, comparison of the ML predicted and TDDFT spectra for obtaining the RSE, and finally applying the XAI technique to obtain here the CAM weights (green). In this example, the CAM weights are compared to ground-truth attributions for core (red) and virtual (blue) orbitals at the highlighted 277 eV peak of the spectrum, using a heatmap[70] on the molecular structure. These ground-truth labels are then compared to CAM weights, giving the AUC values for the core and virtual contributions.

various atoms in a molecule to a peak in the TDDFT spectrum. It uses a combination of orbital populations of all of the initial and final states underlying the respective X-ray excitations and their oscillator strengths to obtain the contribution of each atom to a specific peak in the XAS spectrum. To derive atomic contributions in the ground truth, we first compute the core excitations within this energy range and then determine the atoms contributing to both the core and virtual orbitals of a certain excitation state. The atom contributions were weighted according to the oscillator strength of the corresponding excited state as well as the atom population per molecular orbital. In cases where the calculated weights in the ground truth necessitate the presence of particular atoms in a peak of the XAS spectrum, we label those atoms as 1 and all other atoms as 0. Figure 1 depicts the process of obtaining ground-truth labels for atoms. Given the fact that the optical transitions obtained from TDDFT are discrete lines and that the ML spectra are distributed on a grid and have wide peaks, for the comparison it is necessary to unify all CAM scores of a given peak to a given line from TDDFT spectrum. Hence, we summed up the CAM scores of all atoms in the molecule for all

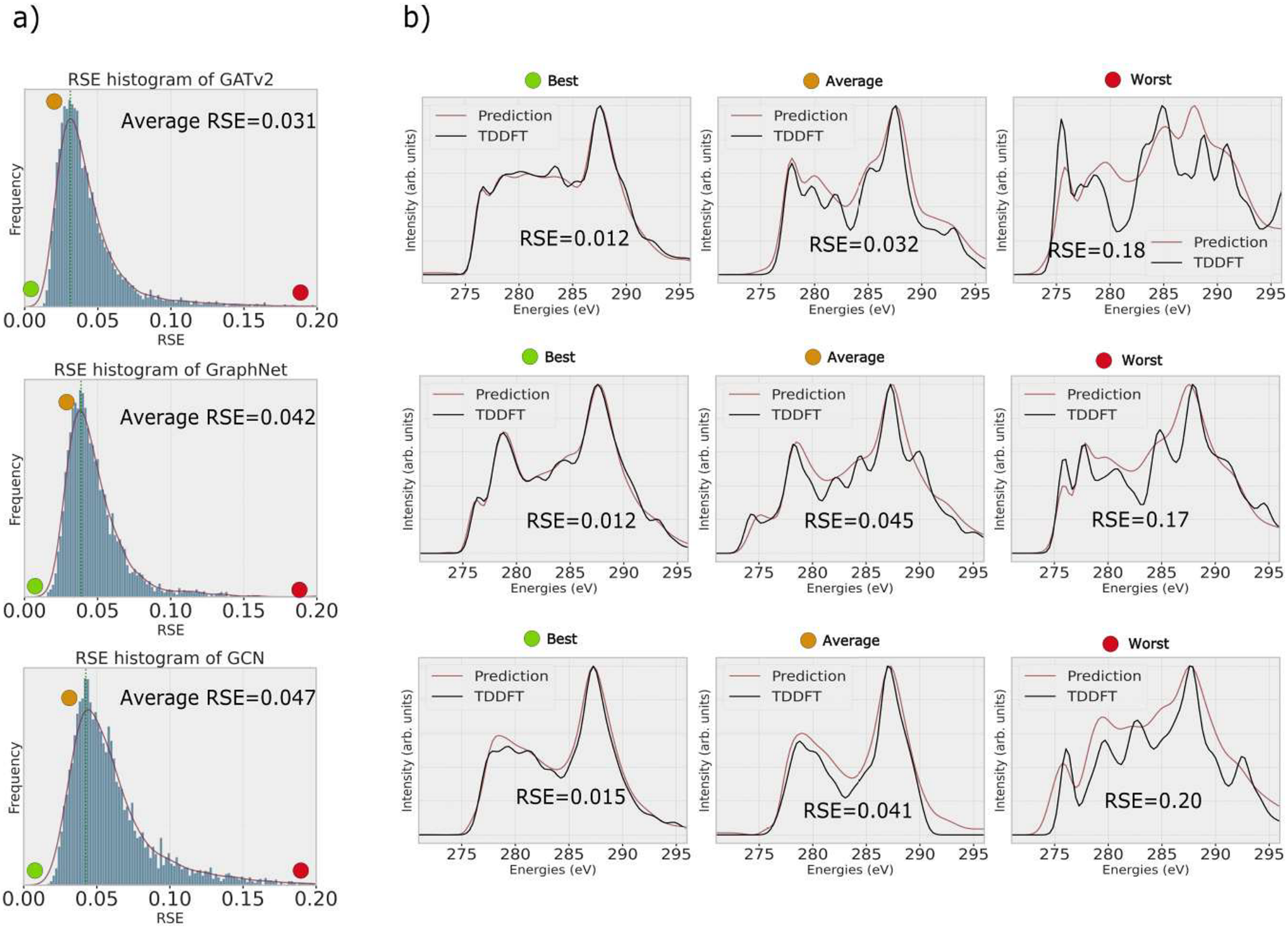


Figure 3. Evaluating the performance of various GNNs on the test data set. RSE histogram for all GNN models (a). While average RSE performances are close, GATv2 has a more left-skewed histogram distribution, indicating better performance over large portions of the data. Best, worst, and average predictions of the three GNN models with their respective RSE values (b).

energy points in a range equivalent to the full width at half-maximum of a peak.

Model Explainability. Explaining a model's predictions involves comparing the ground truth to the attributions obtained from the model by using an XAI method. To measure to what extent our ML models learn the correct atomic contributions to the XAS spectra, we use the area under the curve (AUC) of the receiver operating characteristic (ROC).[38,69] The ROC itself is a curve formed by plotting the rate of true positive outcomes and that of false positive ones at various classification thresholds that divide the assignments between the true and false classes. A true positive outcome occurs when a model tasked with distinguishing two or more classes correctly predicts the class to which an instance belongs. In our case, the CAM weight assigned to an atom at a certain peak matches the ground truth of the atoms belonging to an orbital. Similarly, a false positive occurs when the class under investigation is incorrectly predicted by the model, i.e., when atom contribution in ground truth and CAM disagree. The AUC thus quantifies the performance of a classification model into a single value between 0 and 1, where an area of 0.5 means that a model works only as good as a random classifier. A value of 1.0 means that the model has the ability to perfectly discriminate among different classes. In this particular case, the AUC is indicative of whether the model can correctly identify whether an atom contributes to a peak in the spectrum or not.

Figure 2 illustrates the workflow to make a GNN prediction of a spectrum, determine the CAM attribution, and compare it in the last step to the ground truth, i.e. the contribution of atoms to core and virtual orbitals obtained from TDDFT, here shown for a prediction made by the multihead GATv2 model. More explicitly, a model with a large AUC close to 1.0 would perfectly assign labels 0 and 1 to each atom in the spectrum for all of the molecules in the test set. Moreover, we identify the baseline of AUC as 0.5, which is basically a model classifier that randomly assigns these labels to the atoms in a molecule.

We compute attribution AUC values at each peak in a TDDFT spectrum and average them over all of the peaks to arrive at a final score that explains the degree of agreement between ground-truth logic and CAM attribution scores. The AUC is determined for different model architectures. To demonstrate that the explainability method is stable, we perturb a randomly chosen set of molecules from the test data set and evaluate the change in attribution AUC.

RESULTS AND DISCUSSION

Model Performance. To first visualize the predictions made by these GNN models, the best, average, and worst predictions of the XAS spectrum are demonstrated for each model based on RSE values in Figure 3b. While the best prediction across all models is a near-perfect replica of the TDDFT spectrum, the average and worse ones predict general features of the spectrum correctly, but miss out on the finer peak structure or incorrectly predict peak intensities. In Figure 3a, all RSE values for one model are plotted in a histogram and the average RSE is determined. The GATv2 model has a slightly lower average RSE value of 0.031 compared to 0.042

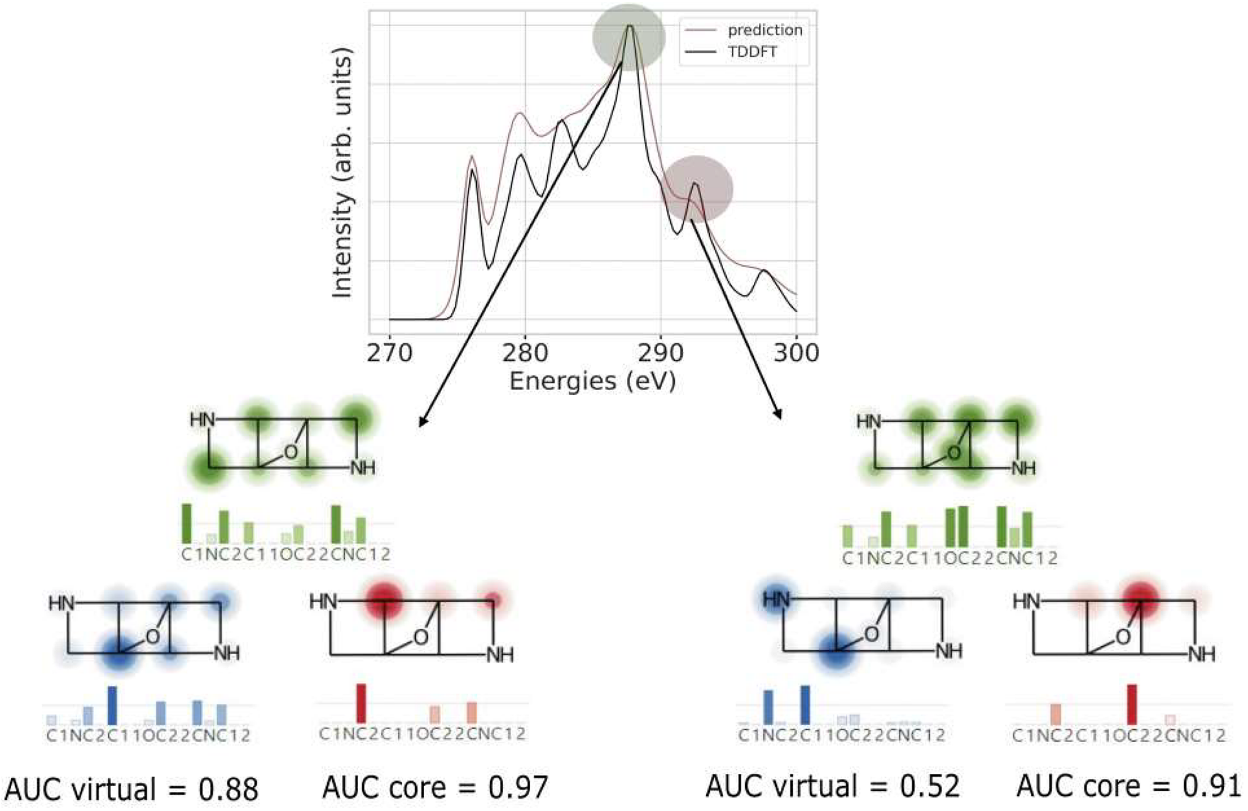


Figure 4. Attributions (green) are compared with the ground truth of core (red) and virtual (blue) orbitals via AUC values for two peaks of an XAS spectrum predicted by the GATv2 model. The model has higher AUC values when a peak in the predicted spectrum follows the TDDFT result.

for GraphNet and 0.047 for GCN. The distributions look similar. They have their onset with a small slope at RSE = 0.0 and then quickly grow to their maximum around the average RSE. The decline is slow following the shape of a skewed distribution with a long tail leading to a low number of structures with RSE values above 0.1. Such structures are fewer for models with the GATv2 and the GraphNet GNN architectures, demonstrating their superiority for XAS predictions compared to GCN.

The above results are consistent with the findings of earlier research, which suggest that integrating an attention mechanism[54] and applying combinatorial generalization,[52] i.e. enabling the network to reason about the global structure of a graph, while learning the graph representation, as done in the GATv2 and GraphNet models, help enhance the learning of target properties related to both local and global structures of the graph.[71,72] In the case of the GATv2 model, computing the importance of the neighboring atoms for a target atom in a molecule using the weighted attention mechanism assigns relevance to a local region of the molecule to a specific excitation energy in the spectrum, which differs from that of traditional GCN layers with fixed weights for connections between atoms. On the other hand, by incorporating relationships and interactions among nodes, edges, and global graph attributes, GraphNet significantly improves the acquisition of structure-properties relationships in XAS spectra.[52]

Explainability of XAS Predictions. While comparing the prediction performance of different ML models is crucial, the similarity observed in the RSE distributions in the previous section motivates exploration of the interpretability of these models. Figure 4 illustrates the peak assignment via core and virtual orbitals from the TD-DFT calculation as red and blue spheres on participating atoms and via the CAM scores given as green spheres. The AUC values for the respective orbitals quantify this assignment. We compare an accurate GATv2 prediction at about 288 eV, in which the intensities of both curves lie on top of each other, with one with a larger deviation from the TD-DFT data at about 292 eV. In both cases, the core orbitals are accurately matched by the CAM score giving AUC values of above 0.9, significant quality differences occur for the virtual orbitals. Those contribute the most to an XAS spectrum in general. Hence, a good prediction comes with a good assignment of the peak with a large AUC of 0.88 eV. By contrast, the poorer XAS prediction with about 10% peak intensity differences also leads to a much reduced AUC of 0.52 only. In this case, one can already visually see that the CAM is much more significantly spread over the entire molecule, while the orbitals contributing are based only on two atoms, of which one is not a part of the CAM at all. Figure 5 gives a close-up visualization of the derivation of the CAM and the core and virtual orbital ground truth, by relating both to local excitations and the latter also to orbitals relevant to the respective excitation. This is done for the first three excitation states of the TDDFT calculation underlying the first signal of the broadened spectrum. Note that later signals are composed of a much larger number of transitions, making the visual comparison very cumbersome. We observe that the first two peaks originate from a transition of an electron on the cyano carbon atom to one of the π^* orbitals of the CN group. This is exactly reflected in the CAM weights obtained at exactly the transition energy. The CAM weights show a low contribution at other atoms, which is insignificant. The third peak belongs

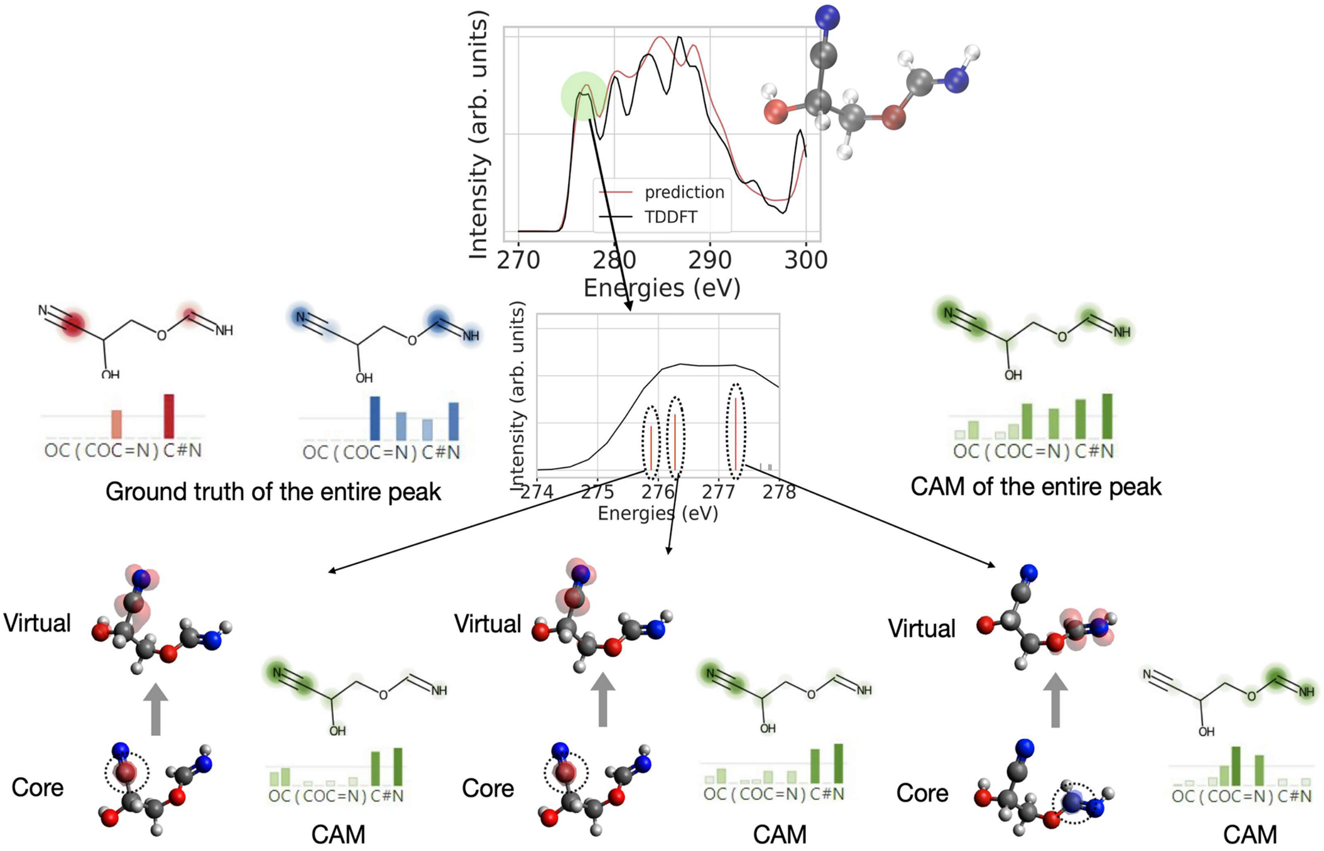


Figure 5. Exploring the correlation between CAM attributions of atoms and transition densities of a peak in the XAS spectrum. CAM attributions (green) and transition densities of three excitation states are visualized for a sample molecule in the test data set in the bottom part of the figure. The transition densities highlight the starting C core orbital, which is encircled for better visibility, in the bottom, and above the virtual orbital on the cyanide group for the two lower-energy peaks and the amide group for the third peak. The overlay of the three transition densities for the core (red) and the virtual (blue) states are shown on the left side of the close-up spectrum, while on the right side, the CAM of the entire peak is shown.

to the $s \rightarrow \pi^*$ transition on the amide group at the other end of the molecules, which is likewise highlighted by the local CAM. The total CAM overlays both transitions, and likewise does the ground truth of the contributing core (red) and virtual (blue) orbitals highlight the two C atoms or multiple bonds, respectively.

