SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Finding Conical Intersections with Tensor Network Methods

Johannes Spies

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Finding Conical Intersections with Tensor Network Methods

Finden konischer Schnittpunkte mit Methoden der Tensornetzwerke

Author:	Johannes Spies
Supervisor:	Prof. Dr. Christian Mendl
Advisors:	M.Sc. Shuo Sun
Submission Date:	15.01.2025

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 15.01.2025

Johannes Spies

Acknowledgments

First, I would like to express my gratitude to Shuo Sun for her support and guidance throughout this thesis. Her quick responses and patience in addresssing my questions have been an a constant source of encouragement. I am grateful for her dedciation and valuable feedback. Second, I would like to thank Christian Mendl for insightful discussions and helping me to shape my ideas. Our interactions have been invaluable in broadening my perspective and refining the focus of my research. Lastly, I want to thank my family and loved ones for their belief in me and standing by me at all times.

Abstract

The potential energy surface (PES) of a quantum chemical system is notoriously hard to compute. In the past decades, tensor networks emerged as a promising tool to study many-body quantum systems. The framework of tensor networks was recently found to provide astonishing results on specific kinds of mathematical functions while being very sample-efficient using the so-called quantics tensor cross interpolation (QTCI) method. This work investigates the question of whether the QTCI method can also be applied to approximating PESs using a parsimonious amount of samples. For that, we apply QTCI on a variety of different quantum systems and compare the sample efficiency and accuracy. We show that while leading to imporessive results on length-separating functions, the PES just might not be the application where QTCIs have the edge over traditional machine learning methods.

Kurzfassung

Die Potential Energy Surfaces (PES) von quantenchemischen Systemen sind aufwendig zu berechnen. In den letzen Jahrzehnen haben sich Tensornetzwerke als vielversprechendes Werkzeug entwickelt um Mehrkörperquantensysteme zu beschreiben und darzustellen. Seit kurzem ist der Werkzeugkasten um Tensornetzwerke erweitert worden. Sie können auch verwendet werden um mathematische Funktionen platzsparend darzustellen; mit Hilfe der Methode der Quantics Tensor Cross Interpolation (QTCI). Die vorliegende Arbeit untersucht die Frage, ob die QTCI Methode auch auf die PES von Quantensystemen angewandt werden kann und dabei nur eine kleine Anzahl von Datenpunkten benötigt. Wir testen unseren Ansatz mit einer Reihe von kleinen Molekülen. Obgleich die QTCI Methode großartige Ergebnisse für einen bestimmten Typ von mathematischen Funktionen liefert, finden wir in dieser Arbeit heraus, dass QTCI sich in unserer Betrachtung nicht besser verhält als traditionelle Methoden des maschinellen Lernens.

Contents

1 Introduction

In the vast expanse of chemistry, photochemistry is a part of the field investigating the interaction between light and matter. Among its numerous applications, photochemistry has been instrumental in understanding fundamental processes within physical and biological systems. In this thesis, we aim to expand the toolkit of quantum chemistry, leveraging various computational tools to enhance the ability to study systems capable of being controlled by the influence of light.

First, we want to give a motivation for what systems exhibit a behavior sensitive to light. For this introduction, we focus on the rhodopsin molecule and its role in the process of photoreactions. Rhodopsin is a light-sensitive protein found in the retina of the eye. This protein plays a crucial role in perceiving light of a certain wavelength. In its normal state, the protein has a specific arrangement of its atoms called a *conformation*. Upon absorption of a photon, rhodopsin undergoes a photoisomerization reaction, resulting in a change in its conformation. This transformation triggers a cascade of events leading to the transduction of light into electrical signals within the visual system. Ultimately, rhodopsin is the first step in the chain that allows us to see. This behavior is present in other proteins or molecules as well. For instance, photosynthesis also leverages the interaction of light with respective proteins. In our investigation, we conducted experiments on *formaldimine*, a small molecule with similar energetic properties to rhodopsin. When light is incident on this molecule, its conformation changes, just like rhodopsin. Therefore, it serves as a simple yet expressive model system to study photochemical reactions.

In chemistry, the field of ab initio methods deals with the task of modeling and simulating chemical systems from first principles. Since the beginning of the last century, we have understood matter at a degree fine enough to mathematically model systems at a high enough accuracy such that chemical systems can be modeled and understood without actual wet-lab experiments. These advances gave rise to a whole new field of computational chemistry, in which a rich toolbox for studying chemical systems using computational tools has been developed over the years. Nevertheless, the problem of modeling photochemical reactions has remained challenging because of the breakdown of common approximations when studying photosensitive compounds.

In this thesis, we build on existing tools developed for static quantum chemistry systems and try to extend the knowledge to rich reaction landscapes. We do this by using a method that has proven itself excellent at representing high-dimensional objects while being memory efficient. In the following, we will first discuss the necessary background to understand the methods that we used. We start by introducing both quantum chemical foundations and the techniques we used to extrapolate from single configurations. In the next part, we explain how we combined the methods introduced to form a pipeline capable of modeling the energy landscape of molecules. After that, we present the findings and observations that we made during our experiments.

2 Background

As alluded to in the introduction, ab initio quantum chemistry allows the understanding of the behavior of physical systems at the atomic scale. In this context, quantum chemistry focuses on applying the principles of quantum mechanics to describe the properties and reactions of molecules and other chemical systems.

In the following part, we will provide a detailed account of the quantum mechanical foundations and computational methods used in this thesis. After that, we introduce tensor network methods and present tensor network-based techniques used to analyze quantum chemistry systems. We only include topics relevant to the problem at hand. This background knowledge is essential for understanding the methodology and results presented in the subsequent chapters of this thesis.

2.1 Quantum Foundations

The experiments conducted during the project require a representation of the chemical problems of the highest accuracy. For this, one usually resorts to quantum chemistry, which possesses a mathematically rigorous way of describing the studied systems. The next two sections will give a short overview of the methods required to analyze systems with the required accuracy. Further information and a deeper treatment can be found in computational chemistry textbooks like [12].

2.1.1 Notation and Formalism

In the framework of quantum chemistry, everything can be formulated using means familiar from algebraic studies. The core of the mathematical view of quantum chemistry is the observation that everything can be expressed in terms of operators, which will be marked by a hat $\hat{\cdot}$, and vectors, which will be written as $|\cdot\rangle$. For instance, $\hat{A}$ will denote an operator named A, and $|b\rangle$ will denote a vector named b. Let V denote the vector space containing all vectors. This means we can write $|a\rangle \in V$ for any vector a that is an element of the vector space. Note that the vectors support all operations to be able to call V a vector space. Two of these operations are addition and scalar multiplication. The addition means that for any $|a\rangle, |b\rangle \in V$, the addition is defined, and the result is an element of V. More precisely, $|a\rangle + |b\rangle \in V$. Scalar multiplication can be defined similarly, for any scalar $a \in \mathbb{C}$ and $|b\rangle \in V$, the statement $a\,|b\rangle \in V$ holds true.

A central relationship of the two mathematical objects introduced is that *operators can act on vectors, producing new vectors.* As an example, one could give the result $\hat{B}\,|\alpha\rangle$ of acting with an operator $\hat{B}$ on a vector $|\alpha\rangle$ a new name, $|\beta\rangle$. One would write this as $|\beta\rangle = \hat{B}\,|\alpha\rangle$. Furthermore, every $|a\rangle$ induces a corresponding linear form denoted by $\langle a| : V \to \mathbb{C}$. The application of such a linear form is typically shortened to $\langle a|\,(|b\rangle) = \langle a|b\rangle$. Since the result of acting on a vector $|c\rangle$ with an operator $\hat{B}$ results in another vector $\hat{B}\,|c\rangle$, the application of a linear form $\langle a|$ on said vector $\langle a|\,\left(\hat{B}\,|c\rangle\right)$ is often shortened as $\langle a|\hat{B}|c\rangle$.

The previous section introduced a very abstract way of dealing with vectors and operators. In quantum chemistry, there are two kinds of vector spaces that are of particular importance. The first one is the one from linear algebra, consisting of n-dimensional vectors. In this notation, operators are represented by matrices that can be multiplied by vectors to yield new vectors. The second kind is functional vector spaces, consisting of mathematical functions. One can then define operators on these as a function that creates another function. An operator of particular importance is the Laplacian operator $\nabla^2 : (\mathbb{R}^n \to \mathbb{R}) \to (\mathbb{R}^n \to \mathbb{R})$, which maps a function to another function that returns the sum of the original function's second derivatives

$$\nabla^2 : f \mapsto \left[\boldsymbol{x} \mapsto \frac{\partial^2 f}{\partial x_1^2}(x_1) + \frac{\partial^2 f}{\partial x_2^2}(x_2) + \cdots + \frac{\partial^2 f}{\partial x_n^2}(x_n) \right].$$

In both the algebraic world and the function vector spaces, the linear form mapping a vector space element to a complex number is defined. The former uses the well-known inner product. For instance, let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ be two vectors with elements a_i and b_i. Then, $\langle \boldsymbol{a}|\boldsymbol{b}\rangle \overset{\text{def}}{=} \boldsymbol{a}^\top \boldsymbol{b} = \sum_{i=1}^{n} a_i \cdot b_i$. For functions, this application is defined via integration. The following shows the definitions for applying a linear form directly and for first applying an operator to a vector:

$$\begin{aligned} \langle \psi|\phi\rangle &= \int \psi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ \langle \psi|\hat{A}|\phi\rangle &= \int \psi(\boldsymbol{x}) \cdot \left[\hat{A}\phi\right](\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{2.1}$$

These definitions are crucial for understanding the methods used in quantum chemistry.

2.1.2 Atoms and Molecules

The real world has a direct translation to the mathematical framework introduced in the previous sections. The fundamental description of quantum chemical systems is as follows. The mechanics of a system specifying the interactions between its constituents, like atoms or electrons, are captured in a single operator $\hat{H}$ called Hamiltonian. The dynamic components of a system, or more exactly, its state, are captured in a single vector, typically named ψ. The relationship between the system definition and the state can be packed into a single equation, the time-independent Schrödinger equation $\hat{H}\left|\psi\right\rangle = E\left|\psi\right\rangle$. In linear algebra parlance, this is a common form called *eigenvalue problem.* In the case of quantum chemistry, one is usually tasked with finding at least one solution, i.e. some $\left|\psi^\star\right\rangle$, to that problem. The collection of these solutions is comprised of so-called called *eigenpairs,* each of them represents a solution to the eigenvalue problem. Each eigenpair $(\left|\psi_i\right\rangle, E_i)$ contains an *eigenvector* $\left|\psi_i\right\rangle$ and a corresponding *eigenvalue* E_i.