To further analyze the explainability of our best GNN model (i.e., GATv2), we performed TDDFT calculation of local atom XAS spectra of individual carbon atoms of a sample molecule in the test data set with the CAM attribution weights assigned to these carbon atoms for which the comparison is displayed in Figure 6. The CAM attribution weights, which are energy-dependent and hence appear as spectra in themselves, exhibit a reasonably accurate alignment with the main features of localized XAS spectra, although they do not entirely replicate all the peaks. In particular, CAM attribution weights of the carbon C1 next to the hydroxy group appear to show discrepancies, which can be due to the attribution technique or weaknesses in the model's explainability concerning this specific atom. Although training a GNN model using localized XAS spectra to predict the spectra of individual carbon atoms is achievable and could potentially enhance the alignment between TDDFT and ML in terms of spectral shape and CAM attribution, generating a data set with atom-localized spectra through various methods requires more computational resources. CAM attributions of atoms from a complete molecular spectrum can provide an opportunity for creating a data set of localized spectra based on arbitrary XAS methods. Moreover, since the ultimate goal is to compare the predicted XAS to experimental spectra, training a model based on entire XAS spectra in certain energy ranges is more favorable.

With this rationalization, the next step is to evaluate the attribution quality overall over the entire data set. Figure 7 shows box plots of the attribution AUC for core and virtual orbitals of the three GNNs evaluated over the full test data set. As seen from the figure, the GCN model gives an average attribution AUC close to 0.5, which means that the model barely outperforms a random classifier. This combination of good spectra predictions on the test data, as shown in Figure 3, and low average attribution AUC value by the GCN model is in line with a previous study, suggesting that the combination of near-perfect model performance and low attribution AUC indicates that the model fails to learn the ground-truth logic.[38] In contrast to this, the GNN models with multihead GATv2 and GraphNet layers have a superior agreement with our developed ground-truth logic, with median values greater than 0.7 for both virtual and core orbitals. As a general trend, we also observe that the spread of core AUC values is lower across all models, while the AUC values for virtual orbitals are more widely spread out, as indicated by the high variances in the figure. Nevertheless, it should be noted that within the presented approach we are not able to learn to distinguish between the more localized core orbitals and the more delocalized virtual orbitals, which could be useful information

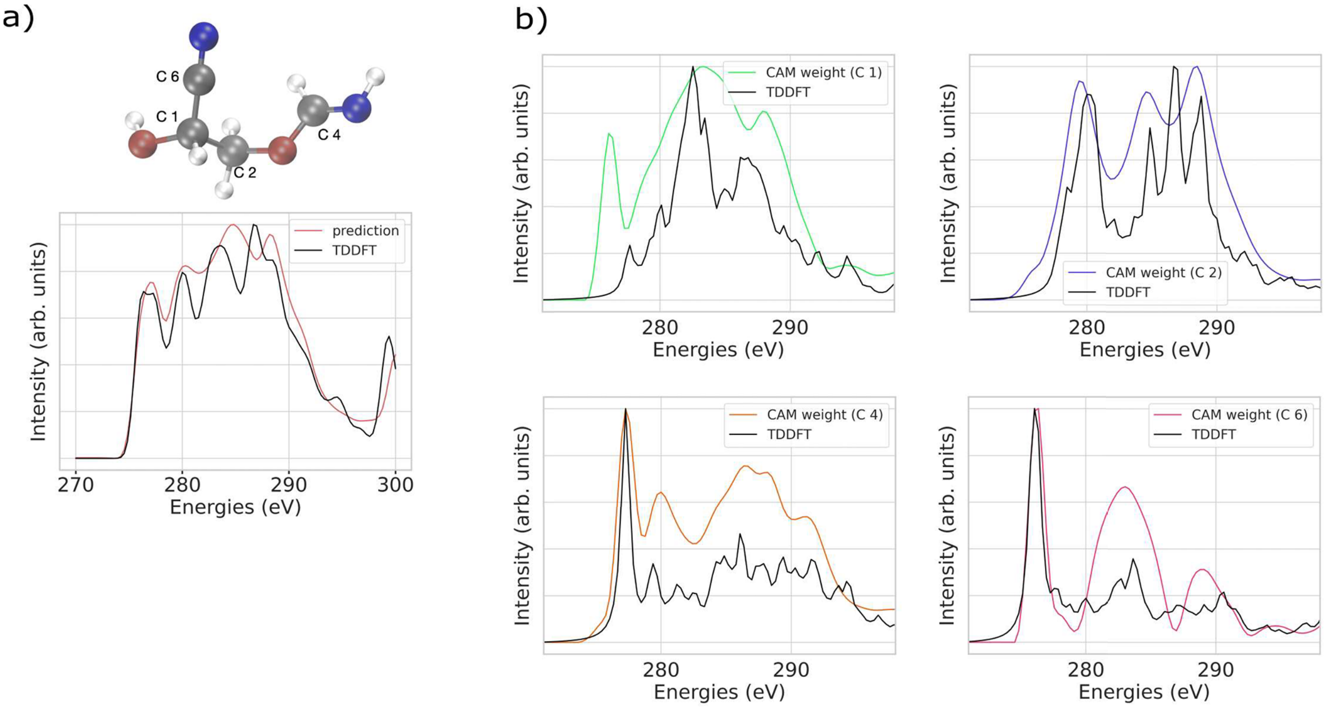


Figure 6. TDDFT (black) and GATv2 (red) predicted C K edge XAS spectra for an entire sample molecule (a). Calculated local XAS spectra (black) and CAM attribution weights (multiple colors) of individual carbon atoms in the molecule (b).

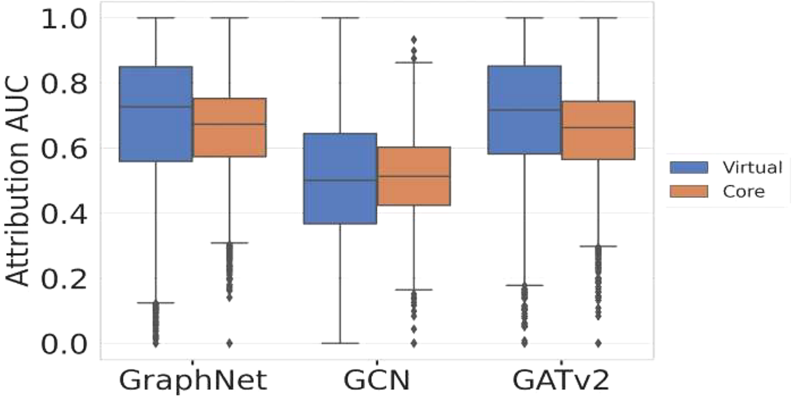


Figure 7. Attribution AUC score boxplots for the core and virtual orbitals of the three GNN models. The vertical line within the box indicates the median AUC value on the test data, while the length of the horizontal lines indicates the variance in AUC values for each model. Points beyond this range are considered outliers.

for the model to be included. Models that have higher attribution AUC values for core and virtual orbitals, i.e., GraphNet and GATv2, demonstrate a greater ability to comprehend the contribution of atoms to the excitation energies of the XAS spectrum. GraphNet models associate and encode global graph context in addition to the message-passing on node and edge level, and this perhaps positively influences CAM attributions, giving them information beyond the local environment. Given that the peaks in XAS analysis are highly dependent on the local geometric and electronic structures of atoms,[73,74] incorporating the interdependence of nodes and the global information on the molecular graph in GNNs, as done in these models, can facilitate capturing complex relationships between atomic coordination and specific excitation states in the XAS spectrum within its ML prediction. We expect that using multihead GATv2 and GraphNet architectures as GNNs for learning XAS spectra aligns with the essential understanding of the delocalized nature of molecular orbitals, which is crucial for accurate XAS prediction. Vaswani et al.[54] have shown previously that multihead attention, incorporated in the multihead GATv2 model, can improve the performance of models by enabling them to attend to different parts of the input molecular graph simultaneously. Wiegreffe and Pinter[75] have additionally shown that models that use the attention mechanism can provide better interpretability compared to nonattention frameworks, since they allow the visualization of which parts of the input are being attended by each head, making it easier to understand how the model is making predictions.

Thus, when it comes to XAS analysis, we can infer that the attention framework, which dynamically assigns importance weights to nodes surrounding a target node, yields superior

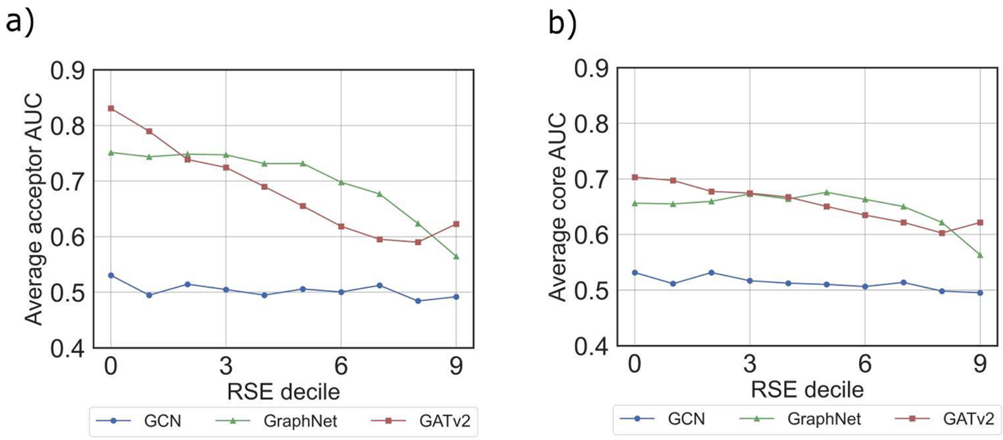


Figure 8. Variation of attribution AUC values for virtual (a) and core (b) orbitals with RSE decile values for three GNN models: GraphNet (green triangles), GATv2 (red squares), and GCN (blue points).

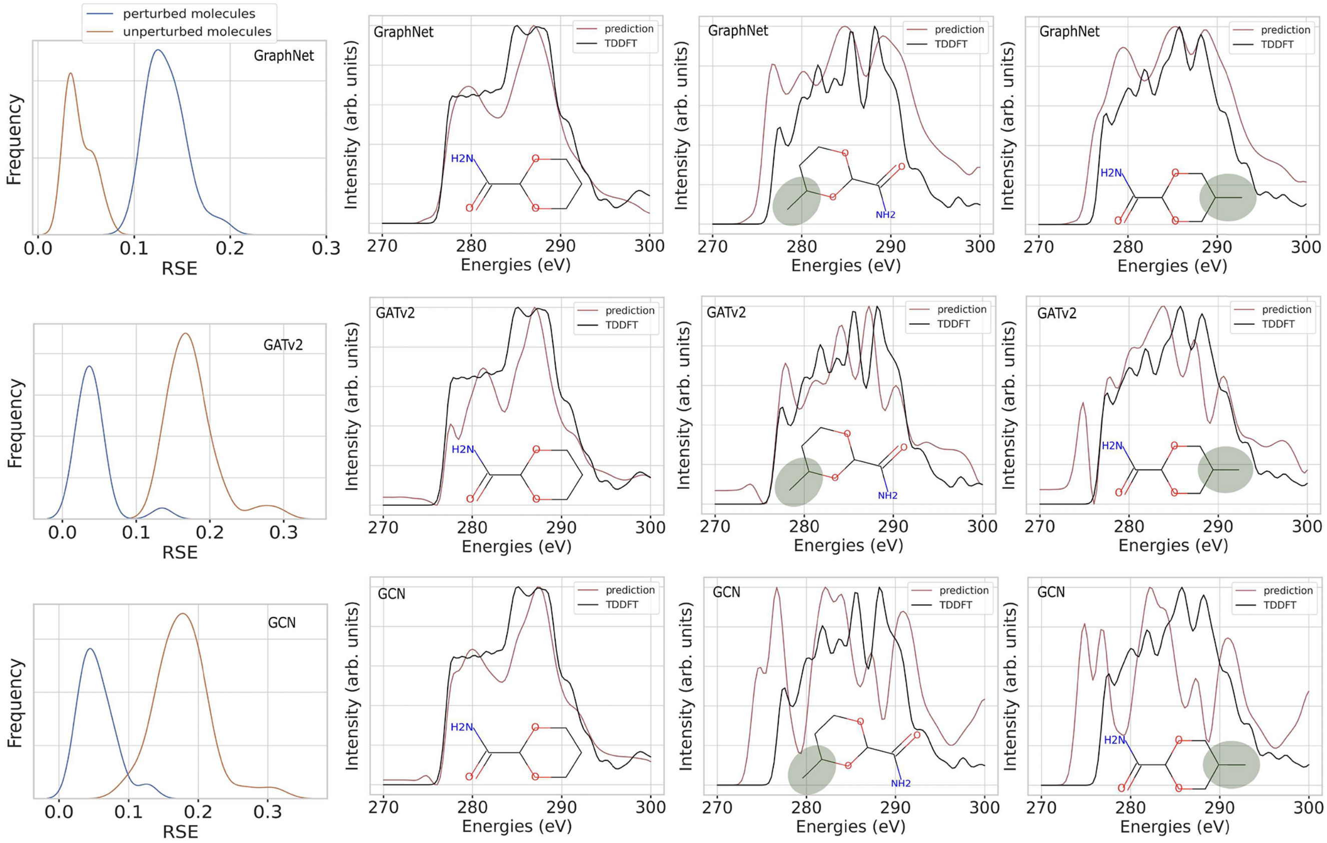


Figure 9. Impact of a perturbation through the replacement of functional groups. The left side of each row displays alterations in the RSE distribution for all GNN models when predicting the spectra for unperturbed structures selected from the data set (blue) and the perturbations of these structures (orange). Additionally, XAS spectra for different exemplary perturbations are shown (right), where a methyl group (highlighted in the gray circle) is added at different positions. The changed TDDFT spectra are shown in black, and their ML predictions are shown in red.

attribution values compared to those of the static node-weighting scheme employed by the GCN framework. Moreover, combinatorial generalization in GraphNets, which enhances their ability to generalize and perform well on new, unseen graph structures and tasks, is crucial to their applicability to XAS predictions in diverse molecular structures. On the other hand, robustness and generalization in GraphNet models, which incorporate relational inductive biases, have achieved improvement compared to traditional GNNs such as GCNs, over a range of graph classification and regression tasks.[76−78]

Robustness of the Explainability Performance of the GNN Models. Having shown that CAM attributions allow the explanation of the individual peaks in predicted XAS, the next task is to determine how robust this explainability is with respect to the prediction accuracy itself and to the changes in the data set. To address the influence of prediction quality on interpretability, we first explored how the attribution AUC

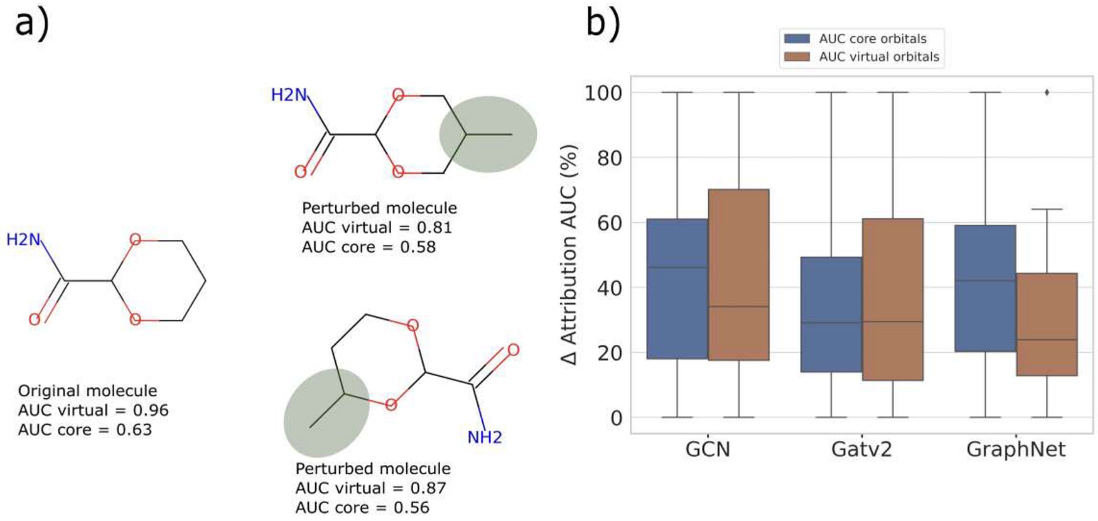


Figure 10. Attribution accuracy measured after perturbing random structures. (a) One specific molecular example to demonstrate the addition of $-CH_3$ groups as perturbation along with the change of AUC values according to the GraphNet model. (b) Δ-AUC plots for the perturbed set of test molecules across the three GNNs.

scores vary across different RSE values for the three GNN models. This is performed for each model by first distributing the molecules of the test data set into ten evenly large groups based on their RSE values. For these RSE deciles the average attribution AUC scores are computed and plotted in Figure 8 for both the virtual (a) and the core (b) orbitals. For the multihead GATv2 (red line) and the GraphNet (green line) models, the attribution AUC scores decline with increasing RSE values. The GATv2 sets on at the overall largest AUC of 0.83 (0.70) for the virtual (core) orbitals and then drops slightly below the GraphNet prediction to a value of about 0.6 (for both models). With the understanding that a larger counter of the RSE decile means a poorer prediction of the XAS spectrum, it becomes apparent that large AUC values are obtained when the overall spectrum prediction itself is reliable as well. Aligning this observation with the broader knowledge of quantum chemistry, we can infer that if ML predicts the spectrum more accurately, its understanding of orbital contributions improves correspondingly. In contrast, the GCN model's average attribution AUC exhibits no variation across RSE deciles, staying close to the random baseline value of 0.5. This suggests that the model has a similar level of understanding of the ground truth for both strong and weak performances in XAS prediction, which was already explained by the ML quality of the GCN model in the last section.

The robustness of model predictions (and their interpretations) usually decreases when there are biases in the training data set that the model erroneously learns.[38] The QM9 data set is only a small representation of the vast chemical space of organic molecules and as such is biased toward molecules with certain functional groups. Furthermore, choosing a random subset of structures from this data set means that the resulting structures in the smaller QM9-XAS data set could also be further biased toward one or several types of functional groups. To identify whether such biases are learned by the model, one approach is to analyze the attributions of the model's predictions and inspect whether CAM attributions are allocated to incorrect features of the input.[39] In this case, the robustness of model predictions is tested by looking at how the model performance varies for predictions across similar chemical environments. The simplest way of doing so is by perturbing the chemical space around a molecule, e.g., by adding one or several functional groups at different places. We investigate the impact of the addition of one methyl group on randomly selected molecules from the test data set on both the attribution AUC and the RSE value obtained with the GNN models. For these novel 40 perturbed structures, XAS spectra were calculated as a reference for RSE determination using the same TDDFT method as above. Adding a methyl group at different positions in a molecule leads to changes in the TDDFT spectrum, as well as in the ML predictions as illustrated in the three right panels of Figure 9. The three GNN architectures respond differently to this change and give vastly varying predictions of the new spectrum, as indicated by their increased RSE values as well. Overall, the ML spectra deviate significantly from the TDDFT spectra. This difference in predictions across all molecules is summarized in the left panel of Figure 9, which illustrates the change in the RSE performance of the models for the 40 selected structures before and after perturbation. The RSE distributions of the unperturbed set of molecules have slightly different shapes for the different models, but all give mostly the same average RSE value of approximately 0.03. With the perturbation, the RSE of the GCN and the GATv2 both shift to an RSE average of 0.18, while the GraphNet model gives about 0.13. The altered RSE distributions of the perturbations of these structures clearly indicate a decrease in model performance for perturbed molecules, with the GraphNet model demonstrating a superior performance compared to the others. This difference indicates that the GraphNet model can generalize better to chemical environments that are rarely encountered in the data set and are less susceptible to biases.

The changes in the RSE are significant, even for the GraphNet model. This change can be attributed to the fact that when a methyl group is included and replaces a hydrogen atom, the size of the molecule increases. The largest molecules within the original QM9-XAS data set consist of a maximum of nine heavy atoms (C, N, O, F), while the perturbed structures, on average, contain more than nine heavy atoms. This increase in molecular size potentially represents outliers to the trained

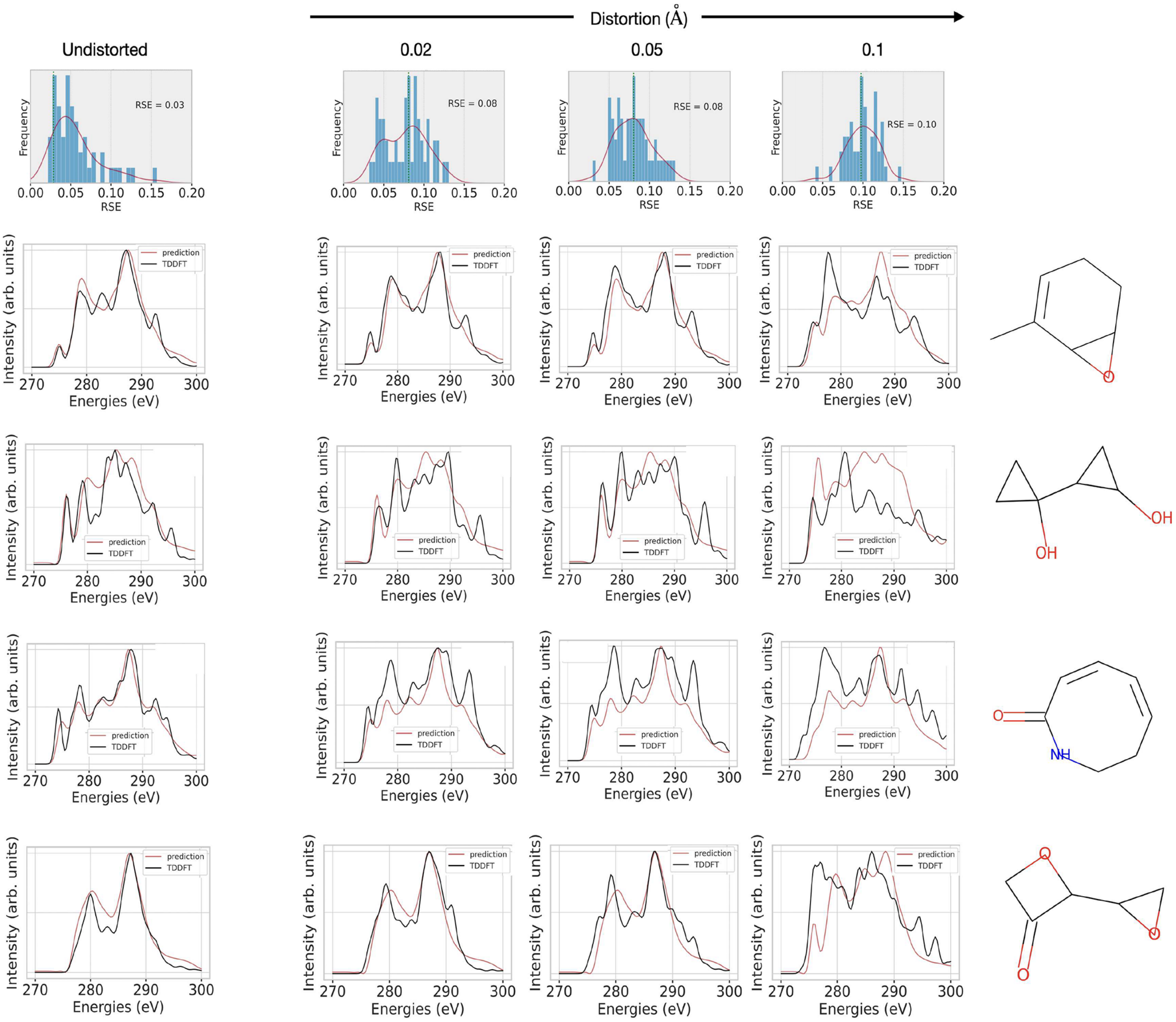


Figure 11. Evaluating the performance of the GATv2 model with structural distortions for four example molecules of the test set. The panels on the left give the TDDFT spectra (black) and the predicted spectra (red) for the undistorted case, while distortion is increasing for the three spectra on the right.

model, thereby leading to a decline in performance when predicting spectra.

Previous studies have demonstrated[35,38] that when a model fails to learn the ground-truth logic, it can result in misplaced attributions and the misclassification of atoms within the molecule after perturbations. We therefore now look at how attribution AUC changes for the spectra of the perturbed structures when compared with the AUC values of the original molecules. Figure 10 shows the Δ-attribution AUC across all the models for the perturbed structures, where the Δ-attribution AUC is the percentage-difference in the attributions of the 40 perturbed structures compared to the AUC values of the unperturbed molecules. While the multihead GATv2 model shows a 30% decline in the attribution AUC of core orbitals after perturbation, GCN and GraphNet models experience over 40% change. In the case of virtual orbitals, GraphNet and multihead GATv2 models decrease by 25% and 30%, respectively, while the GCN model shows a 35% drop. The drop in relative attributions uniformly across all of the models aligns with the increase in RSE values for these molecules, discussed in Figure 9. Such large changes in both core and virtual orbitals in all GNN models can originate from the effects of changes in both local and global molecular features on the spectrum after perturbations which results in changes in atomic contributions to the peaks in the spectrum. Hence, while the local environment of an atom, which refers to the atoms in close proximity to the absorbing atom, strongly affects the spectral features in the XAS spectrum, the global environment of that atom and changes caused by perturbations can also play a significant role in determining the electronic structure and thus the final XAS spectrum. This is also in line with previous research which showed that the presence of long-range interactions between atoms, as well as the coordination number, chemical nature, and distance of these neighboring atoms, can have strong influences on the spectral features, such as the position and width of the XAS peaks.[8,79]

These findings demonstrate the importance of incorporating the local and global environment of nodes while learning structure−property relationships using GNNs.[80]

In addition, we examine whether the GNN model with best performance (i.e., GATv2) mimics the changes expected with structural distortions. To obtain distorted molecules, we choose a distortion parameter $\sigma \in \{0.02, 0.05, 0.1\}$ Å to perturb randomly the atomic coordinates, i.e. in *x*-, *y*- and *z*-directions, in the respective molecule.[81] Figure 11 demonstrates how the TDDFT calculation and model prediction changes with respect to different distortion values. Already for the smallest distortion of only 0.02 Å, the TDDFT spectra change mostly in peak intensities and slightly in peak positions. These changes become more pronounced with stronger distortion. The model's prediction of small distortions looks similar to the undistorted one, i.e., predicting the general features of the spectrum, which, however, results in an increasing RSE with increasing distortion. For the largest distortion of 0.1 Å, the model mimics more closely the changes of the TD-DFT spectra, while the RSE values increase again. Such changes in XAS spectrum prediction suggest that the representation of small molecular conformation changes would require more structural information in node and edge feature vectors beyond the bond lengths. This could be the atomic pairwise distances, dihedral angles, etc.

■ CONCLUSION AND DISCUSSION

The aim of this work is to assist in the interpretation of peaks in X-ray absorption spectra (XAS) using a black-box machine learning (ML) method, i.e., graph neural networks (GNNs), as opposed to obtaining such information from purely conventional quantum chemical calculations. Yet, the underlying ground truth is based on the latter. In order to achieve this, we implement an explainability technique on various architectures of GNNs trained on a custom-developed carbon K-edge XAS data set of 65,000 small organic molecules, denoted as QM9-XAS, in which the molecules are a subset of the original QM9 data set.

The main difficulty in explaining properties with GNN models, as complex as the physical origin of peaks in XAS spectra already is, is the inherent lack of knowledge about the internal mechanisms of the model and how to correlate the properties of the model with the knowledge gained from quantum chemical calculations. We devised an approach that reflects a chemist's understanding of the XAS phenomenon as electronic excitations originating from individual atoms, which treats the underlying excitations of XAS peaks as a linear combination of core-to-valence orbital transitions and calculates the contribution of an individual atom to the participating core and valence orbitals. This produces atom labels denoting whether a particular atom contributes to an XAS peak within a specified energy range, allowing for the acquisition of the chemical ground truth and assessment of the extent to which an ML model comprehends the XAS spectra.

The rationale behind peaks observed in ML-predicted XAS spectra is unraveled via the so-called class activation map (CAM) attributions, highlighting the importance of individual nodes (atoms) in a molecular graph to the target peak of the spectrum. For a quantitative assessment of the graph attributions, we characterize the true and false positive rates of CAM attributions by calculating the area under the curve of the receiver operating characteristic (AUC-ROC), which is effectively a measure of how well the node attributions match the atomic contributions from the ground truth. Through this comparison between the chemical ground truth, i.e., here the core-to-valence orbital transitions, and CAM attributions, we demonstrate that while it is important to consider the overall performance of the GNN model in accurately predicting XAS features, the degree of explainability of the different architectures of GNN models differentiates them. We find that GNN models such as GraphNet and multihead GAT layers, which are in principle able to capture both the local and the more global chemical environment of an atom in a molecule, not only perform well in their spectra predictions but also the explanations obtained from these models are consistent with the quantum chemical interpretation of XAS.

To examine model robustness, we add a methyl group as a perturbation to a random set of molecules of the test data set of QM9-XAS. A decrease in performance is observed for all GNN models, with the GraphNet model showing the least decrease in performance, as assessed by the increase in relative spectral error (RSE). We suspect that the differences in the learning mechanisms between the three GNN architectures used have a significant effect on the changes in the RSE distribution and AUC attributions. The observed changes in attribution AUC highlight the limitations of relying only on the prediction accuracy obtained on a test data set to evaluate the performance of a model.

In conclusion, the approach presented here provides a recipe for incorporating explainability into GNN models using custom-generated data, which provides insight into the physical origin of spectroscopic predictions. Although the GNN models in this work are trained to predict the entire XAS spectrum, the model's attributions provide an opportunity to obtain some insights into local XAS spectra, i.e., for individual carbon atoms, with cost-effective computational resources. While our framework was demonstrated for carbon K-edge XAS prediction, the approach can be easily extended to other energy regimes, such as nitrogen and oxygen edges of molecules and metal complexes or even other spectroscopic techniques. Further, since this approach relies on theoretical data obtained from quantum chemical calculations, it can also be used to obtain ground-truth data for models trained on experimental data.

Direct comparison of predictions made in this approach to experimental spectra is challenging due to several factors influencing the experimental observations including solvent effects, experimental conditions like temperature and pressure, and structure-determining factors such as coexistence of multiple metastable conformers contributing to the experimental spectra. Incorporating these effects is often not so trivial using the existing theoretical approaches, and thus, corrections to theoretical spectra are necessary, often done on a case-by-case basis, depending on the molecular system and its environment. Considering the configurational phase space of the molecule in data set generation for training the model is one of the ways one can improve the discrepancy between a model's prediction and experimental spectra. For large molecular structures such as proteins and nanoparticles, computation of spectra at *ab initio* level of theory is often a challenge, although their XAS spectra can give insights into their different local environments.

While traditionally these have been tackled by the use of fingerprints determined on an ad-hoc basis, we believe that the development of more sophisticated and efficient machine learning frameworks, while maintaining explainability, offers a promising avenue for predicting spectra at low costs as well as getting insights into local molecular environments.

■ ASSOCIATED CONTENT

Data Availability Statement

The code used to train the models and generate the figures in this publication is publicly available at https://github.com/AI-4-XAS/XASNet-XAI. The QM9-XAS data set is available at 10.5281/zenodo.8276902.

■ AUTHOR INFORMATION

Corresponding Author

Annika Bande − *Helmholtz-Zentrum Berlin für Materialien und Energie GmbH, Berlin, DE 10409, Germany; Leibniz University Hannover, Institute of Inorganic Chemistry, Hannover, DE 30167, Germany;* orcid.org/0000-0003-3827-9169; Email: annika.bande@helmholtz-berlin.de

Authors

Amir Kotobi − *Helmholtz-Zentrum Hereon, Institute of Surface Science, Geesthacht, DE 21502, Germany;* orcid.org/0000-0002-1488-2847

Kanishka Singh − *Helmholtz-Zentrum Berlin für Materialien und Energie GmbH, Berlin, DE 10409, Germany; Institute of Chemistry and Biochemistry, Freie Universität Berlin, Berlin, DE 14195, Germany;* orcid.org/0000-0003-2700-9307

Daniel Höche − *Helmholtz-Zentrum Hereon, Institute of Surface Science, Geesthacht, DE 21502, Germany*

Sadia Bari − *Deutsches Elektronen-Synchrotron DESY, Hamburg, DE 22607, Germany; Zernike Institute for Advanced Materials, University of Groningen, Groningen 9712, Netherlands;* orcid.org/0000-0003-3985-2051

Robert H. Meißner − *Helmholtz-Zentrum Hereon, Institute of Surface Science, Geesthacht, DE 21502, Germany; Hamburg University of Technology, Institute of Polymers and Composites, Hamburg, DE 21073, Germany;* orcid.org/0000-0003-1926-114X

Complete contact information is available at:
https://pubs.acs.org/10.1021/jacs.3c07513

Funding

HIDA Trainee Network program, HAICU AI-4-XAS, DASHH and HEIBRiDS graduate schools

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

A.K. and K.S. gratefully acknowledge the support received from the Data Science in Hamburg - HELMHOLTZ Graduate School for the Structure of Matter (Grant-No. HIDSS-0002 DASHH) and Helmholtz Einstein International Berlin Research School in data science graduate schools, respectively. Furthermore, A.K. and A.B. thank the HIDA Trainee Network program for financing a stay of A.K. at HZB in 2022. The authors thank the DESY and JFZ computing centres for providing computing resources that enabled the creation of the dataset and subsequent training of the ML models. All authors thank the Impuls- und Vernetzungsfonds of the Helmholtz-Gemeinschaft for the Helmholtz-AI project AI-4-XAS.

■ REFERENCES

(1) Krasnikov, S. A.; Preobrajenski, A. B.; Sergeeva, N. N.; Brzhezinskaya, M. M.; Nesterov, M. A.; Cafolla, A. A.; Senge, M. O.; Vinogradov, A. S. Ni(II)-porphyrins with different ligands on the porphyrin ring, significant change of XAS spectra for different ligands on the ring. *Chem. Phys.* **2007**, *332*, 318−324.

(2) Guilherme Buzanich, A. Recent developments of X-ray absorption spectroscopy as analytical tool for biological and biomedical applications. *X-Ray Spectrometry* **2022**, *51*, 294−303.

(3) Frateloreto, F.; Tavani, F.; Mancini, M. D. B.; Giudice, D. D.; Capocasa, G.; Kieffer, I.; Lanzalunga, O.; Stefano, S. D.; D'Angelo, P. Following a Silent Metal Ion: A Combined X-ray Absorption and Nuclear Magnetic Resonance Spectroscopic Study of the Zn2+ Cation Dissipative Translocation between Two Different Ligands. *J. Phys. Chem. Lett.* **2022**, *13*, 5522−5529.

(4) Dörner, S.; Schwob, L.; Atak, K.; Schubert, K.; Boll, R.; Schlathölter, T.; Timm, M.; Bülow, C.; Zamudio-Bayer, V.; von Issendorff, B.; Lau, J. T.; Techert, S.; Bari, S. Probing Structural Information of Gas-Phase Peptides by Near-Edge X-ray Absorption Mass Spectrometry. *J. Am. Soc. Mass Spectrom.* **2021**, *32*, 670−684.

(5) Eisenberger, P.; Kincaid, B. M. EXAFS: New Horizons in Structure Determinations. *Science* **1978**, *200*, 1441−1447.

(6) Henderson, G. S.; de Groot, F. M.; Moulton, B. J. X-ray Absorption Near-Edge Structure (XANES) Spectroscopy. *Reviews in Mineralogy and Geochemistry* **2014**, *78*, 75−138.

(7) Cutsail, G. E., Iii; DeBeer, S. Challenges and Opportunities for Applications of Advanced X-ray Spectroscopy in Catalysis Research. *ACS Catal.* **2022**, *12*, 5864−5886.

(8) Rehr, J. J.; Albers, R. C. Theoretical approaches to x-ray absorption fine structure. *Rev. Mod. Phys.* **2000**, *72*, 621−654.

(9) Choudhary, K.; DeCost, B.; Chen, C.; Jain, A.; Tavazza, F.; Cohn, R.; Park, C. W.; Choudhary, A.; Agrawal, A.; Billinge, S. J. L.; Holm, E.; Ong, S. P.; Wolverton, C. Recent advances and applications of deep learning methods in materials science. *npj Computational Materials* **2022**, *8*, 59.

(10) Wellawatte, G. P.; Gandhi, H. A.; Seshadri, A.; White, A. D. A Perspective on Explanations of Molecular Prediction Models. *J. Chem. Theory Comput.* **2023**, *19*, 2149−2160.

(11) Mishin, Y. Machine-learning interatomic potentials for materials science. *Acta Mater.* **2021**, *214*, 116980.

(12) Schienbein, P. Spectroscopy from Machine Learning by Accurately Representing the Atomic Polar Tensor. *J. Chem. Theory Comput.* **2023**, *19*, 705−712.

(13) Faber, F.; Hutchison, L.; Huang, B.; Gilmer, J.; Schoenholz, S.; Dahl, G.; Vinyals, O.; Kearnes, S.; Riley, P.; von Lilienfeld, A. Prediction errors of molecular machine learning models lower than hybrid DFT error. *J. Chem. Theory Comput.* **2017**, *13* (11), 5255−5264.

(14) Duch, W.; Swaminathan, K.; Meller, J. Artificial Intelligence Approaches for Rational Drug Design and Discovery. *Current pharmaceutical design* **2007**, *13*, 1497−1508.

(15) Ko, T. W.; Finkler, J. A.; Goedecker, S.; Behler, J. A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer. *Nat. Commun.* **2021**, *12*, 398.

(16) Guda, A. A.; Guda, S. A.; Martini, A.; Kravtsova, A. N.; Algasov, A.; Bugaev, A.; Kubrin, S. P.; Guda, L. V.; Sot, P.; van Bokhoven, J. A.; Copéret, C.; Soldatov, A. V. Understanding X-ray absorption spectra by means of descriptors and machine learning algorithms. *npj Computational Materials* **2021**, *7*, 203.

(17) Rankine, C. D.; Penfold, T. J. Accurate, affordable, and generalizable machine learning simulations of transition metal x-ray absorption spectra using the XANESNET deep neural network. *J. Chem. Phys.* **2022**, *156*, 164102.

(18) Singh, K.; Münchmeyer, J.; Weber, L.; Leser, U.; Bande, A. Graph Neural Networks for Learning Molecular Excitation Spectra. *J. Chem. Theory Comput.* **2022**, *18*, 4408−4417.