As alluded to before, the choice of the Hamiltonian $\hat{H}$ depends on the systems at hand and its molecular configuration. The foundations presented here are taken from [21]. In the view of a quantum chemist, a molecule is described by a set of N nuclear positions $\{R_I \in \mathbb{R}^3\}_{I=1}^N$ with corresponding electronic charges $\{Z_I\}_{I=1}^N$ and mass-ratios $\{M_I\}_{I=1}^N$, and n electron positions $\{\boldsymbol{r}_i \in \mathbb{R}^3\}_{i=1}^n$. The Hamiltonian for the corresponding systems is then expressed in terms of these properties:

$$\hat{H} = \underbrace{-\frac{1}{2}\sum_{i=1}^{n}\nabla_i^2 - \frac{1}{2}\sum_{I=1}^{N}\frac{1}{M_I}\nabla_I^2}_{\text{kinetic operators}} \underbrace{-\sum_{i=1}^{n}\sum_{I=1}^{N}\frac{Z_I}{r_{iI}}}_{\substack{\text{Coulomb}\\ \text{attraction}}} \underbrace{+\sum_{i=1}^{n}\sum_{j>i}^{n}\frac{1}{r_{ij}} + \sum_{I=1}^{N}\sum_{J>I}^{N}\frac{Z_I Z_J}{R_{IJ}}}_{\text{repulsion terms}},$$

where ∇_α^2 denotes the Laplacian operator that only acts on a set of coordinates indexed by α. Note that this description is in so-called *atomic units,* which do not include multiplicative prefactors that are part of standardized units.

The solution of the Schrödinger equation with this Hamiltonian is hard to obtain. Therefore, one can leverage the so-called *Born-Oppenheimer approximation* and reduce the computational cost. Because nuclei are much heavier than electrons, it is common to neglect the movement of the nuclei. They are treated as static parameters of the Hamiltonian, and the solution of the Schrödinger equation only describes the movement of the electrons. It is common to solve the Schrödinger equation using functions. The Schrödinger equation can thus be interpreted as a differential equation. A solution to the Schrödinger equation is then called *wave function.* Unfortunately, this is not possible analytically for any but the smallest systems.

2.2 Potential Energy Surfaces

As described before, solving the Schrödinger equation for a single set of nuclear coordinates may give rise to a set of solutions that correspond to different electronic states a quantum system can be in. Furthermore, the Born-Oppenheimer approximation essentially converts the Hamiltonian operator to a function of the nuclear coordinates $\hat{H}\left(\{R_I\}_{I=1}^{N}\right)$. This results in a continuous surface of solutions. Each set of nuclear coordinates has an energy value for a specific electronic state. These ingredients, the different energy levels, and the nuclear coordinates, create a set of surfaces, the so-called *potential energy surfaces (PES)*. Obtaining a PES analytically is not possible because solving the Schrödinger equation for a specific set of coordinates is already impossible for many systems of relevant size. Therefore, one has to resort to approximations. It is common to obtain solutions to the Schrödinger equation in some locations, and have a cheap approximation to fill in the gap.

2.2.1 Conical Intersections

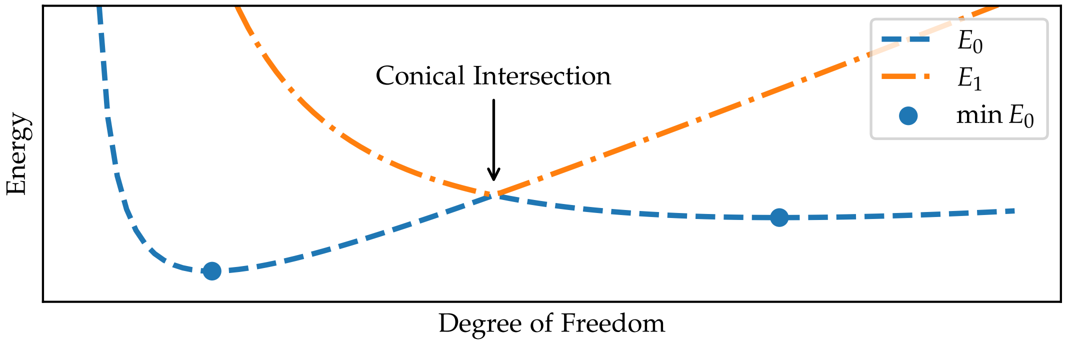


Figure 2.1: A plot illustrating the behavior of two PES around a conical intersection. The chart plots the energy as two functions of an arbitrary degree of freedom in a chemical system. The one-dimensional degree of freedom here can be imagined as a chemical bond length or a rotation angle. The lower curve labeled E_0 corresponds to the electronic ground state. E_1 labels the first excited state respectively. The point where the two curves touch is referred to as a conical intersection. Furthermore, the energetically most favorable states are marked as dots on the ground state surface. Note that the functions shown here do not correspond to a particular real molecule.

PESs possess many important properties with respect to chemistry. For key importance of this thesis are *conical intersections* of the PES. These points occur at degeneracies of the solutions of the time-independent Schrödinger equation. Degenerate means that two solutions of the Schrödinger equation have the same energy eigenvalue. For complex molecules, these degeneracies usually occur for unique points on the energy surfaces of molecules. A qualitative example of such a system is shown in Figure 2.1.

The chart shows two hypothetical one-dimensional surfaces that touch at a marked point of degeneracy.

Conical intersections are of relevance for chemistry because they allow transitions from an electronically higher state into an electronically lower-lying state that would not be reachable without the excitation. A molecule always seeks to minimize its internal energy because it is in the most favorable state. A radiationless transition from one electronic state into another is possible at a conical intersection. In the following, we give an example of how such a transition could take place.

2.2.2 Photo-Induced Reactions

Assume that, for instance, a molecule is in a minimum of the ground state corresponding to the left dot on E_0 in Figure 2.1. After a photonic excitation, the molecule can be put into an electrically excited state, corresponding to E_1. One option for the molecule is to reemit a photon and thus fall back into the ground state. Another option is described as follows: Since the conformation is not energetically favorable, the molecule relaxes into another state by traveling downwards along the energy surface of E_1. When reaching the point of conical intersection, it can transition from the first excited state into the depicted ground state without emitting a photon. The molecule relaxes further until it reaches the second local minimum, indicated by the dot on the right of the ground state surface. This reaction is what lies at the heart of many photochemical processes.

2.3 Interpolations and Approximations

One important distinction is of crucial importance for this work. Concretely, we want to point out the difference between approximations and interpolations. Both deal with the representation of a function by another function, which may be called *model*. This model may have favorable properties, such as a more parsimonious memory representation or requiring fewer computations, leading to better tractability. A function $f : \mathbb{R}^n \to \mathbb{R}$ can be *approximated* by another function $\tilde{f} : \mathbb{R}^n \to \mathbb{R}$. This approximation $\tilde{f}$ should be close to f within a certain margin of error.

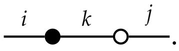
An *interpolation* extends the closeness requirement of an approximation to the original f. It is based on data points that are assumed to be samples taken from f. The interpolation provides a function $\hat{f}$ that provides a new set of values between the known data points by choosing $\hat{f}$ such that it passes through these points. The most common techniques include linear and polynomial interpolation, where a straight line or a polynomial is fitted to the given data points. In this thesis, we utilize a different kind of interpolation that will be explained in Section 2.6.

For our purposes, it is important to distinguish between approximations and interpolations. An interpolation is defined to be exact at the known data points. For an approximation, only the margin of error constraint has to be obeyed.

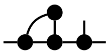
2.4 Tensor networks

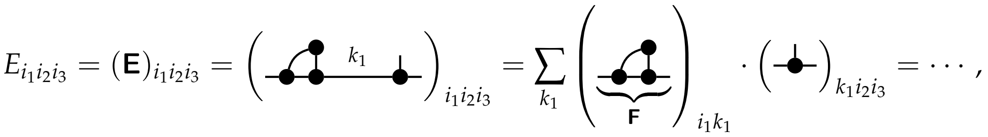
When studying tensor networks, it is common to print tensors and tensor networks in the Penrose convention [15]. For that, we would like to make use of the graphical notation. In the following, we will give examples of mathematical drawings and notations. Let $a =$ ● $\in \mathbb{R}$ be a scalar. One says that a has zero dimensions because it's a scalar. In the graphical notation, adding dimensions corresponds to *adding a leg* to the object. Let $\boldsymbol{b} =$ ●— $\in \mathbb{R}^{\ell}$ be a ℓ-dimensional vector containing real numbers. According to the same logic, $\boldsymbol{C} =$ —●— $\in \mathbb{R}^{m \times n}$ represents a matrix with $m \cdot n$ many entries. Adding one more dimension yields a tensor $\mathsf{D} =$ —┴●— $\in \mathbb{R}^{\cdot' \times \cdot'' \times \cdot'''}$ with three legs. Therefore, D is said to be *of order three.* This can be continued for tensors with arbitrary many dimensions, which correspond to dots with equivalently many legs. The number of legs or dimensions is called *order of a tensor.*

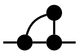
In tensor networks, matrix multiplication is generalized to a scalar product over an arbitrary dimension. For instance, the multiplication of two matrices $\boldsymbol{A} \in \mathbb{R}^{\cdot' \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times \cdot''}$ can be represented in the following way in the graphical notation:

$\boldsymbol{A} =$ —●—, $\boldsymbol{B} =$ —○—, $\boldsymbol{A} \cdot \boldsymbol{B} =$ .

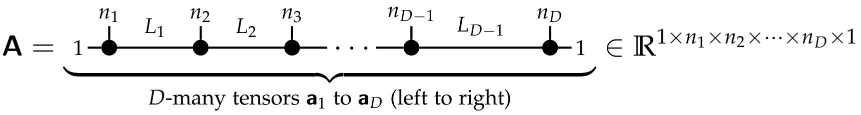
The connecting leg can be considered summing over the corresponding dimensions. By convention for matrix multiplication, this is the second dimension of the first input matrix $\boldsymbol{A}$ and the first dimension of the second input matrix $\boldsymbol{B}$. This can be written as the sum over the shared dimension $(\boldsymbol{A} \cdot \boldsymbol{B})_{ij} = \sum_{k=1}^{n} A_{ik} \cdot B_{kj}$. The shared dimension is indexed by k. In the context of tensor networks, it is very common to generalize the matrix product to tensors of higher order than two. This can be done by summing over different dimensions of the input tensors. Furthermore, a summation or so-called *contraction* can include more than two tensors, unlike a standard matrix product. When visualizing such a contraction, one quickly arrives at a so-called *tensor network.* It is nothing but a graphical depiction of the underlying contraction.