(19) Ghosh, K.; Stuke, A.; Todorović, M.; Jørgensen, P. B.; Schmidt, M. N.; Vehtari, A.; Rinke, P. Machine Learning: Deep Learning

Spectroscopy: Neural Networks for Molecular Excitation Spectra (Adv. Sci. 9/2019). *Advanced Science* **2019**, *6*, 1970053.

(20) Gastegger, M.; Behler, J.; Marquetand, P. Machine learning molecular dynamics for the simulation of infrared spectra. *Chemical Science* **2017**, *8*, 6924−6935.

(21) Kotobi, A.; Schwob, L.; Vonbun-Feldbauer, G. B.; Rossi, M.; Gasparotto, P.; Feiler, C.; Berden, G.; Oomens, J.; Oostenrijk, B.; Scuderi, D.; Bari, S.; Meißner, R. H. Reconstructing the infrared spectrum of a peptide from representative conformers of the full canonical ensemble. *Communications Chemistry* **2023**, *6*, 46.

(22) Aarva, A.; Deringer, V. L.; Sainio, S.; Laurila, T.; Caro, M. A. Understanding X-ray Spectroscopy of Carbonaceous Materials by Combining Experiments, Density Functional Theory, and Machine Learning. Part I: Fingerprint Spectra. *Chem. Mater.* **2019**, *31*, 9243−9255.

(23) Aarva, A.; Sainio, S.; Deringer, V. L.; Caro, M. A.; Laurila, T. X-ray Spectroscopy Fingerprints of Pristine and Functionalized Graphene. *J. Phys. Chem. C* **2021**, *125*, 18234−18246.

(24) Golze, D.; Hirvensalo, M.; Hernández-León, P.; Aarva, A.; Etula, J.; Susi, T.; Rinke, P.; Laurila, T.; Caro, M. A. Accurate Computational Prediction of Core-Electron Binding Energies in Carbon-Based Materials: A Machine-Learning Model Combining Density-Functional Theory and *GW*. *Chem. Mater.* **2022**, *34*, 6240−6254.

(25) Rankine, C. D.; Madkhali, M. M. M.; Penfold, T. J. A Deep Neural Network for the Rapid Prediction of X-ray Absorption Spectra. *J. Phys. Chem. A* **2020**, *124*, 4263−4270.

(26) Carbone, M. R.; Topsakal, M.; Lu, D.; Yoo, S. Machine-Learning X-Ray Absorption Spectra to Quantitative Accuracy. *Phys. Rev. Lett.* **2020**, *124*, 156401.

(27) Yuan, H.; Yu, H.; Gui, S.; Ji, S. Explainability in Graph Neural Networks: A Taxonomic Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2022**, *45*, 5782−5799.

(28) Castillo-Michel, H. A.; Larue, C.; Pradas del Real, A. E.; Cotte, M.; Sarret, G. Practical review on the use of synchrotron based micro- and nano- X-ray fluorescence mapping and X-ray absorption spectroscopy to investigate the interactions between plants and engineered nanomaterials. *Plant Physiology and Biochemistry* **2017**, *110*, 13−32.

(29) Zimmermann, P.; Peredkov, S.; Abdala, P. M.; DeBeer, S.; Tromp, M.; Müller, C.; van Bokhoven, J. A. Modern X-ray spectroscopy: XAS and XES in the laboratory. *Coord. Chem. Rev.* **2020**, *423*, 213466.

(30) Doshi-Velez, F.; Kim, B. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv* **2017**, 1702.08608 DOI: 10.48550/arXiv.1702.08608.

(31) Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* **2019**, *267*, 1−38.

(32) Gunning, D.; Aha, D. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Magazine* **2019**, *40*, 44−58.

(33) Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM:Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision* **2020**, *128*, 336−359.

(34) Pope, P. E.; Kolouri, S.; Rostami, M.; Martin, C. E.; Hoffmann, H. Explainability Methods for Graph Convolutional Neural Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* **2019**, 10764−10773, DOI: 10.1109/CVPR.2019.01103.

(35) Sanchez-Lengeling, B.; Wei, J.; Lee, B.; Reif, E.; Wang, P.; Qian, W.; McCloskey, K.; Colwell, L.; Wiltschko, A. Evaluating Attribution for Graph Neural Networks. *Advances in Neural Information Processing Systems* **2020**, *33*, 5898−5910.

(36) Sattarzadeh, S.; Sudhakar, M.; Lem, A.; Mehryar, S.; Plataniotis, K. N.; Jang, J.; Kim, H.; Jeong, Y.; Lee, S.; Bae, K. Explaining Convolutional Neural Networks through Attribution-Based Input Sampling and Block-Wise Feature Aggregation. *Proceedings of the AAAI Conference on Artificial Intelligence* **2021**, *35* (13), 11639−11647.

(37) Bartička, V.; Pražák, O.; Konopík, M.; Sido, J. Evaluating Attribution Methods for Explainable NLP with Transformers. In *Text, Speech, and Dialogue*; 2022; pp 3−15. DOI: 10.1007/978-3-031-16270-1_1

(38) McCloskey, K.; Taly, A.; Monti, F.; Brenner, M. P.; Colwell, L. J. Using attribution to decode binding mechanism in neural network models for chemistry. *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116*, 11624−11629.

(39) Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic Attribution for Deep Networks. *Proceedings of the 34th International Conference on Machine Learning*; 2017; pp 3319−3328.

(40) Marques, M. A. L., Ed. Time-dependent density functional theory; *Lecture notes in physics 706*; Springer: Berlin, 2006.

(41) Egorov, D.; Schwob, L.; Lalande, M.; Hoekstra, R.; Schlathölter, T. Near edge X-ray absorption mass spectrometry of gas phase proteins: the influence of protein size. *Phys. Chem. Chem. Phys.* **2016**, *18*, 26213−26223.

(42) Minasian, S. G.; Keith, J. M.; Batista, E. R.; Boland, K. S.; Kozimor, S. A.; Martin, R. L.; Shuh, D. K.; Tyliszczak, T.; Vernon, L. J. Carbon K-Edge X-ray Absorption Spectroscopy and Time-Dependent Density Functional Theory Examination of Metal−Carbon Bonding in Metallocene Dichlorides. *J. Am. Chem. Soc.* **2013**, *135*, 14731−14740.

(43) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data* **2014**, *1*, 140022.

(44) Petersilka, M.; Gossmann, U. J.; Gross, E. K. U. Excitation Energies from Time-Dependent Density-Functional Theory. *Phys. Rev. Lett.* **1996**, *76*, 1212−1215.

(45) Neese, F. The ORCA program system. *WIREs Computational Molecular Science* **2012**, *2*, 73−78.

(46) Becke, A. D. Density-fnnctional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A* **1988**, 3098 DOI: 10.1103/PhysRevA.38.3098.

(47) Weigend, F.; Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297.

(48) RDKit: Open-source cheminformatics (Accessed 2020-04-16).

(49) Fey, M.; Lenssen, J. E. *Fast Graph Representation Learning with PyTorch Geometric*. 2019.

(50) Wieder, O.; Kohlbacher, S.; Kuenemann, M.; Garon, A.; Ducrot, P.; Seidel, T.; Langer, T. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies* **2020**, *37*, 1−12.

(51) Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*; 2015; Vol. 2, pp 2224−2232.

(52) Battaglia, P. W. Relational inductive biases, deep learning, and graph networks. *arXiv* **2018**, 1806.01261 DOI: 10.48550/arXiv.1806.01261.

(53) Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *International Conference on Learning Representations*; 2017.

(54) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *Advances in Neural Information Processing Systems*; 2017.

(55) Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *International Conference on Learning Representations*; 2017.

(56) Despraz, J.; Gomez, S.; Satizábal, H. F.; Peña-Reyes, C. A. Towards a Better Understanding of Deep Neural Networks Representations using Deep Generative Networks. *Proceedings of the 9th International Joint Conference on Computational Intelligence*. Funchal, Madeira, Portugal, 2017; pp 215−222.

(57) Ancona, M.; Ceolini, E.; Öztireli, C.; Gross, M. Towards better understanding of gradient-based attribution methods for Deep Neural

Networks. *arXiv* **2018**, 1711.06104 DOI: 10.48550/arXiv.1711.06104.

(58) Jiménez-Luna, J.; Skalic, M.; Weskamp, N. Benchmarking Molecular Feature Attribution Methods with Activity Cliffs. *J. Chem. Inf. Model.* **2022**, *62*, 274−283.

(59) Shrikumar, A.; Greenside, P.; Kundaje, A. Learning Important Features through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning*; 2017; vol. *70*, pp 3145−3153.

(60) Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016; pp 2921−2929.

(61) Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D.; Grad-CAM:Visual Explanations from Deep Networks via Gradient-Based Localization. *2017 IEEE International Conference on Computer Vision (ICCV)*; 2017; pp 618−626.

(62) Oviedo, F.; Ferres, J. L.; Buonassisi, T.; Butler, K. T. Interpretable and Explainable Machine Learning for Materials Science and Chemistry. *Accounts of Materials Research* **2022**, *3*, 597−607.

(63) Yano, J.; Yachandra, V. K. X-ray absorption spectroscopy. *Photosynthesis Research* **2009**, *102*, 241−254.

(64) Weber, F.; Ren, J.; Petit, T.; Bande, A. Theoretical X-ray absorption spectroscopy database analysis for oxidised 2D carbon nanomaterials. *Phys. Chem. Chem. Phys.* **2019**, *21*, 6999−7008.

(65) Sharma, A.; Lysenko, A.; Boroevich, K. A.; Vans, E.; Tsunoda, T. DeepFeature: feature selection in nonimage data using convolutional neural network. *Briefings in Bioinformatics* **2021**, *22*, bbab297.

(66) Agarwal, C.; Queen, O.; Lakkaraju, H.; Zitnik, M. Evaluating explainability for graph neural networks. *Scientific Data* **2023**, *10*, 144.

(67) Tang, B.; Kramer, S. T.; Fang, M.; Qiu, Y.; Wu, Z.; Xu, D. A self-attention based message passing neural network for predicting molecular lipophilicity and aqueous solubility. *Journal of Cheminformatics* **2020**, *12*, 15.

(68) Brumboiu, I. E.; Fransson, T. Core−hole delocalization for modeling x-ray spectroscopies: A cautionary tale. *J. Chem. Phys.* **2022**, *156*, 214109.

(69) Bradley, A. P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* **1997**, *30*, 1145−1159.

(70) Heberle, H.; Zhao, L.; Schmidt, S.; Wolf, T.; Heinrich, J. XSMILES: interactive visualization for molecules, SMILES and XAI attribution scores. *Journal of Cheminformatics* **2023**, *15*, 2.

(71) Lee, S.; Park, H.; Choi, C.; Kim, W.; Kim, K. K.; Han, Y.-K.; Kang, J.; Kang, C.-J.; Son, Y. Multi-order graph attention network for water solubility prediction and interpretation. *Sci. Rep.* **2023**, *13*, 957.

(72) Withnall, M.; Lindelöf, E.; Engkvist, O.; Chen, H. Building attention and edge message passing neural networks for bioactivity and physical-chemical property prediction. *Journal of Cheminformatics* **2020**, *12*, 1.

(73) Frati, F.; Hunault, M. O. J. Y.; de Groot, F. M. F. Oxygen K-edge X-ray Absorption Spectra. *Chem. Rev.* **2020**, *120*, 4056−4110.

(74) Risch, M.; Morales, D. M.; Villalobos, J.; Antipin, D. What X-Ray Absorption Spectroscopy Can Tell Us About the Active State of Earth-Abundant Electrocatalysts for the Oxygen Evolution Reaction. *Angew. Chem., Int. Ed.* **2022**, *61*, No. e202211949.

(75) Wiegreffe, S.; Pinter, Y. Attention is not not explanation. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* **2019**, 11−20.

(76) Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; Welling, M. *Modeling Relational Data with Graph Convolutional Networks*; The Semantic Web: Cham, 2018; pp 593−607.

(77) Busbridge, D.; Sherburn, D.; Cavallo, P.; Hammerla, N. Y. Relational Graph Attention Networks. *arXiv* **2019**, 1904.05811 DOI: 10.48550/arXiv.1904.05811.

(78) Yun, S.; Jeong, M.; Yoo, S.; Lee, S.; Yi, S. S.; Kim, R.; Kang, J.; Kim, H. J. Graph Transformer Networks: Learning meta-path graphs to improve GNNs. *Neural Networks* **2022**, *153*, 104−119.

(79) de Groot, F. X-ray absorption and dichroism of transition metal compounds. *AIP Conf. Proc.* **1997**, *389*, 497−520.

(80) Chen, Z.; Andrejevic, N.; Drucker, N. C.; Nguyen, T.; Xian, R. P.; Smidt, T.; Wang, Y.; Ernstorfer, R.; Tennant, D. A.; Chan, M.; Li, M. Machine learning on neutron and x-ray scattering and spectroscopies. *Chemical Physics Reviews* **2021**, *2*, 031301.

(81) Ghose, A.; Segal, M.; Meng, F.; Liang, Z.; Hybertsen, M. S.; Qu, X.; Stavitski, E.; Yoo, S.; Lu, D.; Carbone, M. R. Uncertainty-aware predictions of molecular x-ray absorption spectra using neural network ensembles. *Physical Review Research* **2023**, *5*, 013180.

A.3 Accelerating wavepacket propagation with Machine Learning

Kanishka Singh, Ka Hei Lee, Daniel Pelaéz, Annika Bande

Subnmitted to J.Comput.Chem.
DOI:
URL:

Contributions: The primary groundwork for this work was laid by research done in the group of DP. AB and KS helped streamline the work through the selection of test cases, the FNO model as the framework of choice, and demonstrating project viability using the MCMC optimization approach. KS wrote the code for the project using existing Python implementations of the FNO model, and KHL helped fine-tune and optimize it. KHL also helped generate the data for the 2D system using MCTDH. KS wrote the manuscript, with extensive support from the co-authors. KHL further contributed to the manuscript through his inputs in the data, theory, and various figures. AB and DP provided extensive feedback for writing the manuscript and guiding the project to completion.

Accelerating wavepacket propagation with Machine Learning

Kanishka Singh[*,†], Ka Hei Lee[‡,§], Daniel Peláez[¶], Annika Bande[‖,**,††,‡‡]

December 15, 2023

Abstract

In this work, we discuss the use of a recently introduced machine learning (ML) technique known as Fourier neural operators (FNO) as an efficient alternative to the *traditional* solution of the time-dependent Schrödinger equation (TDSE). FNOs are ML models which are employed in the approximated solution of partial differential equations. Specifically, we demonstrate that FNOs can be a suitable replacement for traditional TDSE solvers in cases where the results of the quantum dynamical simulation are required repeatedly such as in the case of parameter optimization problems (e.g. control). The speed-up from the FNO method allows for its combination with the Markov-chain Monte Carlo approach in applications that involve solving inverse problems such as optimal and coherent laser control of the outcome of dynamical processes.

Keywords: Machine Learning, Quantum Dynamics, Fourier Neural Operators ■

[*] Theory of Electron Dynamics and Spectroscopy, Helmholtz-Zentrum Berlin für Materialien und Energie GmbH, Hahn-Meitner-Platz 1, 10409 Berlin, Germany

[†] Institute of Chemistry and Biochemistry, Freie Universität Berlin, Arnimallee 22, 14195 Berlin, Germany

[‡] Theory of Electron Dynamics and Spectroscopy, Helmholtz-Zentrum Berlin für Materialien und Energie GmbH, Hahn-Meitner-Platz 1, 10409 Berlin, Germany

[§] Fachbereich Physik, Freie Universität Berlin, Arnimallee 14, 14195 Berlin, Germany

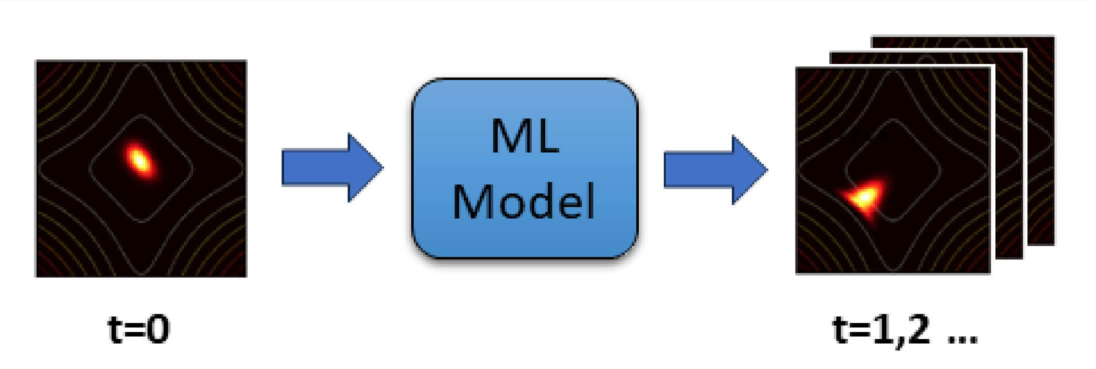
[¶] Université Paris-Saclay, CNRS, Institut des Sciences Moléculaires d'Orsay, 91405, Orsay, France

[‖] Theory of Electron Dynamics and Spectroscopy, Helmholtz-Zentrum Berlin für Materialien und Energie GmbH, Hahn-Meitner-Platz 1, 10409 Berlin, Germany

[**] Institute of Inorganic Chemistry, Leibniz University Hannover, Callinstr. 9, 30167 Hannover, Germany

[††] Cluster of Excellence PhoenixD, Leibniz University Hannover, Welfengarten 1A, 30167 Hannover, Germany

[‡‡] email: annika.bande@helmholtz-berlin.de



Solving the Time-Dependent Schrödinger equation provides insights into several time-dependent molecular phenomena. Conventional numerical methods can accurately solve this equation but often have high computational cost. In this work, we show how an ML model that can generate entire wavepacket simulations in one computation, can be a useful alternative to these methods, especially in applications that require multiple dynamical calculations.