In the following, the concept of a contraction will be illustrated with an example. Let $\mathsf{E} =$  be a tensor network composed of four tensors, three of order three and one of order two. The network has three *open legs,* which are legs without a connection inside the network. The whole network E represents a new tensor of order three, which is governed by its three open legs. By recursively contracting **E** like in the following sum



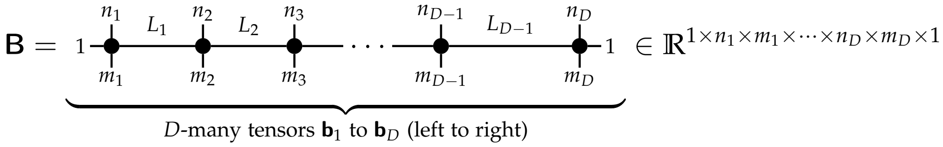
one can obtain a tensor's entries, here at the coordinates i_1, i_2, and i_3. The intermediate tensor  has been given a new name, **F**. If one fully extracts the entry $E_{i_1 i_2 i_3}$, it would be necessary to break down **F** by applying more contractions summing over the subtensor's legs. Only once all tensors are part of the sum in their pure form, without any connected tensors, can the sum be evaluated.

A common mathematical object is called *matrix product state*, short MPS. It is a special way of expressing a tensor with D dimensions, as shown in the following diagram.


$\mathbf{A} =$
1
n_1
L_1
n_2
L_2
n_3
$\cdots$
n_{D-1}
L_{D-1}
n_D
1
$\in \mathbb{R}^{1 \times n_1 \times n_2 \times \cdots \times n_D \times 1}$
D-many tensors $\mathbf{a}_1$ to $\mathbf{a}_D$ (left to right)

Like before, each open leg corresponds to one dimension in the represented tensor **A**. The so-called *bond dimensions* $\{L_i\}_{i=1}^{D-1}$ specify the number of elements in the dimension joining two tensors. Note that the left-most tensor and the right-most tensor have one leg forced to one. It is common to set $L_0 = L_D = 1$. Each element of the tensor **A** is a three-tensor, and we can interpret the MPS as a set comprised of these tensors $\mathbf{A} \equiv \left\{ \mathbf{a}_i \in \mathbb{R}^{L_{i-1} \times n_i \times L_i} \right\}_{i=1}^{D}$.

One close relative to the MPS that will be important in the current discussion is the *matrix product operator*, short MPO, representation of tensors. Similar to the MPS representation, it consists of a chain of tensors. The important distinction is that MPOs have one more set of legs. The following diagram shows an MPO in its general form.


$\mathbf{B} =$
1
n_1
L_1
n_2
L_2
n_3
$\cdots$
n_{D-1}
L_{D-1}
n_D
1
$\in \mathbb{R}^{1 \times n_1 \times m_1 \times \cdots \times n_D \times m_D \times 1}$
m_1
m_2
m_3
m_{D-1}
m_D
D-many tensors $\mathbf{b}_1$ to $\mathbf{b}_D$ (left to right)

A couple of years ago, tensor networks qualified themselves as powerful tools to represent and solve problems in electronic structure theory. The usage of MPS and MPO representations will be touched on in the following part about quantum chemistry.

2.5 Ab Initio Quantum Chemistry

A big part of the achievements of quantum chemistry is solving the Schrödinger equation and accurately representing the many-body quantum state. As alluded to before, one seeks to find a solution to the Hamiltonian operator $\hat{H}$ that represents the quantum system in the Born-Oppenheimer approximation as

$$\hat{H} = -\frac{1}{2}\sum_{i=1}^{n} \nabla_i^2 + \sum_{i=1}^{n} V(\boldsymbol{r}_i) + \sum_{i=1}^{n}\sum_{j>i}^{n} \frac{1}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2}, \tag{2.2}$$

where $V(\boldsymbol{r}_i)$ denotes the potential each electron experiences from being influenced by the nuclei. Note that the kinetic movement of the nuclei and the influence of the electrons on the nuclei have been removed. This means that the Hamiltonian is parametrized by the positions and charges of the nuclei $\{(\boldsymbol{R}_i, Z_i)\}_{i=1}^{N} \subseteq \left(\mathbb{R}^3 \times \mathbb{N}\right)$. The solutions of the corresponding Schrödinger equation are functions of the electron positions $\psi\left(\{\boldsymbol{r}_i\}_{i=1}^{n}\right)$ and are referred to as *wave functions*. These can be found using various quantum chemistry methods. In the following, we will briefly explain the main ideas behind the methods we used in our experiments for finding electronic ground-state solutions. Then, we will explain the extensions for finding excited states.

2.5.1 Basis Sets

One important concept of quantum chemistry is the use of *basis sets.* In general, it is a tool frequently used to simplify the use of quantum mechanical operators because it removes the computation of complicated integrals from the optimization stage. A basis set refers to a finite set of normalized functions that can be used as a starting point to represent the wave function. A clever choice of functions makes it possible to precompute the values of important operators and use the matrix representation in further computations. In quantum chemistry, a huge variety of basis sets is available to choose from. A popular choice among them is Gaussian-type functions because they allow analytic solutions to the integrals arising from operators composing the Hamiltonian.

An operator can generally be expressed as a matrix if one restricts the set of admissible functions to a finite set of m functions $\{\psi_i\}_{i=1}^{m}$. Then, one can reformulate an operator $\hat{A}$ as a matrix $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ via the relationship

$$\boldsymbol{A}_{ij} = \left\langle \psi_i \middle| \hat{A} \middle| \psi_j \right\rangle \in \mathbb{C}.$$

The so-called *overlap matrix* $\boldsymbol{S}$ is an important matrix induced by the identity operator. Its element s are the values $\boldsymbol{S}_{ij} = \left\langle \psi_i \middle| \psi_j \right\rangle$ that measure the overlap of any two functions in the corresponding basis set.