1 Introduction

An accurate theoretical description of molecular phenomena requires the solution of the Schrödinger equation (SE) in any of its variants, time-independent (TISE)[1] or time-dependent (TDSE)[2]. It should be emphasized the more general character of the latter owing to the fact that not only time-dependent processes can be visualized but also, at the same time, time-independent quantities such as eigenstates or spectra can be computed[3,4]. The efficient solution of the SE (in terms of CPU time and storage) has constituted and still constitutes one of the greatest bottlenecks to its widespread use. Two are the main limiting aspects: (i) the need for storing large amounts of information, as epitomized by the exponential growth in the number of data points with dimensionality, and (ii) the actual integration of the SE (i.e. *application of a propagator*) or, equivalently, the numerical solution of some kind of equations of motion. The former issue has been successfully overcome through the use of efficient (large) tensor decomposition schemes (see for instance[5–9]). Concerning the latter aspect, clever and sophisticated integration schemes[3,10,11], many of them relying on different data structures (e.g. tensor networks[12]) have been developed over the past fifty years.
ML approaches are modern numerical computer science techniques that are in terms of efficiency a true alternative to conventional algorithms in many domains. They have recently shown success in replacing the quantum-mechanical evaluation of the TDSE and obtaining molecular properties at a faster speed without losing accuracy. Deep-learning approaches involving complex neural networks can be used for the accurate generation of potential energy surfaces[13] and to predict chemical reaction outcomes[14,15], or for property prediction[16,17]. Another profiting domain is classical molecular dynamics using Newtonian equations of motion.[18,19] Compared to the numerous applications in these two domains, there are only relatively few attempts to use ML to solve quantum-dynamics problems described by the TDSE, which is structurally a partial differential equation (PDE).[20–27]
The solution of PDEs, such as the Burgers or Navier-Stokes equation for the simulation of macroscopic diffusion and flow processes[28], the wave equation which is useful in wave propagation in acoustics and physics[29], the heat equation, which is used to model heat diffusion in engineering and materials research[30], or even the TDSE rely on the discretization of the configuration space, in other words, the creation of a grid. An accuracy-grid size tradeoff arises in such methods, which means that while finer grids can give accurate solutions to a PDE, the cost of evaluating the PDE is higher here. This bottleneck prevents conventional numerical methods from being used for large systems or long simulation times.
Complementary data-driven approaches for solving PDEs have recently gained attention as they hold promise for lower-cost accurate solutions for various differential equations across a variety of domains[31–35]. Data-driven methods of PDE solving can be classified into three main categories. The first category focuses on improving the computational efficiency, scaling, and accuracy of existing numerical PDE solvers[36,37]. The advantage of such an approach is that it builds on years of development in numerical approaches and improves their applicability for larger and complex systems. The second approach uses neural networks to approximate solution functions of a PDE, often using physical constraints to improve the accuracy of predictions[38,39]. These two approaches require prior knowledge of the PDE either to improve the computation of various quantities for the numerical solver (in the first approach) or for incorporating boundary conditions and physically inspired losses for model

training (in the second approach). Additionally, the second approach is limited by the fact that one can only train for one instance (set of parameters) of a PDE at one time, and training for a new instance requires retraining the model from scratch.

Operator learning[40–43] is a third and emerging approach of learning PDEs from data using ML neural operator architectures. An operator, simply put, is a function capable of mapping one space of functions with another (assuming there exists such a mapping). Several approaches for solving PDEs in quantum mechanics, electrodynamics, or elasticity have relied on the development of such operators capable of mapping initial states to outputs. A neural operator uses an artificial neural network to learn this mapping. As such, these architectures take as inputs data generated using different variables/conditions of the same PDE and learn to map them to their respective outputs. The advantage of such an approach is apparent: Instead of learning a single instance of PDE for a set of initial conditions, the neural operator framework allows learning different instances (set of variables) of a PDE within a certain range of parameters. Additionally, the method is entirely data-driven and requires no prior knowledge of the PDE of the physical system. This extends the usability of the neural-operator approach to experimental data as well, where an explicit PDE for realizing the dynamics of the system is often not available. The neural-operator approach can also be used for different grid sizes, owing to its ability to transfer solutions between different meshes.

A major deficiency of neural operators is their poor training speed when compared to other methods mentioned above. Hence to remedy this, the Fourier neural operator (FNO) approach was introduced. FNOs employ Fourier transforms to alleviate this computational bottleneck (elaborated in Section on Fourier Neural Operators) associated with the neural operator approach[41]. With this FNO approach, the learning of a family of PDEs at low computational costs of training and inference has been possible, leading to its applications in weather predictions[44], geology[45], imaging[46], fluid dynamics,[47] as well as physically-inspired solutions to model PDE systems in various scientific applications[48].

This ability to learn a family of PDEs from data opens up the possibility of using data-driven approaches for quantum-dynamical problems within the Fourier neural operator approach. Quantum dynamics involves solving the TDSE for a varied range of potentials and wavepackets representing quantum systems, which leads to a variety of output propagations. Many quantum-dynamical applications require exploring different combinations of variables in the same system, thus requiring several simulations for similar quantum systems. To demonstrate the viability of FNO-trained models for quantum dynamical applications, we show that FNO-trained models can faithfully and accurately reproduce simulations for two different quantum dynamical problems.

In this work the FNO models are trained on two datasets of outputs of different Gaussian initial wavepackets propagating on different potentials in low-dimensional spatial grids. The first example studied consists of a two-dimensional anharmonic potential system. Dynamical systems with such anharmonic potentials are used to model the vibrational motion of small molecules[49–51]. This work demonstrates that, given a dataset of various wavepackets centered initially at different grid points on a 2D anharmonic potential, an FNO model can be trained to accurately and swiftly generate an entire propagation for previously unseen wavepackets. The second example is a double well potential representing the internal rotation within a molecule along one of its covalent single bonds, which has been studied extensively[52–55]. In the specific case of this study, the rotation goes back to quantum tunneling induced (and explicitly controllable) by a pulsed electromagnetic field. For the control of molecular rotation, the final state (after successful tunneling) is of interest. The question lies in finding the optimal electromagnetic field to reach this state completely and fast. Such problems can be termed inverse problems, wherein a target state is already known, and the goal is to find the initial conditions leading to this target. In respective state-of-art quantum-dynamics applications, optimal control[56–60] has often been employed to obtain the field conditions necessary for attaining a target state. While most control problems involve some iterative/deterministic procedure to solve the Hamiltonians such that an optimal set of parameters is realized, the speed up from FNOs can be utilized to solve such a problem using Bayesian optimization[61] and here specifically the Bayesian technique of Markov Chain Monte Carlo (MCMC)[62]. A parameter space for the laser pulse and an objective function is defined, which is then procedurally sampled to obtain the optimal pulse leading to the desired output. Since the FNO can accurately model the Hamiltonians and the wavepacket propagation for a large number of varied instances of potentials at a low-compute cost, it can be employed in such an optimization process as a faster alternative to conventional solvers in these applications. For a model coherent control system already explored with ML in Thomas and Henriksen[52], this work shows that MCMC can be efficiently combined with FNOs to solve such problems. Furthermore, to the best of our knowledge, this constitutes the first application of FNO models to the solution of the time-dependent Schrödinger equation.

The article is structured as follows: A formulation of how quantum dynamics solves the TDSE to study wavepacket propagation and how numerical methods are used to implement this for chemical systems is given first. This is followed by a description of the model chemical systems for which propagation using ML is demonstrated. The theory section then concludes by describing the FNO architecture and the MCMC approach for functional optimization. This is followed by a description of the training procedure and computational features of the models, before delving into an analysis of the results of training them. Finally, a discussion on how the MCMC combined with the FNO approach can be used as an alternative to optimal/coherent control is presented, before addressing the limitations and providing directions for further research in the domain of ML accelerated quantum dynamics.

2 Quantum Dynamics

Quantum dynamics involves propagating a wavepacket $\Psi(\mathbf{r}, t_0)$ at time $t_0 = 0$ for a finite time period t_{sim} using the TDSE

$$i\hbar \frac{\partial \Psi(\mathbf{r}, t)}{\partial t} = \hat{H}\Psi(\mathbf{r}, t) \ . \tag{1}$$

The Hamiltonian operator $\hat{H}$ can be split into kinetic and potential energy operators, $\hat{T}$ and $\hat{V}$, respectively, that are separable in spatial coordinates and can be applied separately to the wavepacket of the system. The potential of the operator $\hat{V}$ can be a function of time, in the case of the potential of the system being time-dependent. In this work, these two cases will be considered, one where the potential is time-independent and another where the potential varies with time. The TDSE, furthermore, can be converted to a propagation equation with a propagator $\mathbf{U}$, reading

$$\Psi(\mathbf{r},t) = \mathbf{U}\Psi(\mathbf{r},t_0) = \exp\left(-\frac{i}{\hbar}\hat{H}(t-t_0)\right)\Psi(\mathbf{r},t_0)\ , \tag{2}$$

The propagator then models the quantum dynamics of the wavepacket by mapping the initial state at $t = 0$ to a series of wavepackets $(\mathbf{r}, t_0), (\mathbf{r}, t_1)...(\mathbf{r}, t_{sim})$ in the future advancing by time steps dt. Different numerical schemes exist to implement the propagation of wavepackets, of which we discuss two methods that were employed in this study in the following sections.

2.1 Conventional methods for solving the TDSE

Several numerical techniques based on the propagator method have been developed to solve the TDSE. The Chebyshev scheme[63] approximates the global propagator $\mathbf{U}$ using the Chebsyhev polynomial. Similarly, the second-order difference (SOD)[10] scheme expands the time evolution (propagation) operator in a Taylor series. The split operator is another numerical scheme that converts the operators of the Hamiltonian to their diagonal forms (in the Fourier space) to propagate the time evolution operator efficiently. In contrast, wavepacket-based approaches like multiconfiguration time-dependent Hartree (MCTDH)[3] or variational multiconfiguration Gaussian (vMCG)[64] define a specific ansatz for the functional form of wavepackets. Once defined, the wavepackets are propagated under the Hamiltonian in adherence with the variational principle, giving the equations of motion (EOM) required for obtaining wavepacket coefficients. Efficient numerical integration methods are then employed to solve these EOMs and compute the observables of the system. To demonstrate the viability and versatility of the ML approach for data generated by either method, we employ the split operator method for one binding potential and the MCTDH method for the other.

2.1.1 Split-Operator Method

The split-operator method first introduced in 1982[65] is a numerical technique primarily employed to tackle time-dependent PDEs, including the TDSE. It splits the Hamiltonian operator into position and momentum components by executing a sequence of operations using Fourier transforms.

Considering the Hamiltonian $\hat{H}$ of Equation 1 and assuming that it can be separated into momentum (p) and position (r) components, $\hat{H}$ can be written as

$$\hat{H} = \hat{H}_p + \hat{H}_r\ , \tag{3}$$

where $\hat{H}_p = -\frac{1}{2}\sum_{i=1}^{f}\nabla_i^2$ is the kinetic energy operator (in atomic units) and $\hat{H}_r = V(\mathbf{r})$ is the potential operator. At an arbitrary point in time, t, and using the time step as dt, the respective propagator, a product of the exponential function of the non-commuting operators $\hat{H}_p$ and $\hat{H}_r$, can be written as

$$e^{-i\hat{H}_p dt}e^{-i\hat{H}_r dt} = e^{-i(\hat{H}_p+\hat{H}_r)dt+\frac{1}{2}[-i\hat{H}_p dt,-i\hat{H}_r dt]+...} \tag{4}$$

using the Baker-Campbell-Hausdorff formula. For a more accurate approximation, a technique known as Strang splitting[65] is employed, which involves dividing the system's evolution into two distinct steps. Initially, a half-step is performed in position space, followed by a subsequent full step in momentum space. This approach allows for the approximation of the propagated wavefunction as

$$\Psi(\mathbf{r}, t+dt) = \Big[e^{-i\hat{H}_r\frac{dt}{2}}e^{-i\hat{H}_p dt}e^{-i\hat{H}_r\frac{dt}{2}}\Big]\Psi(\mathbf{r}, t) + \mathcal{O}(dt^3) \tag{5}$$

The transformation between position and momentum space can be reached by using fast Fourier transforms. Defining $\mathcal{F}$ and $\mathcal{F}^{-1}$ as the Fourier transform and its inverse, the propagating wavepacket now can be expressed in

$$\Psi(\mathbf{r}, t+dt) = \Big[e^{-i\hat{H}_r\frac{dt}{2}}\mathcal{F}^{-1}\Big[e^{-i\hat{H}_p dt}\mathcal{F}\Big[e^{-i\hat{H}_r\frac{dt}{2}}\Psi(\mathbf{r}, t)\Big]\Big]\Big] + \mathcal{O}(dt^3) \tag{6}$$

The split-operator method is popular for many applications in PDE solving as it is easy to implement and gives fast and accurate solutions for few-atom systems. Its notable disadvantage is error accumulation over a large number of time steps, which means it cannot be used for long propagations.

2.1.2 Multiconfiguration Time-Dependent Hartree Method

The MCTDH algorithm was invented by Meyer, Manthe, and Cederbaum in 1990[66]. It is a sophisticated theoretical framework in quantum mechanics designed to address complex systems with multiple degrees of freedom (DOFs). MCTDH is numerically exact and provides a systematic approach to solving the TDSE. In MCTDH, the wavefunction for an f-dimensional system is expanded in terms of a sum of Hartree products of time-dependent orbitals ($\{\varphi(q,t)\}$), the so-called single-particle functions (SPFs). Each configuration is weighted by a time-dependent coefficient $A(t)$. An MCTDH wavefunction then reads

$$\Psi(\mathbf{q}, t) = \sum_{j_1}^{n_1} \dots \sum_{j_f}^{n_f} A_{j_1 \dots j_f}(t) \prod_{\kappa=1}^{f} \varphi_{j_\kappa}^{(\kappa)}(q_\kappa, t) \ , \tag{7}$$

where $\mathbf{q} \equiv (q_1, q_2, \dots, q_f)$ and n_κ represents the number of SPF for the κth DOF. The total number of configurations is thus given by $\prod_{\kappa}^{f} n^{(\kappa)}$. Each of the j_k SPFs is, in turn, expressed in a primitive time-independent basis, $\chi(q)$. For the κth DOF, the φ_j is written as

$$\varphi_j^{(\kappa)}(q_\kappa, t) = \sum_{\mu=1}^{N_\kappa} c_{j,\mu}(t)\chi^{(\kappa)}(q_\kappa) \ , \tag{8}$$

where N_κ is the number of grid points associated to the κth DOF. In MCTDH, discrete variable representation (DVR) functions[67] are used as a primitive basis. Using the Dirac-Frankel variational principle $\langle \delta\Psi | \, H - i\partial_t \, |\Psi\rangle = 0$ and the above ansatz (Equation 7), the MCTDH equations of motion (EOM) for the SPFs and the A-coefficients are derived.

3 The Fourier Neural Operator Method

Neural Operator (NO) approaches[40,41,68] aim at learning trajectories from data generated by solving a PDE for different combinations of initial conditions and parameters. The operator approach is based on the universal operator approximation theorem[69], which states that neural networks with a single layer can approximate any nonlinear continuous operator. Thus, for an input function space $\mathbf{I}$ and an output function space $\mathbf{U}$ the operator theorem states that there exists a neural operator $\mathbf{G}_\theta$ such that

$$\mathbf{U} = \mathbf{G}_\theta(\mathbf{I}) \ , \tag{9}$$

where θ denotes the neural network weights learned during the training process. Such a neural operator is then trained using the input and output data that represent $\mathbf{I}$ and $\mathbf{U}$, respectively. In this sense, the neural-operator approach learns the relationships from data without explicitly involving the equations of the underlying PDE at any point of the training process and requires only tuples of input and outputs in data.

The FNO architecture is a special case of the NO framework and both of them contain the same three basic components, as shown in Figure 1. The input $\mathbf{I}$ is created by concatenating the wavepacket density and the potential, along with the grid information into a tensor object (Panel (a)). First, a linear neural layer transforms the input into a higher dimension, denoted as the lifting layer in Figure 1 (b). Such a transformation enables the capture of patterns and relationships in data and is also used in frameworks such as convolutional neural networks (CNNs)[70]. The operator layers of the neural operator also serve this purpose, enabling the capture of relationships on different levels. The different operations in an FNO (or equally, a NO framework) aim at learning patterns from data using transformation operations that aggregate local/global information from different regions of the input tensor. In the case of a PDE such information is not as easy to aggregate, and therefore special layers are introduced for efficient learning in the hidden layers following the linear input layer.

The hidden layers of the neural operator framework are constructed using representations that are updated using a kernel-based approach. These layers are depicted as the Fourier layers in panel (b) of Figure 1. In the lines that follow, it is presented how one arrives at the Fourier layer from a neural operator layer.

Mathematically, the output $a^{(l)}(x)$ of the l-th hidden neural operator layer is written as[68]

$$a^{(l)}(x) = \sigma(\mathcal{W}_\theta^{(l)} a^{(l-1)}(x) + \mathcal{K}_\phi^{(l)} a^{(l-1)}(x)), \tag{10}$$

where σ is a non-linear activation function, while $a^{(l-1)}$ represents the output function from the previous layer. $\mathcal{W}$ is a linear operator represented by a layer similar to the linear layer that transforms the input value linearly, $\mathcal{K}$ is a non-local integral transformation operator expressed using a neural network, which henceforth is termed as the kernel integral operator depicted as the yellow block in the Fourier layer in panel (c) of Figure 1. θ and ϕ represent the adjustable neural network parameters in $\mathcal{W}$ and $\mathcal{K}$, respectively. A non-local operator, $\mathcal{K}$ in this case, performs an operation or transformation that considers not only the local data point during its operation but the entire data set by virtue of integration. Li et al.[68] show that the kernel integration operation for the kernel $\mathcal{K}$ over the coordinate space D is

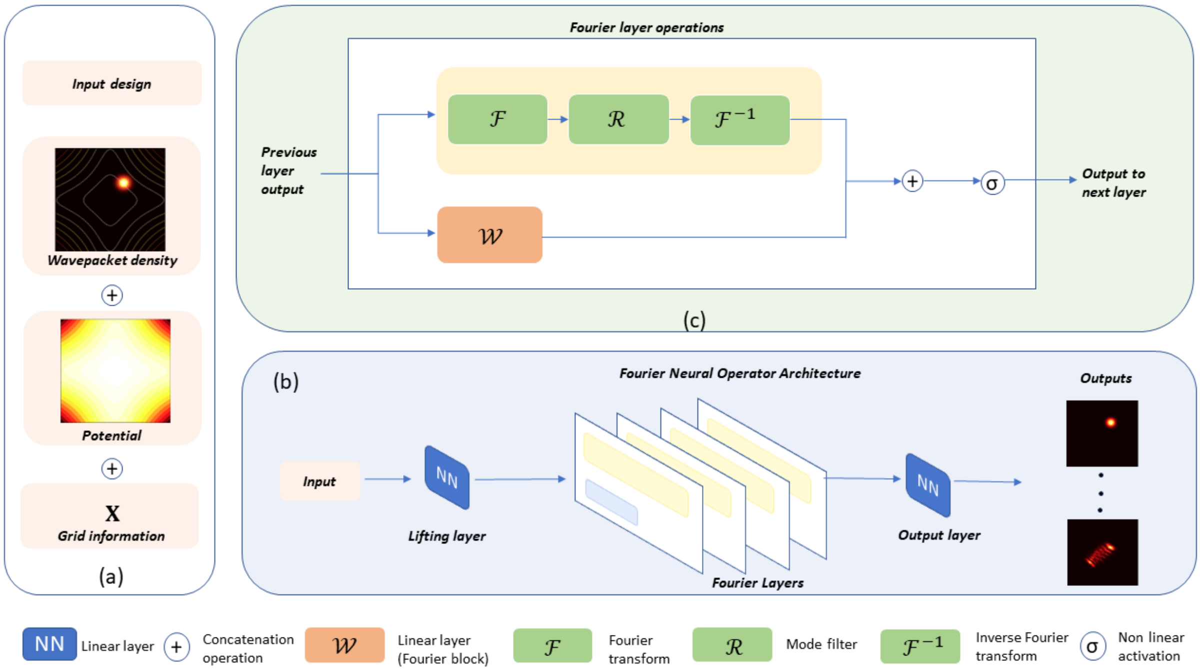


Figure 1: Flow chart of the machine learning of wavepacket density propagation using the FNO model. Each input instance is a concatenation of the wavepacket density at an initial time $t = 0$ a.u., along with the static potential and the grid information (a). The entire FNO framework is depicted in (b). The FNO consists of several Fourier layers, the architecture of one of them is shown in (c). Given the input (a) the model learns to predict the entire simulation for the T time steps in the output.

defined as

$$\mathcal{K}_\phi^{(l)} a^{(l-1)}(x) = \int_D \kappa_\phi^{(l)}(x,z) a^{(l-1)}(z) dz \ , \tag{11}$$

where $\kappa^{(l)}$ is the neural network layer with learnable parameters ϕ. A key differentiator for the FNO compared to other operator-based approaches is the Fourier layer shown in the upper part of Figure 1 (c).

The integration in Equation 11 can be performed using a Fourier transform giving rise to the Fourier layers of the FNO.[68] Let $\mathcal{F}$ be the Fourier transform of a function $f(x)$ from the x domain to the k domain and $\mathcal{F}^{-1}$ its inverse.

The kernel integral operator can then be written as

$$\mathcal{K}_\phi^{(l)} a^{(l-1)}(x) = \mathcal{F}^{-1}\left\{\mathcal{F}\{(\kappa_\phi^{(l)} \cdot a^{(l-1)})(x)\}\right\} = \mathcal{F}^{-1}\left\{(\mathcal{F}\{\kappa_\phi^{(l)}(x)\} \cdot \mathcal{F}\{a^{(l-1)}(x)\})\right\} . \tag{12}$$

In practice, FNO models use the Fast Fourier transform method to perform this integration. Mathematically, a Fourier series expansion is infinite, however, practical implementations require the termination of the series after a few terms, defined as the modes of the Fourier series expansion. The maximum number of modes is a user-defined hyper-parameter (for each Fourier layer) defined before training the model. In the implementation of the Fourier layer, this pre-defined number of modes helps to truncate the Fourier series. The operations of the Fourier layer are thus summarised as the Fourier transform, filtering of these Fourier modes, and the inverse Fourier transform as depicted by the F, R, and F^{-1} blocks in panel (c) of the Figure 1, respectively. Since Fourier transforms are more suited to periodic conditions, the linear layer $\mathcal{W}$ ensures that the grid conditions of the model are also learned and added before the output is forwarded to the subsequent layer. Iterating through several FNO layers allows the model to learn local and global features. This multi-level learning approach is also found in frameworks that use CNNs for various ML tasks. To finally transform the output to the same dimensions as the desired output, a linear transformation layer, such as the output layer in Figure 1(b), is employed.

The advantage of FNOs is the use of Fourier layers, which enable learning input-output relationships while keeping the computational cost of integration relatively low[68]. Another notable advantage is its independence of the resolution of the grid space, with the models being capable of giving one-shot predictions (in this case, one-shot prediction means the model outputs the entire propagation at once) at a higher resolution than the data that they were trained on. In summary, the FNO approach involves learning the relationship between input instances in the time step range from $t = 0$ to $t = t_{train}$ and the propagations from time $t = t_{train+1}$ to time $t = t_{sim}$ output data. Thus the FNO model can also be trained to predict the entire propagation governed by a PDE, given the initial state at time $t = 0$.

4 Quantum Dynamics Test Systems

To investigate the ability of FNO models to accurately capture the dynamics of different quantum systems, FNO models were trained to approximate propagators for two test systems. The first system involves a wavepacket randomly placed on a spatial grid propagating in a time-independent two-dimensional potential. The second system involves propagating a fixed initial wavepacket under the influence of different time-dependent potentials. Both systems are outlined in detail in this section.

4.1 Propagation on a 2D Anharmonic Potential

The first model system (aka family of PDEs) has a two-dimensional anharmonic potential surface, on which a wavepacket undergoes propagation. The TDSE for this system (in atomic units) is given by

$$i\frac{\partial}{\partial t}\Psi = -\frac{1}{2}\left(\frac{\partial^2}{\partial q_1^2} + \frac{\partial^2}{\partial q_2^2}\right)\Psi + \frac{1}{2}(q_1^2 + q_2^2)\Psi + \lambda q_1^2 q_2^2 \Psi \ , \tag{13}$$

where the parameter λ is varied in the range $(0, 0.15)$ a.u. to generate a set of anharmonic potentials, and subsequently a set of Hamiltonians along the two coordinates q_1 and q_2. The initial wavepacket, in this case, is chosen to be a normalized product of two Gaussians centered at random positions on the grid.

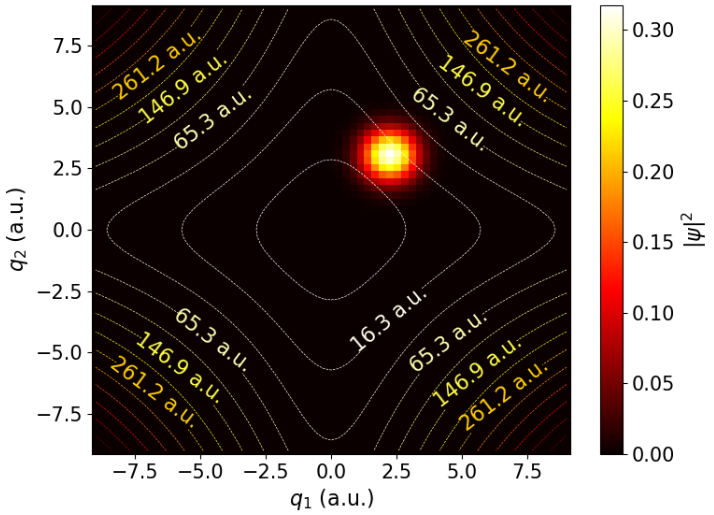


Figure 2: Contour plot of an anharmonic potential with an initial wavepacket density $|\Psi|^2$ as a function of two spatial coordinates q_1 and q_2. The contour lines indicate the value of the potential in a.u. at various positions in space, with the minimum being at the center.

4.2 Laser-Driven Tunneling in a 1D Double Well

The second system models the rotation of the molecule $F_2H_3C_6 - C_6H_3Br_2$ along the torsional bond connecting the two phenyl rings, as represented in Figure 3(a). In line with the approximations discussed in previous works[52,54], this rotation can be modeled as the propagation of a 1D wavepacket in a double-well potential along the dihedral angle coordinate (ϕ_d). Under the influence of a laser, the target is to tunnel the wavepacket initially localized on the left side (blue) through the barrier and into the right side of the double well (pink), as indicated in Figure 3(b).

The steady-state Hamiltonian of this system, in the absence of an external field, is

$$\hat{H} = \frac{-\hbar^2}{2I_{rel}} \frac{\partial^2}{\partial \phi_d^2} + V(\phi_d) \ , \tag{14}$$

where

$$V(\phi_d) = \sum_{n=0}^{6} A_n \cos(n\phi_d) \tag{15}$$

is the double well potential in which the initial wavepacket is trapped. It has been numerically fitted in previous studies to a potential curve obtained using density functional theory calculations[52,55]. I_{rel} is the relative moment of inertia calculated as $I_{rel} = I_{Br}I_F/(I_{Br} + I_F)$. The rotation of this molecule along the axis can be realized using an external laser field aligned along the axis, giving an extended Hamiltonian

$$\hat{H}' = \hat{H} - \frac{1}{4}\epsilon^2(t)\alpha(\phi_d), \tag{16}$$

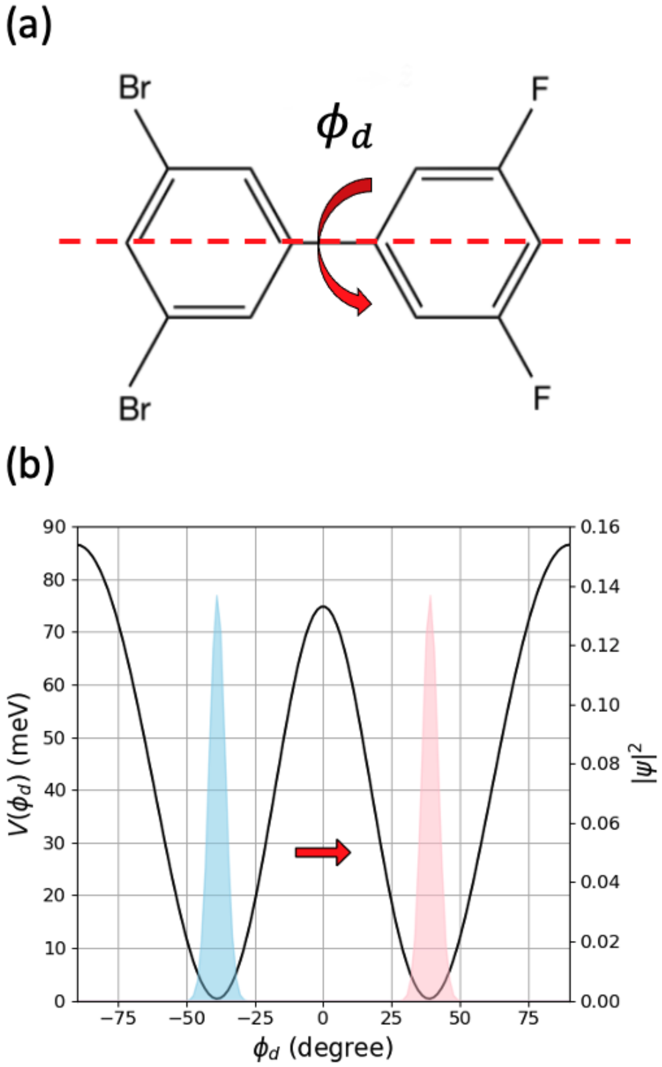


Figure 3: (a) Visual representation of $F_2H_3C_6{-}C_6H_3Br_2$ with the dihedral angle ϕ_d indicated. (b) The 1D potential model of the rotation shows a wavepacket in the lowest state on the left side (blue) and the target state of the rotated molecule (pink), which is the lowest state on the right side.

where the induced electric field due to the applied laser is given by ϵ, as demonstrated by[52], which is applied in the time window $t = 0$ to $t = T$. The term $\alpha(\phi_d)$ is the molecular polarizability term, the functional form of which is provided in Thomas and Henriksen[54]. The Hamiltonian additionally contains perturbations calculated using the technique elaborated in Thomas and Henriksen[52] to realistically represent vibrations in the molecule. A laser pulse-induced electric field applied to the perturbed wave packet can have several effects on this system, ranging from oscillations within the left side of the well without crossing, tunneling across the well, or a complete dissociation due to a large field. As a result, optimization techniques are necessary to design a pulse shape that ensures the tunneling to the target state.

5 Pulse Optimization

As discussed in the previous section, the 1D system, which involves molecular rotation along its dihedral angle (ϕ_d), is an application of laser-driven quantum dynamics. The design of a laser pulse in this double-well system, such that the wavepacket can optimally tunnel the rotational barrier, is an active research problem, usually tackled with optimal or coherent control methods[56–60,71–73]. As an alternative to these methods, laser pulses can be designed using a combination of neural networks and genetic algorithms[52]. While a crucial advantage of the neural network method is that it does not rely on a pre-defined functional form of the laser, it requires training several neural networks to get an accurate wavefunction.

In this work, an attempt is made to use Bayesian optimization for shaping a laser pulse[74,75]. As a pre-requisite, a functional form for the laser pulse is defined as this allows for control of the range of parameters of the pulse (and thus the pulse shape). Defining a functional form can be useful, as it can eventually be used in an experiment. As a design choice, a chirped pulse with the functional form

$$\epsilon(t) = Ae^{-\frac{1}{2}(\frac{t-t_c}{\sigma})^2} cos(2\pi(f + \beta(t - t_c))(t - t_c) + \theta) \tag{17}$$

is selected, where A is the amplitude of the pulse at time t, t_c is its central time, and σ controls the width of the Gaussian envelope. The chirp rate factor β determines how quickly the frequency f changes in time. θ is the phase offset, which can be used to control the phase of the pulse in time. Hence, six variables control the chirped pulse used in the propagation of the wavepacket that can be optimized to bring it from the left side of the double well to the right.

However, in light of the computational expense of the optimization process over these parameters (see next Section), we restrict ourselves to three variables. For instance, the laser has to be physically meaningful and therefore the amplitude for the pulse is set to be in the range of 0 to 0.024 a.u (0 to 20 TW/cm^2). We similarly fix the range for two other parameters, frequency f and pulse width σ, to $[0.05, 0.10]$ a.u. and $[1, 3]$ a.u., respectively, giving us a search space of three parameters suitable for demonstrating our approach's viability. These parameter ranges were chosen after a few test runs to check the propagation's final target state. The other parameters are kept fixed: t_c to a value of 0 a.u., β to 0.1 a.u., and the phase shift factor θ is chosen to be π for the sake of these calculations. For the pulse optimization process, a final target state is chosen. Since the FNO models in this work employ densities, we choose a cost function C such that it minimizes the Frobenius norm between the target density of the final state and the predicted one. The best pulse obtained from such an optimization gives maximal overlap $C(A, \sigma, f)$ between the state obtained using a laser and the target reference state.

5.1 Markov Chain Monte Carlo

The Markov chain Monte Carlo (MCMC) technique is employed to estimate the initial input parameters of the electric field used to drive the rotation of the molecular system described in the section above. MCMC is a Bayesian optimization procedure that helps find an optimal set of parameters through a sampling process. Traditionally, optimization processes have involved deterministic techniques such as calculus, which systematically search for optimal parameters using mathematical principles. Bayesian optimization, on the other hand, involves methods that help extract an optimal set of parameters from a large distribution of input variables. Among those, Monte Carlo sampling methods are a class of Bayesian methods that can be used to generate meaningful samples for such optimization problems. Markov chain Monte-Carlo sampling[76] introduces dependencies in the sampling procedure using Markov chains, thus ensuring that the selection (rejection) of a previous sample influences the sample space that ultimately models the posterior distribution of variables. MCMC methods can also be used as tools for optimizing functions[62,68,77]. In the case of laser-pulse optimization, MCMC methods can be combined with a method to calculate the cost function, which compares an optimized and a target state for a set of laser input parameters to give those parameter sets that maximize the said cost function. The main steps in this process are discussed below:

1. **Random Sample Selection:** The MCMC process begins by randomly selecting initial parameters from the prior distribution of variables. For the task of optimizing the parameters of the laser pulse, the prior distribution for each variable is chosen to be a Gaussian distribution.

2. **Forward Step:** After selecting initial parameters, one calculates the cost function for the chosen parameters. For the first sample, this value is used to evaluate a proposed distribution. This step involves utilizing the FNO model or the split-operator method.

3. **Metropolis-Hastings (MH) step[78]:** In this step, one decides whether to accept the new sample, denoted as x', based on the MH selection criterion defined through

   $$\text{Accept new sample } x' \text{ if } r \leq \min\left(1, \frac{C(x')}{C(x)}\right), \text{else reject.}$$

   The MH algorithm compares the likelihood of the new parameter values to the likelihood of the current Markov state. If the proposed values improve the likelihood, they are accepted and added as the next parameter set for the Markov chain evolution. However, even if the likelihood does not improve, there is still a chance of acceptance, which is determined probabilistically. If the decision is made to reject the new sample, the old sample becomes also the current sample. The Metropolis-Hastings step ensures that the Markov chain explores the parameter space while respecting the target distribution.

Repeating steps 1-3 iteratively results in a distribution of target parameters for C with respect to the input variables. Depending on the choice of the prior distribution and the complexity of the cost function, convergence to a stable distribution may take some time. Initial non-convergent samples, typically constituting around $0-20\%$ of the total samples, are referred to as "burn-in" and are often discarded. Once convergence is achieved, the remaining set of samples can be used to infer various properties of the distribution, such as the mean and the mode. To facilitate the analysis of the samples generated using MCMC, kernel density estimation (KDE)[79] is employed. KDE generates a smooth, continuous representation of the target distribution, making its analysis and interpretation easier, i.e. here it can help identify the combination of parameters that yields the optimal value of the cost function (or other key characteristics of interest). A key roadblock that often hinders the use of Bayesian optimization techniques such as the MCMC described above, is the step of evaluating the function (calculations that lead to the cost function evaluation) for samples. It is therefore advantageous to use models that can quickly evaluate this step and perform MCMC for a large number of samples.

6 Dataset Generation and Model Training

To train the FNO models for the two systems, data is generated using the split operator (for the 1D system) and the MCTDH (for the 2D system) method. The ML task for the 2D case is to use the dataset of 1000 simulations to train a model that, when given an initial wavepacket and an anharmonic potential as input, can generate the simulation for a fixed number of time steps. To generate the dataset for training these simulations, Gaussian wavepackets are randomly placed on a 2D grid. The potentials of datasets are also different by varying the parameter λ of Equation 13 in the range $(0, 0.15)$ a.u. The dynamics of the system are simulated for each combination of potential and initial wavepacket for 51 time steps within $t = [0, 10]$ a.u.

For the 1D system, the potential is a combination of the double well and the induced electric field. The initial wavepacket for this system is determined as the lowest eigenstate of the double well obtained from the Fourier grid Hamiltonian method, which is located within the left potential well. The final dataset consists of 4,000 different samples of wavepackets propagating in different laser-dressed potentials obtained by varying the laser parameters in the constrained manner introduced above.

The two systems of different dimensionality require training two different FNO architectures. Specifically, the 1D double well system is trained using the FNO-2D (accounting for one spatial and one temporal coordinate) model, and the FNO-3D (to model two spatial and one temporal coordinate) model is used for training the two-dimensional case. In both cases, the input for the FNO model contains a concatenation of the initial wavepacket density, potential (at all time steps), and grid information. The models are optimized using the standard ADAM optimizer on a single NVIDIA A100 40GB GPU. Table 1 summarizes the various hyperparameters used for training the model. The models are trained to minimize the H1 loss, while the accuracy of the predicted propagations with the reference data is measured using both the mean square error (MSE) loss and the H1 loss. The H1 loss, also known as the Sobolev loss[80,81], is a specialized loss function tailored for the approximation of continuous functions. It emphasizes solutions that not only closely match the given data but also demonstrate a high degree of smoothness. To achieve this, it penalizes abrupt or discontinuous variations between consecutive instances by considering gradients between them. This loss function is particularly valuable in applications such as PDE solving and image denoising, where the promotion of smooth and continuous solutions is essential[80]. Hence we train FNO models to minimize the H1 loss, for all time steps of the propagation.

Table 1: Hypaparameters for the FNO models for the 1D and 2D system

Metric	1D	2D
Size of training set	4,000	1,000
Grid size	128	65 x 65
Number of time steps	150	51
Number of spatial Fourier modes	75	20, 20
Number of temporal Fourier modes	64	20
Learning rate	10^{-4}	10^{-2}
Number of FNO layers	4	4
Width of layers	128	40
Trainable model parameters (million)	629	205
Training epochs	300	50
Time step length (a.u.)	0.048	0.200

For demonstrating laser-pulse control using FNOs, the final state of the wavepacket propagation for different laser pulses is evaluated using the FNO model. The cost function is then calculated using these wavepackets as the final states. The cost function that maximizes the overlap between the target and final states is defined as $C = 1 - \text{MSE}(\text{final}, \text{target})$. To provide a fruitful comparison of the speedup we also carry out a comparative test of the MCMC process with the split-operator code that was used to generate the data. Moreover, since Bayesian sampling processes can be affected by the choice of the initial random seed, to demonstrate the stability of our results, the MCMC process is carried out for 20 different seed values in the case of the FNO model. For each seed, the MCMC sampling process runs over 5,000 iterations to converge to a set of parameters that give the largest overlap. 2,000 samples are discarded in the beginning as burn-in samples. Having removed these iterations, the final set of parameters is obtained as their temporal mean.