Using pre-specified function basis sets allows defining arbitrary vectors as a linear combination of basis functions like $|\psi_\sigma\rangle = \sigma_1 |\psi_1\rangle + \sigma_2 |\psi_2\rangle + \cdots + \sigma_m |\psi_m\rangle$. Here, $|\psi_\sigma\rangle$ is parametrized by a coefficient vector $\sigma \in \mathbb{C}^m$. It is very common to express atomic orbitals as linear combinations of basis functions. The coefficient vector parametrizes the shape of the corresponding orbital. Finding such a coefficient vector is a crucial task in quantum chemistry, which will be explained in more detail in the next section. In general, a basis set with more functions can be used to tackle more complex problems.

2.5.2 Self-Consistent Field Methods

This section explains the rationale behind the Hartree-Fock and Density Functional Theory based on [8]. The last term in the Born-Oppenheimer approximation's Hamiltonian makes the solution of the Schrödiner equation dependent on the solution itself. This is because the electron's motion is guided not only by the system but also by the potential they represent themselves. One can simplify the problem of solving the full Hamiltonian by considering an alternative version of the Hamiltonian as

$$\hat{H}^{(0)} = -\frac{1}{2}\sum_{i=1}^{n} \nabla_i^2 + \sum_{i=1}^{n} V(\boldsymbol{r}_i) + \sum_{i=1}^{n} v(\boldsymbol{r}_i). \tag{2.3}$$

Note that the only change from Equation 2.2 is that the pairwise interaction sum is reduced to a general potential that summarizes the contributions of all electrons in the system. Therefore, solving the corresponding Schrödinger equation will yield a solution close to the ground-state, which is not exact because the intricate electron-electron interactions are treated in a simplified form. The methods solving this problem are referred to as *self-consistent field (SCF)* methods.

It is common to decompose the total energy of a quantum chemical system into the five terms

$$E = E_\mathrm{T} + E_\mathrm{V} + E_\mathrm{J} + \underbrace{E_\mathrm{X} + E_\mathrm{C}}_{\overset{\mathrm{def}}{=} E_\mathrm{XC}}.$$

Over the years, many methods have been developed to accurately find a value for each of the terms from the wave function of a molecule. The inverse problem of finding a wave function for the whole energy expression from the individual terms is not as easy and has not been solved yet.

The Hartree-Fock method solves an eigenvalue equation similar to the molecular Schrödinger equation. Contrary to an analytical solution, the Hartree-Fock method converts finding the eigenfunction into an iterative minimization procedure. Furthermore, the Hartree-Fock solution uses a simplified version of the Hamiltonian. The adjusted

Schrödinger equation $\hat{F} |\psi\rangle = E |\psi\rangle$ uses the so-called *Fock operator*

$$\hat{F} = \hat{T} + \hat{V} + \hat{J} + \hat{K}.$$

As before, the problem is expressed as an operator, here $\hat{F}$, that is a sum of other operators. The first two operators, $\hat{T}$ and $\hat{K}$, are the electron's kinetic energy and the potential energy induced by the static nuclei. The operator $\hat{J}$ is comprised of the sum of the Coulomb potentials of the electrons. Thus, each electron is assumed to move in the field induced by the sum of the Coulomb potential. The $\hat{K}$ operator is the so-called *exchange operator.*

The orbital wave functions can be expanded in the form of a basis set using the technique described in Section 2.5.1. Each molecular orbital ϕ_i of the Slater determinant is expanded as a linear combination $|\phi_j\rangle = \sum_{i=1} C_{ij} |\chi_i\rangle$ of atomic orbitals χ_i. For these atomic orbitals, basis functions as discussed in Section 2.5.1 are employed. Because of the sum, one can evaluate the integral induced by each element of the Fock operator with each atomic orbital separately. Similar to the techniques discussesd in Section 2.5.1, the Fock operator can be expressed as a matrix $\boldsymbol{F}$. This reduces the problem to finding the solutions in the form of the coefficient matrix $\boldsymbol{C}$ of

$$\boldsymbol{F}\boldsymbol{C} = \boldsymbol{S}\boldsymbol{C}\varepsilon. \tag{2.4}$$

In this equation, $\boldsymbol{S}$ is the overlap matrix mentioned in Section 2.5.1, and ε is a vector collecting all energy eigenvalues of each respective molecular orbital. This changes the problem of finding the coefficients to a form well-known from linear algebra. The concrete derivation is omitted here, but [12] presents an explanation of how to derive the so-called *Roothaan-Hall equations* shown in Equation 2.4 from the full Lagrangian of the studied system.

The goal of Hartree-Fock solvers is to find a set of coefficients corresponding to the quantum state with the minimum energy, called *ground state* of a system. This can only be done in an iterative fashion because a direct solution of Equation 2.4 is impossible. For the scope of this thesis, it is important to note that this minimum energy depends on the arrangement and nature of the nuclei. Therefore, the SCF procedure and other quantum chemistry tools can be seen as a black box. The nuclear configuration is passed in as input, and the result comes in the form of coefficients corresponding to an energy value. This value is precisely the energy on the PES for that specific conformation.

Over the past years, programs like `pyscf` [20] have been developed to provide access to quantum chemistry tools, like the Hartree-Fock method or density functional theory (DFT). Important for this thesis is to understand that the methods use an iterative approach to minimize the energy corresponding to a set of parameters. In this iterative scheme, the Fock matrix is formed for each iterative because it depends on the current set of wave function coefficients. In this inner step, the Fock matrix needs to be diagonalized to obtain a new set of coefficients. After this step, the algorithm decides if the iteration has converged and either continues the optimization with the new set of coefficients or returns them as the result of the algorithm.

2.5.3 Tensor Networks for Electronic Structure Problems

Quantum systems usually suffer from a computational blowup with respect to increasing system size. The tensor network techniques described in Section 2.4 can be used to represent quantum systems in a more sparse fashion by reducing the exponential overhead through the sacrifice of accuracy. In this setting, the quantum state, expressed through a set of basis set coefficients, can be represented by an MPS. The Hamiltonian operator is, similar to the matrix form, represented by an MPO. Schollwöck [19] introduces the DMRG algorithm, a method to iteratively find an optimal MPS representation for the quantum state.

It is important to know, that the inner diagonalization step of the SCF procedure can take various forms. Since the rise of tensor networks, it is common to represent the wave function coefficients using an MPS and the corresponding operator using an MPO. This allows for an approximate but configurable and fast diagonalization using the DMRG algorithm. In this work, we use this combination of the DMRG algorithm as an inner loop for the Hartree-Fock, or SCF, method. In the following, we refer to this combination as *DMRG-SCF*.

2.5.4 Time-Dependent Methods

The Hartree-Fock method and DFT can serve as a starting point for the computation of excited states. The excited state energies can be computed by considering the time-dependent (TD) Schrödinger equation instead of the time-independent version. For a detailed derivation of both the TD-HF and the TD-DFT equations, see [5]. The main takeaway for the scope of this thesis is, that both time-dependent methods can compute the energy of an excited state above the ground state for any system configuration.

2.6 Quantics Tensor Cross Interpolation

A recent approach leveraging tensor networks is called *quantics tensor cross interpolation (QTCI)*. It is a method to obtain a tensor network in MPS representation for a mathematical function. [17] introduces the QTCI technique and [7] reviews the state-of-the-art techniques in the ecosystem of TCIs. Here, we give a brief intuition of the method and the algorithmic version used in this thesis.

2.6.1 Approximating Functions with Scale-Separation

The main purpose of QTCI is to build an approximation for a special kind of function. These have the property that they both exhibit inherently local behavior and behavior that only affects the function globally. For instance, a function can have rapid oscillations locally but exhibit a fall-off behavior when zooming out. With traditional interpolation techniques, this is very hard to capture because of averaging effects: Either the local behavior is retained but not the global, and vice versa. QTCI allows for capturing both effects using a feature called *scale separation,* which can be solved using a so-called *quantics grid.*

2.6.2 Quantics Grid

Any number $x \in [a, b)$ can be decomposed into a so-called *quantics representation.* This is equivalent to assigning each number in the interval a corresponding bit string of B binary digits $(\sigma_1, \ldots, \sigma_B)$ such that

$$x \approx u = a + (b - a) \cdot \sum_{b=1}^{B} \frac{\sigma_b}{2^b}, \ \sigma_i \in \{0, 1\} \tag{2.5}$$

This is done by dividing $[a, b)$ into 2^B many equally spaced subintervals. u is the closest approximation to the number x possible at the given bit depth B. The resulting representation has the advantage of being finite and adjustable. One can freely choose B to arrive at a sufficient precision. Note that the bit strings cannot uniquely be assigned to a number because the representation is finite for a given but fixed B.

The quantization procedure can be extended to n dimensional vectors. Each component of a vector $\boldsymbol{x}$ can be quantized separately, yielding a matrix of bits, where each column corresponds to a bit string representing one dimension of the vector. If we use the notation from Equation 2.5, we can rewrite each component of a vector $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\top$ as

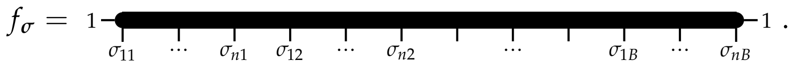
$$x_i \approx u_i = a + (b - a) \cdot \sum_{b=1}^{B} \frac{\sigma_{ib}}{2^b}, \ \sigma_{ib} \in \{0, 1\}, \ i \in \{1, \ldots, n\}.$$

The resulting matrix $\boldsymbol{\sigma}$ contains all bits that fully define the quantized vector.

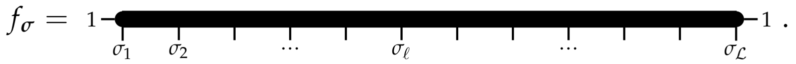
For this project, the number of bits per dimension B is a crucial hyperparameter. The system geometry dictates the number n and cannot be tuned.

2.6.3 MPS Representation of a Function

The goal of QTCI is to obtain an arbitrary compact representation of a n-dimensional function f by means of interpolation. This is done using the quantization tools introduced in the previous section and representing f by a tensor instead. Each function input is first quantized. This quantization converts the continuous input to a finite representation in bits. The legs of the tensor correspond to the bits and have two dimensions. Diagrammatically, f can be represented by a tensor



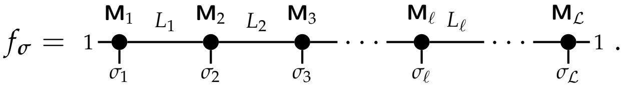
Note that the indices of the tensor can also be rearranged, as long as all $\mathcal{L} = nB$ indices are kept. This corresponds to the following diagrammatic representation



(2.6)

Grouping indices together that correspond to the same length scale is referred to as *fusing*. Instead of having two-dimensional legs for each bit, each leg corresponds to a specific length scale of dimension 2^n. This reduces the number of legs on the overall tensor f_σ.