7 Results and Discussion

In this section, the ability of a trained FNO model to simulate wavepacket density propagation is first analyzed. It is then shown how an FNO model can be used as a low-cost surrogate option for finding an optimal electric field to drive a molecular rotation.

7.1 Learning Wavepacket Propagation with FNOs

Our first goal is to demonstrate that the FNO models trained for the test systems can faithfully produce wavepacket propagation for new instances of data. The dataset for the 1D wavepacket system always involves propagating the same initial wavepacket in different potentials that arise from the application of different laser pulses in each sample. This problem is challenging since the changes in the total potential lead to varying behavior in the simulations. For example, a weak laser pulse might induce oscillations in the wavepacket, but fail to drive the motion of the wavepacket across the barrier. Or a very strong laser might drive the wavepacket to the right side of the double-well, but also crossover from the

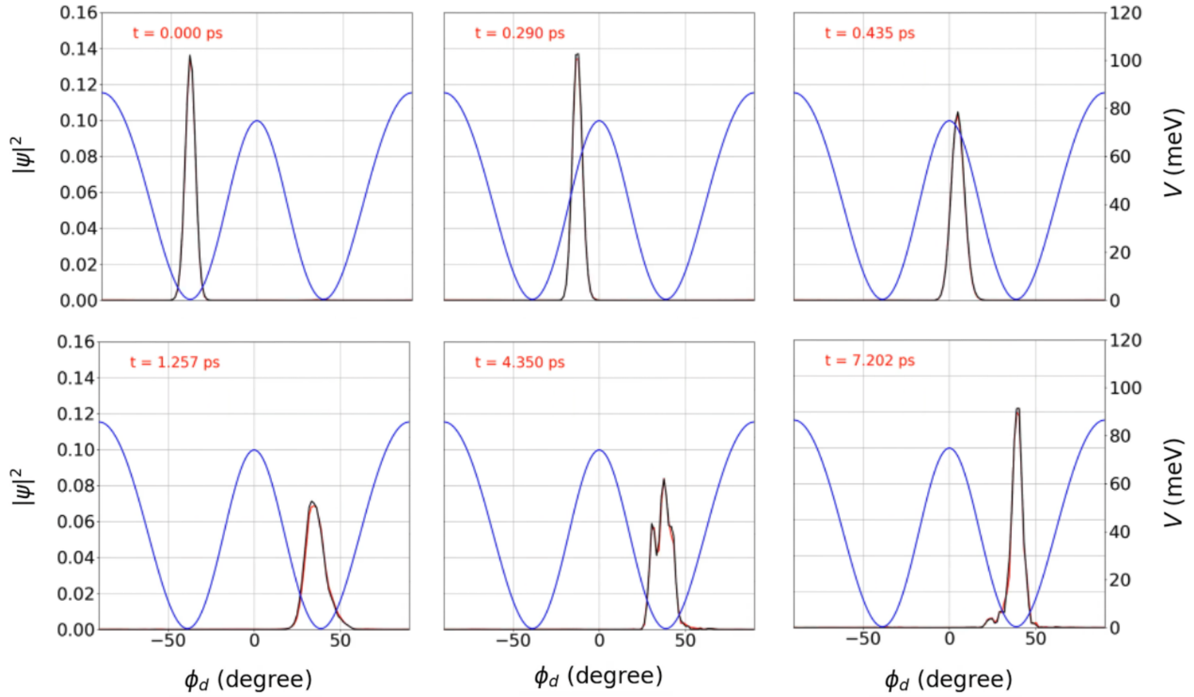


Figure 4: Comparison of the wavepacket densities obtained from the split-operator method (black) and the FNO model (red) for the laser-driven tunneling in a double-well potential (blue) shown at different time steps of a propagation ending at $T = 7.25$ ps.

second barrier as well. Furthermore, some laser pulses can induce the desired behavior of a crossover to the right side of the double well, but not result in a steady state. In such a case the final wavepacket can attain a more or less oscillatory motion. In Figure 4 the wavepacket propagation of a randomly chosen test instance is shown at different time steps. The split-operator-generated propagation (in black) is compared to the FNO-generated propagation (in red). The laser pulse is applied to the system during the entire propagation, starting at t=0, when the wavepacket density is localized at the left side. The electric field induced by the laser imparts momentum to this wavepacket thereby driving its crossing over the rotational barrier, as demonstrated for the next two time steps $t = 0.290$ and 0.435 a.u. After hitting the final boundary on the right side (not shown), the wavepacket scatters back to and is trapped within the confinement of the right well, existing as a superposition of the eigenstates of the excited wavepacket. The H1 loss for the propagation of this randomly chosen sample is 0.0854 a.u.$^{-1}$. As seen in the figure, the FNO model accurately reproduces the dynamics of the system under the influence of the time-varying laser field.

The 2D wavepacket system involves fewer variations than the 1D system since the potential is time-independent, but the dataset has variations in the form of the position of the wavepackets on the spatial grid, and the differences introduced in the potentials by varying the parameter λ of Equation 13. For low values of λ, the system acts as a harmonic oscillator, and the wavepacket oscillates back and forth in the potential well. For higher values concerned with a stronger anharmonicity, while the wavepacket oscillates back and forth, it

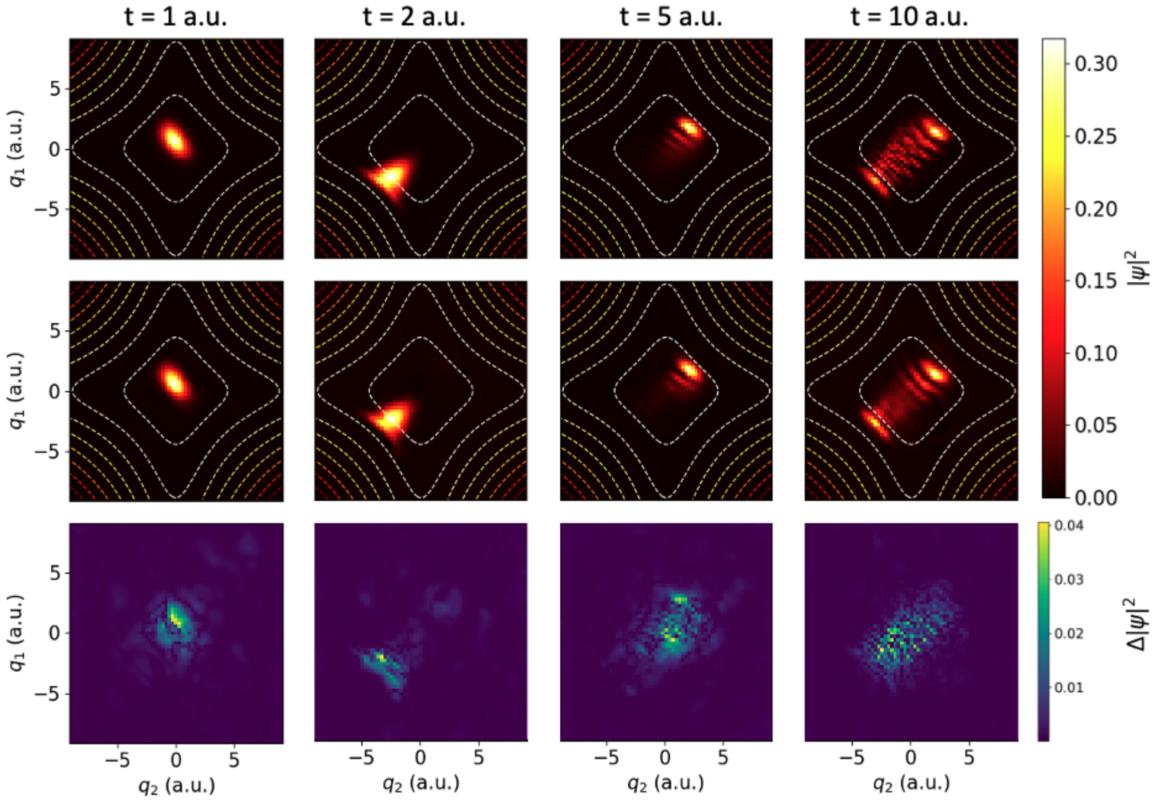


Figure 5: Comparison of the MCTDH and FNO model generated wavepacket densities shown at different time steps. The top panel shows the subsequent densities at $t = 1$, $t = 2$, $t = 5$, and $t = 10$ a.u. generated using MCTDH. The middle panel shows the corresponding graphs from the FNO model. The bottom panel depicts the absolute difference between the wavepacket densities obtained from the two calculations.

loses its shape, as it dissipates into unbound states. An ML model that predicts propagation for these wavepackets, therefore, must account for these variations.
Figure 5 shows a comparison between the MCTDH-generated propagation for a randomly chosen wavepacket from the test data and the prediction of the FNO-trained model for the density of the same wavepacket. The initial wavepacket density for this propagation is the same as depicted in Figure 2. Wavepacket densities arising from the propagation using the MCTDH method are reported for four different time steps in the top part of the figure. The Gaussian wavepacket centered initially at positive and identical $q_1 = q_2 = 2.5$ a.u. is following the steepest descent of the potential and reaching the potential minimum at about $t = 1$ a.u. (leftmost panel). It continues to the turning point, from which it is scattered back at $t = 2$ a.u., experiencing here already a widening and deformation of the Gaussian shape. While oscillating back and forth during the remaining time, it further broadens and starts to establish interference patterns with its contributions of different velocities ($t = 10$ a.u.).

The second row of Figure 2 shows the FNO predicted propagation when provided with the same initial wavepacket density. The absolute difference in the densities is shown in the bottom part of the figure. As is seen in the figure, the model accurately reproduces all the major features of the wavepacket at different time steps. While in general, the error is more concentrated in the areas where the wavepacket motion itself takes place, it is difficult to ascertain whether errors in the propagation follow any deterministic pattern that varies in time or position. The average H1 loss of the given example for the duration of the propagation is 0.2184 a.u.$^{-1}$.

Table 2 summarises the training results for FNO models for both datasets. The average test loss for the FNO model for the 2D system is 0.25 a.u.$^{-1}$, which is close to the value of 0.22 a.u.$^{-1}$ for the sample above, giving a perspective on the average quality of the propagation ability of the FNO model. On the other hand, the average test H1 loss of the 1D model is 0.154 a.u.$^{-1}$ compared to the value of 0.08 a.u.$^{-1}$ for the sample in Figure 5. The sample shown, therefore, is one of the better-performing samples of the test set. The larger size of the training set in the 1D system, the larger model size, and the higher number of epochs as compared to the model for the 2D system result in a higher training time for this model, which is a few hours rather than only minutes.

The results demonstrate the ability of the FNO method to generalize over large instances of simulations in the 1D and 2D systems and learn the physics of both model systems. The speed of inference for predicting propagations for new instances once the model is trained is one of the biggest advantages of using the FNO models for PDEs. This is reflected in the inference time values of 0.011 s and 0.036 s for the two FNO models for the 1D and 2D systems, respectively. In contrast, evaluating the propagation for a 2D wavepacket sample using MCTDH requires 19.15 s (calculated on 1 node, 16 cores per node, 4GB per node). The FNO model therefore can generate samples faster than the MCTDH method for this particular system. This high speed of inference gives ML models an advantage over conventional PDE solvers for downstream applications involving iterations over several samples.

Table 2: Training results for the FNO model for the 1D and 2D system

Metric	Value (1D)	Value (2D)
Training time	4 h 51 m	26 m 22 s
Inference time	0.0111 s	0.0363 s
Average H1 test loss	0.15 a.u.$^{-1}$	0.25 a.u.$^{-1}$
Average MSE test loss	$6.50 \cdot 10^{-3}$	$3.41 \cdot 10^{-7}$

7.2 Laser pulse control using FNOs and MCMC

One of the main goals of this work is to show that trained FNO models can be used along with the Bayesian technique of MCMC as a low-cost surrogate model to fine-tune the laser pulse shape and with that achieve a desired propagation outcome. In the case of the present 1D system, the optimal electric field drives the tunneling of the wavepacket across the double-well barrier. MCMC optimizes the set of parameters that maximize the overlap between the final and target state.

Figure 6(a) shows the results of the MCMC optimization process for 20 representative simulations, by showing how the resulting pulse intensities vary with time. While the largest

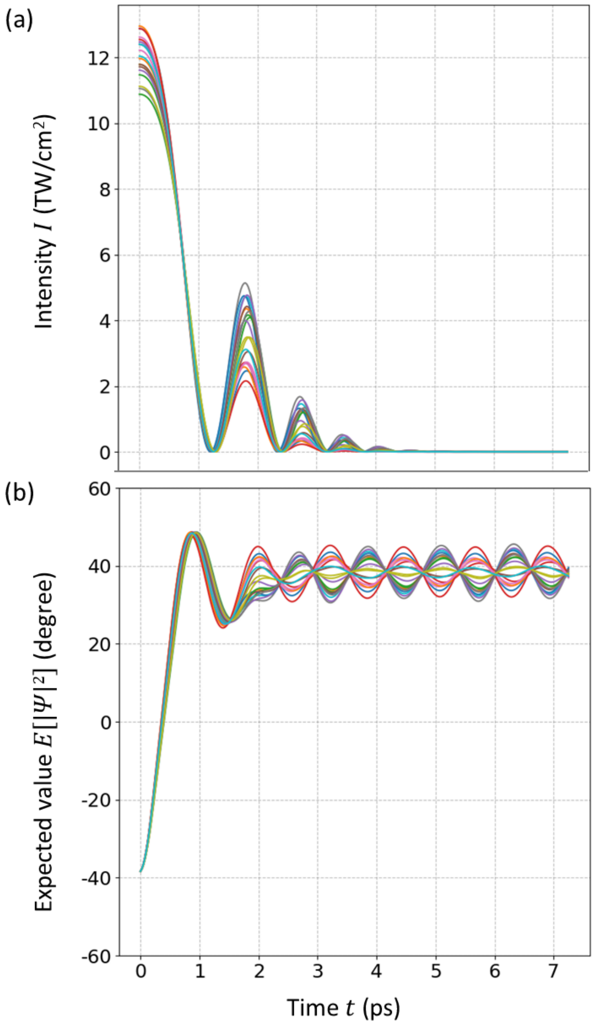


Figure 6: Time-resolved results of 20 MCMC pulse optimization tasks using the FNO 2D model. The top panel (a) shows optimized 20 pulse shapes, all with a similar pattern. The bottom panel (b) shows the expected values of the wavepacket density for the 20 seeds. In the beginning, all the wavepackets are localized in the left well, indicated by the negative values, while applying the pulse transports them to the right during the first laser cycle.

pulse contributes to the actual tunneling, the subsequent pulses stabilize the wavepacket in the second well, as discussed in Thomas and Henriksen[52]. These results also agree with this previous work, in which the authors obtained similar pulse shapes, albeit here we also attain a functional form for the pulse, which might prove to be useful for experimental investigations. Figure 6(b) shows the expected values of the wavepacket density at each step of the simulation, with negative values indicating that the wavepacket is localized on the left side of the double well, and the values turn positive once the wavepacket crosses over to the right. As seen in the figure, the FNO MCMC approach can successfully generate pulses to induce the sought behavior for the target state. The low inference times of the FNO model thus make the FNO MCMC combined approach a useful technique for solving inverse problems in quantum dynamics.

We find that of the 20 MCMC runs, the final cost function values concerning the target state lie between 0.63 and 0.95, with the average being 0.85. The variation in these C can be attributed to two causes: 1. Randomness of the seed and 2. The kernel density estimation (KDE) technique. The first is a natural byproduct of the Bayesian process which initializes using a randomly sampled point from the prior but this can be minimized by increasing the number of MCMC samples from 5,000 to higher values, at the expense of increased computation times. Regarding the second, the KDE process employed in this work uses a 'thinned' posterior, i.e. a uniform selection of values after the burn-in samples have been discarded. For the sake of speed, the number of samples for KDE is kept to 40, which ensures that the simulations complete quickly ($\leq$ 10 minutes) per run on the same computing architecture that is used to train and run the model. When comparing one run of the MCMC with FNO to the MCMC split operator approach, the FNO method is found to be twice as fast. This, of course, is not conclusive, as the split operator method we use could further be optimized for computational efficiency, but the exercise is indicative of the ease and speed of the usage of the FNO method.

8 Conclusions and Future Perspectives

Machine learning approaches have been used successfully in many problems of theoretical chemistry, but there are still very few applications to time-dependent phenomena that involve solving the TDSE. We show here that an ML technique, namely the Fourier neural operator method, can faithfully and accurately reproduce the time evolution of wavepackets. To robustly test the framework's capabilities of learning the physics of the underlying systems, datasets were generated, in which variations are introduced in the potentials as well as the initial wavepackets in a way that they cover a large space of simulation possibilities. The FNO-trained models for wavepacket propagation are capable of single-shot inferences of entire simulations at a fraction of the time and costs associated with traditional differential equation solvers. One, of course, has to note, that the FNO requires data from these solvers and training resources to do so. Furthermore, methods such as MCTDH are optimized for systems with several degrees of freedom, the performance (training plus inference) of which has not been compared to FNO models here, yet.

Additionally, it was shown how the quick inference ability of FNOs can be applied to solve inverse problems using the Bayesian Markov Chain Monte Carlo technique. Since the MCMC method requires sampling over a large number of simulations, using a traditional, slower method is not feasible for such an application. FNOs, however, can be used as low-cost surrogate models to quickly iterate through large sample spaces and give an estimate of the optimal parameters. As demonstrated in this work, the FNO MCMC method can give an estimate of the best laser pulse to maximize the overlap between the final state with a target state at a lower cost than even the split-operator method combined with MCMC. Given the low inference times of the FNO method, such optimization will be able to solve problems in larger dimensions as well, and the computational gains there will possibly be even larger than the ones demonstrated here. While the MCMC approach is demonstrated for a relatively conservative search space of three input variables of the laser, using a larger model and dataset can allow for training FNO models and performing the MCMC on a larger search space with more variables. Training larger FNO models could necessitate parallelization over several GPUs, which to the best of our knowledge has not been implemented yet. Subsequent projects will aim to utilize parallelization to implement larger models with less conservative search subspaces for creating a dataset.

The models trained in this work employ wavepacket densities to represent the time evolution of dynamical systems. While models can be trained on the wavepacket values themselves, it amounts to an increase in the dimensions of both the data and the FNO architecture. In our explorations for training wavepackets on their complex-valued functions, we found the results to be poorer for the 1D system, compared to models that employed densities. This is an aspect that certainly deserves more attention as models trained on wavepacket values open up new avenues for applications. For example, more accurate physics-inspired neural operator models cannot be employed for our models using densities, since physics-based loss terms in these models require wavepacket values rather than densities, thereby making those models more accurate[48].

Lastly, the highest dimensions that have been learned involve three spatial and one temporal coordinates, and we are not aware of FNO models operating for higher-dimensional data. Training these models should be possible without major changes in the architecture of the FNO model. However, computational expenses for generating the data and training can be prohibitive for larger systems and our subsequent work will aim to explore optimal ways in which both these issues can be handled. We hope that our work opens up further investigations into how the FNO framework can be used to push forth investigations in quantum dynamics and help drive meaningful applications of this field to larger systems.

9 Data availability

The code used to train the model as well as the datasets will be made publicly available post review.

10 Acknowledgments

KS and AB acknowledge support from the Helmholtz Einstein International Berlin Research School in Data Science (HEIBRiDS). Computing resources for generating data using

MCTDH were kindly provided by the Freie Universit Berlin HPC cluster Curta.[82]. The authors gratefully acknowledge the Helmholtz Data Federation (HDF) for providing services and computing time on the HDF Cloud cluster at the Jülich Supercomputing Centre (JSC)[83].

References

[1] Brown, J.; Carrington, J., Tucker Using an expanding nondirect product harmonic basis with an iterative eigensolver to compute vibrational energy levels with as many as seven atoms. *The Journal of Chemical Physics* **2016**, *145*, 144104.

[2] Sathyamurthy, N.; Mahapatra, S. Time-dependent quantum mechanical wave packet dynamics. *Phys. Chem. Chem. Phys.* **2021**, *23*, 7586–7614.

[3] Beck, M. The multiconfiguration time-dependent Hartree (MCTDH) method: a highly efficient algorithm for propagating wavepackets. *Physics Reports* **2000**, *324*, 1–105.

[4] Peláez, D.; Meyer, H.-D. On the infrared absorption spectrum of the hydrated hydroxide (H3O2-) cluster anion. *Journal of Chemical Physics* **2017**, *482*, 100–105.

[5] Peláez, D.; Meyer, H.-D. The multigrid POTFIT (MGPF) method: Grid representations of potentials for quantum dynamics of large systems. *The Journal of Chemical Physics* **2013**, *138*, 014108.

[6] Otto, F. Multi-Layer Potfit: An accurate potential representation for efficient high-dimensional quantum dynamics. *The Journal of Chemical Physics* **2014**, *140*, 014106.

[7] Schröder, M. Transforming high-dimensional potential energy surfaces into a canonical polyadic decomposition using Monte Carlo methods. *The Journal of Chemical Physics* **2020**, *152*, 024108.

[8] Panadés-Barrueta, R. L.; Peláez, D. Low-rank sum-of-products finite-basis-representation (SOP-FBR) of potential energy surfaces. *The Journal of Chemical Physics* **2020**, *153*, 234110.

[9] Nadoveza, N.; Panadés-Barrueta, R.; Shi, L.; Gatti, F.; Peláez, D. Analytical high-dimensional operators in Canonical Polyadic Finite Basis Representation (CP-FBR). *The Journal of Chemical Physics* **2023**, *158*, 114109.

[10] Leforestier, C.; Bisseling, R.; Cerjan, C.; Feit, M.; Friesner, R.; Guldberg, A.; Hammerich, A.; Jolicard, G.; Karrlein, W.; Meyer, H.-D.; Lipkin, N.; Roncero, O.; Kosloff, R. A comparison of different propagation schemes for the time dependent Schrödinger equation. *Journal of Computational Physics* **1991**, *94*, 59–80.

[11] Conte, D.; Lubich, C. An error analysis of the multi-configuration time-dependent Hartree method of quantum dynamics. *ESAIM: Mathematical Modelling and Numerical Analysis* **2010**, *44*, 759–780.

[12] Larsson, H. A tensor network view of multiconfiguration time-dependent Hartree methods. *arXiv* **2023**,

[13] Schmitz, G.; Godtliebsen, I. H.; Christiansen, O. Machine learning for potential energy surfaces: An extensive database and assessment of methods. *The Journal of Chemical Physics* **2019**, *150*, 244113.

[14] Meuwly, M. Machine Learning for Chemical Reactions. *Chemical Reviews* **2021**, *121*, 10218–10239.

[15] Fooshee, D.; Mood, A.; Gutman, E.; Tavakoli, M.; Urban, G.; Liu, F.; Huynh, N.; Van Vranken, D.; Baldi, P. Deep learning for chemical reaction prediction. *Molecular Systems Design & Engineering* **2018**, *3*, 442–452.

[16] Hansen, K.; Biegler, F.; Ramakrishnan, R.; Pronobis, W.; Von Lilienfeld, O. A.; Müller, K.-R.; Tkatchenko, A. Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *The Journal of Physical Chemistry Letters* **2015**, *6*, 2326–2331.

[17] Cova, T. F. G. G.; Pais, A. A. C. C. Deep Learning for Deep Chemistry: Optimizing the Prediction of Chemical Patterns. *Frontiers in Chemistry* **2019**, *7*.

[18] Noé, F.; Tkatchenko, A.; Müller, K.-R.; Clementi, C. Machine Learning for Molecular Simulation. *Annual Review of Physical Chemistry* **2020**, *71*, 361–390.

[19] Wang, S.; Wang, H.; Perdikaris, P. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science Advances* **2021**, *7*, eabi8605.

[20] Sehanobish, A.; Corzo, H. H.; Kara, O.; van Dijk, D. Learning Potentials of Quantum Systems using Deep Neural Networks. 2021; http://arxiv.org/abs/2006.13297, arXiv:2006.13297 [physics, physics:quant-ph, stat].

[21] Lin, K.; Peng, J.; Gu, F. L.; Lan, Z. Simulation of Open Quantum Dynamics with Bootstrap-Based Long Short-Term Memory Recurrent Neural Network. *The Journal of Physical Chemistry Letters* **2021**, *12*, 10225–10234.

[22] Choi, M.; Flam-Shepherd, D.; Kyaw, T. H.; Aspuru-Guzik, A. Learning quantum dynamics with latent neural ordinary differential equations. *Physical Review A* **2022**, *105*, 042403.

[23] Radu, A.; Duque, C. A. Neural network approaches for solving Schrödinger equation in arbitrary quantum wells. *Scientific Reports* **2022**, *12*, 2535.

[24] Yao, Y.; Cao, C.; Haas, S.; Agarwal, M.; Khanna, D.; Abram, M. Emulating Quantum Dynamics with Neural Networks via Knowledge Distillation. 2022; http://arxiv.org/abs/2203.10200, arXiv:2203.10200 [cond-mat, physics:quant-ph].

[25] Jin, H.; Mattheakis, M.; Protopapas, P. Physics-Informed Neural Networks for Quantum Eigenvalue Problems. **2022**,

[26] Shah, K.; Stiller, P.; Hoffmann, N.; Cangi, A. Physics-Informed Neural Networks as Solvers for the Time-Dependent Schrödinger Equation. **2022**,

[27] Secor, M.; Soudackov, A. V.; Hammes-Schiffer, S. Artificial Neural Networks as Propagators in Quantum Dynamics. *The Journal of Physical Chemistry Letters* **2021**, *12*, 10654–10662.

[28] Temam, R.; Chorin, A. Navier Stokes Equations: Theory and Numerical Analysis. *Journal of Applied Mechanics* **1978**, *45*, 456–456.

[29] Crank, J.; Nicolson, P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Mathematical Proceedings of the Cambridge Philosophical Society* **1947**, *43*, 50–67.

[30] Evans, L. C. *Partial differential equations*, 2nd ed.; Graduate studies in mathematics v. 19; American Mathematical Society: Providence, R.I, 2010; OCLC: ocn465190110.

[31] Eivazi, H.; Tahani, M.; Schlatter, P.; Vinuesa, R. Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations. *Physics of Fluids* **2022**, *34*, 075117.

[32] Gao, H.; Sun, L.; Wang, J.-X. PhyGeoNet: Physics-Informed Geometry-Adaptive Convolutional Neural Networks for Solving Parameterized Steady-State PDEs on Irregular Domain. *Journal of Computational Physics* **2021**, *428*, 110079, arXiv:2004.13145 [physics].

[33] Tartakovsky, A. M.; Marrero, C. O.; Perdikaris, P.; Tartakovsky, G. D.; Barajas-Solano, D. Physics-Informed Deep Neural Networks for Learning Parameters and Constitutive Relationships in Subsurface Flow Problems. *Water Resources Research* **2020**, *56*, e2019WR026731.

[34] Berner, J.; Dablander, M.; Grohs, P. Numerically Solving Parametric Families of High-Dimensional Kolmogorov Partial Differential Equations via Deep Learning. 2020; http://arxiv.org/abs/2011.04602, arXiv:2011.04602 [cs, math, stat].

[35] Han, J.; Jentzen, A.; E, W. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences* **2018**, *115*, 8505–8510.

[36] Greenfeld, D.; Galun, M.; Basri, R.; Yavneh, I.; Kimmel, R. Learning to Optimize Multigrid PDE Solvers. Proceedings of the 36th International Conference on Machine Learning. 2019; pp 2415–2423.

[37] Bar-Sinai, Y.; Hoyer, S.; Hickey, J.; Brenner, M. P. Learning data driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences* **2019**, *116*, 15344–15349, arXiv:1808.04930 [cond-mat, physics:physics].

[38] Raissi, M.; Perdikaris, P.; Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* **2019**, *378*, 686–707.

[39] Kochkov, D.; Smith, J. A.; Alieva, A.; Wang, Q.; Brenner, M. P.; Hoyer, S. Machine learning accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences* **2021**, *118*, e2101784118, arXiv:2102.01010 [physics].

[40] Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G. E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence* **2021**, *3*, 218–229.

[41] Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Neural Operator: Graph Kernel Network for Partial Differential Equations. 2020; http://arxiv.org/abs/2003.03485, arXiv:2003.03485 [cs, math, stat].

[42] Khoo, Y.; Lu, J.; Ying, L. Solving parametric PDE problems with artificial neural networks. *European Journal of Applied Mathematics* **2021**, *32*, 421–435, arXiv:1707.03351 [math].

[43] Nelsen, N. H.; Stuart, A. M. The Random Feature Model for Input-Output Maps between Banach Spaces. *SIAM Journal on Scientific Computing* **2021**, *43*, A3212–A3243, arXiv:2005.10224 [physics, stat].

[44] Pathak, J.; Subramanian, S.; Harrington, P.; Raja, S.; Chattopadhyay, A.; Mardani, M.; Kurth, T.; Hall, D.; Li, Z.; Azizzadenesheli, K.; Hassanzadeh, P.; Kashinath, K.; Anandkumar, A. FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators. 2022; http://arxiv.org/abs/2202.11214, arXiv:2202.11214 [physics].

[45] Wen, G.; Li, Z.; Long, Q.; Azizzadenesheli, K.; Anandkumar, A.; Benson, S. M. Real-time high-resolution CO2 geological storage prediction using nested Fourier neural operators. *Energy & Environmental Science* **2023**, *16*, 1732–1741.

[46] Guan, S.; Hsu, K.-T.; Chitnis, P. V. Fourier Neural Operator Networks: A Fast and General Solver for the Photoacoustic Wave Equation. *Algorithms* **2023**, *16*, 124, arXiv:2108.09374 [cs, eess].

[47] Wen, G.; Li, Z.; Azizzadenesheli, K.; Anandkumar, A.; Benson, S. M. U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources* **2022**, *163*, 104180.

[48] Rosofsky, S. G.; Majed, H. A.; Huerta, E. A. Applications of physics informed neural operators. *Machine Learning: Science and Technology* **2023**, *4*, 025022, arXiv:2203.12634 [astro-ph, physics:gr-qc, physics:physics].

[49] Tesch, C. M.; Kompa, K.-L.; Vivie-Riedle, R. d. Design of optimal infrared femtosecond laser pulses for the overtone excitation in acetylene. *Chemical Physics* **2001**, *267*, 173–185.

[50] Carreira, L. A.; Mills, I. M.; Person, W. B. Two-Dimensional Anharmonic Oscillator. Application to 2,5-Dihydrofuran. *The Journal of Chemical Physics* **1972**, *56*, 1444–1448.

[51] Conte, R.; Parma, L.; Aieta, C.; Rognoni, A.; Ceotto, M. Improved semiclassical dynamics through adiabatic switching trajectory sampling. *The Journal of Chemical Physics* **2019**, *151*, 214107.

[52] Thomas, E. F.; Henriksen, N. E. Applying artificial neural networks to coherent control experiments: A theoretical proof of concept. *Physical Review A* **2019**, *99*, 023422.

[53] Christensen, L.; Nielsen, J. H.; Brandt, C. B.; Madsen, C. B.; Madsen, L. B.; Slater, C. S.; Lauer, A.; Brouard, M.; Johansson, M. P.; Shepperson, B.; Stapelfeldt, H. Dynamic Stark Control of Torsional Motion by a Pair of Laser Pulses. *Physical Review Letters* **2014**, *113*, 073005.

[54] Thomas, E. F.; Henriksen, N. E. Phase-Modulated Nonresonant Laser Pulses Can Selectively Convert Enantiomers in a Racemic Mixture. *The Journal of Physical Chemistry Letters* **2017**, *8*, 2212–2219.

[55] Madsen, C. B.; Madsen, L. B.; Viftrup, S. S.; Johansson, M. P.; Poulsen, T. B.; Holmegaard, L.; Kumarappan, V.; Jørgensen, K. A.; Stapelfeldt, H. A combined experimental and theoretical study on realizing and using laser controlled torsion of molecules. *The Journal of Chemical Physics* **2009**, *130*, 234310.

[56] Zhu, W.; Botina, J.; Rabitz, H. Rapidly convergent iteration methods for quantum optimal control of population. *The Journal of Chemical Physics* **1998**, *108*, 1953–1963.

[57] Kosloff, R.; Rice, S. A.; Gaspard, P.; Tersigni, S.; Tannor, D. J. Wavepacket dancing: Achieving chemical selectivity by shaping light pulses. *Chemical Physics* **1989**, *139*, 201–220.

[58] Peirce, A. P.; Dahleh, M. A.; Rabitz, H. Optimal control of quantum-mechanical systems: Existence, numerical approximation, and applications. *Physical Review A* **1988**, *37*, 4950–4964.

[59] Schröder, M.; Carreón-Macedo, J.-L.; Brown, A. Implementation of an iterative algorithm for optimal control of molecular dynamics into MCTDH. *Physical Chemistry Chemical Physics* **2008**, *10*, 850–856.

[60] Gordon, R. J.; Rice, S. A. ACTIVE CONTROL OF THE DYNAMICS OF ATOMS AND MOLECULES. *Annual Review of Physical Chemistry* **1997**, *48*, 601–641.

[61] Mockus, J. *Bayesian approach to global optimization: theory and applications*; Mathematics and its applications. Soviet series; Kluwer Academic: Dordrecht ; Boston, 1989.

[62] Cotter, S. L.; Roberts, G. O.; Stuart, A. M.; White, D. MCMC Methods for Functions: Modifying Old Algorithms to Make Them Faster. *Statistical Science* **2013**, *28*, 424–446.

[63] Tal-Ezer, H.; Kosloff, R. An accurate and efficient scheme for propagating the time dependent Schrödinger equation. *The Journal of Chemical Physics* **1984**, *81*, 3967–3971.

[64] Richings, G.; Polyak, I.; Spinlove, K.; Worth, G.; Burghardt, I.; Lasorne, B. Quantum dynamics simulations using Gaussian wavepackets: the vMCG method. *International Reviews in Physical Chemistry* **2015**, *34*, 269–308.

[65] Feit, M.; Fleck, J.; Steiger, A. Solution of the Schrödinger equation by a spectral method. *Journal of Computational Physics* **1982**, *47*, 412–433.

[66] Meyer, H.-D.; Manthe, U.; Cederbaum, L. The multi-configurational time-dependent Hartree approach. *Chemical Physics Letters* **1990**, *165*, 73–78.

[67] Light, J. C.; Carrington Jr., T. Discrete variable representations and their utilization. *acp* **2000**, *114*, 263–310.

[68] Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Fourier Neural Operator for Parametric Partial Differential Equations. 2021; http://arxiv.org/abs/2010.08895, arXiv:2010.08895 [cs, math].

[69] Chen, T.; Chen, H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks* **1995**, *6*, 911–917.

[70] O'Shea, K.; Nash, R. An Introduction to Convolutional Neural Networks. 2015; http://arxiv.org/abs/1511.08458, arXiv:1511.08458 [cs].

[71] Khaneja, N.; Reiss, T.; Kehlet, C.; Schulte-Herbrüggen, T.; Glaser, S. J. Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *Journal of Magnetic Resonance* **2005**, *172*, 296–305.

[72] Magann, A.; Chen, L.; Ho, T.-S.; Rabitz, H. Quantum optimal control of multiple weakly interacting molecular rotors in the time-dependent Hartree approximation. *The Journal of Chemical Physics* **2019**, *150*, 164303.

[73] Ramos Ramos, A. R.; Kühn, O. Direct Optimal Control Approach to Laser-Driven Quantum Particle Dynamics. *Frontiers in Physics* **2021**, *9*.

[74] Deng, Z.; Tutunnikov, I.; Averbukh, I. S.; Thachuk, M.; Krems, R. V. Bayesian optimization for inverse problems in time-dependent quantum dynamics. *The Journal of Chemical Physics* **2020**, *153*, 164111.

[75] Lazin, M. F.; Shelton, C. R.; Sandhofer, S. N.; Wong, B. M. High-dimensional multi-fidelity Bayesian optimization for quantum control. *Machine Learning: Science and Technology* **2023**, *4*, 045014.

[76] Brooks, S. Markov chain Monte Carlo method and its application. *Journal of the Royal Statistical Society: Series D (The Statistician)* **1998**, *47*, 69–100.

[77] Jensen, H. A.; Jerez, D. J.; Valdebenito, M. An adaptive scheme for reliability-based global design optimization: A Markov chain Monte Carlo approach. *Mechanical Systems and Signal Processing* **2020**, *143*, 106836.

[78] Robert, C. P. The Metropolis-Hastings algorithm. 2016; http://arxiv.org/abs/1504.01896, arXiv:1504.01896 [stat].

[79] Parzen, E. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics* **1962**, *33*, 1065–1076.

[80] Son, H.; Jang, J. W.; Han, W. J.; Hwang, H. J. Sobolev training for physics-informed neural networks. *Communications in Mathematical Sciences* **2023**, *21*, 1679–1705.

[81] Czarnecki, W. M.; Osindero, S.; Jaderberg, M.; Świrszcz, G.; Pascanu, R. Sobolev Training for Neural Networks. 2017; http://arxiv.org/abs/1706.04859, arXiv:1706.04859 [cs].

[82] Bennett, L.; Melchers, B.; Proppe, B. *Curta: A General-purpose High-Performance Computer at ZEDAT, Freie Universität Berlin*; Freie Universität Berlin, Germany, 2020.

[83] Hagemeier, B. HDF Cloud – Helmholtz Data Federation Cloud Resources at the Jülich Supercomputing Centre. *Journal of large-scale research facilities JLSRF* **2019**, *5*, A137.