As described in Section 2.4, this tensor can be represented by a smaller tensor network in MPS form if f_σ does not have full rank



(2.7)

The higher the rank of the tensor f_σ, the higher the required bond dimension to accurately represent the full tensor. The bond dimensions grow exponentially if the represented tensor is of full rank. Due to the construction on a quantics grid, this should not be the case for the class of functions exhibiting length-separation characteristics. Instead, these functions should yield a tensor that requires a modest bond dimension when represented by an MPS on a quantics grid. This is because of a low entanglement between different length scales, which leads to a small connecting bond dimension.

2.6.4 Matrix and Tensor Decompositions

In the previous section, a parsimonious representation of functions was described. The next important part of the discussion is to understand how the TCI algorithm can be used to construct such a representation. [17] gives a brief intuition of the method introduced in [14]. An in-depth discussion of the extended algorithmic foundations is given in [7]. In this part here, we briefly describe how to obtain the MPS representation of a function using TCI.

For matrices, such a decomposition already exists in the form of the *singular value decomposition (SVD).* The goal of SVD is to find a factorization of a matrix $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\dagger}$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices, and $\boldsymbol{\Sigma}$ is a square diagonal matrix with nonnegative entries. These entries are called *singular values* of $\boldsymbol{A}$. The number of nonzero entries r in $\boldsymbol{\Sigma}$ is exactly the rank of $\boldsymbol{A}$. Because of this property, it is possible to represent the full matrix $\boldsymbol{A}$ using r many columns of $\boldsymbol{U}$, r many rows of $\boldsymbol{V}$, and r many entries of $\boldsymbol{\Sigma}$. This is also referred to as a compact SVD.

It is possible to further reduce the amount of information necessary to represent $\boldsymbol{A}$ by allowing the sacrifice of accuracy and obtain a lossy approximation of $\boldsymbol{A}$. By only retaining the largest $r' < r$ singular values, corresponding columns of $\boldsymbol{U}$, and rows of $\boldsymbol{V}$, one obtains an optimal approximation that is as close to $\boldsymbol{A}$ as possible. [6] The resulting SVD is referred to as *truncated* because the original matrix cannot be fully reconstructed based on the decomposition anymore. [4] argue that this leads to a representation requiring fewer evaluations and less memory than the original matrix if $r \ll \min\{m, n\}$.

This approach can be generalized to tensors with more than two dimensions. Tensors can be interpreted as a matrix by regrouping the legs. This is also known as reshaping. All but one leg, the one at site ℓ as shown in Equation 2.7, are joined into a single index. The resulting matrix is then decomposed using a standard matrix SVD. This yields a new tensor train element $\boldsymbol{U}\boldsymbol{\Sigma}$ at site ℓ. The dimensions of the matrices can be adjusted in a similar way to before resulting in a compact or even truncated version. The number of elements retained in this process defines the bond dimension of the tensor train at site ℓ. Then, the matrix can be reshaped into the desired MPS form. After repeating this process, one obtains a compact or truncated version of the original tensor. In general, the minimum required rank, or bond dimension, between elements of the tensor train gives an indication of coupling between the individual dimensions.

The algorithms described for constructing compact versions of matrices and tensors have one crucial downside. Both require knowing the full matrix or tensor, respectively. When approximating a mathematical function with only as few evaluations as possible, starting from the full tensor is unfeasible. In this case of function representation, this is equivalent to having access to all values evaluated at every possible location. This is computationally counterintuitive for representing continuous functions. Instead, it is desired to leverage the structure inherent in the function to reduce the number of evaluations required as far as possible. TCI solves this problem. Instead of truncating a full tensor until a desired compactness is reached, the construction algorithm interactively constructs a tensor with large enough bond dimensions.

2.6.5 Cross Interpolation

The previous section introduced traditional matrix and tensor decompositions, the rationale behind them, and why they are insufficient for representing mathematical functions. In this part, we discuss a different approach described in [14] for obtaining a tensor network representation from sampling a compressible tensor. Although it is not as optimal as starting from the full tensor, this alleviates the necessity of knowing all the items of a tensor before compressing it. The algorithm used to construct the MPS representing the high-dimensional compressible tensor is iterative and reasonably approximates the full tensor over several steps. The process termed *matrix cross interpolation* can be generalized to arbitrary tensors. The rationale behind this bottom-up approach is discussed here. [4] presents several techniques. In the following, we explain one matrix cross interpolation technique. It is the one used during our experiments.

In the previous part, we already discussed the ability to decompose a matrix with SVD. Furthermore, we explained that the columns and rows of the unitary matrices in the decomposition can be interpreted as orthogonal vectors forming a basis of the underlying subspace. The matrix cross interpolation technique is based on cleverly choosing such vectors instead of obtaining them through a decomposition technique. In the cross interpolation method, these vectors are chosen as rows and columns of the matrix to be approximated. Unfortunately, the resulting factorization is not as accurate as an SVD, but the exponential overhead of knowing the full tensor can be avoided.

Formally, matrix cross interpolation can be described as finding a factorization

$$\boldsymbol{A} \approx \tilde{\boldsymbol{A}} = \boldsymbol{C}\boldsymbol{P}^{-1}\boldsymbol{R} \tag{2.8}$$

where $\boldsymbol{C}$ is a matrix containing columns of $\boldsymbol{A}$, $\boldsymbol{R}$ is a matrix containing rows of $\boldsymbol{A}$, and $\boldsymbol{P}$ is the so-called *pivot matrix.* This matrix $\boldsymbol{P}$ contains entries of $\boldsymbol{A}$ at the precise locations where each row and column meet, i.e., at the intersection of each cross formed by a row from $\boldsymbol{R}$ and a column from $\boldsymbol{C}$. Hence the name cross interpolation. At this point, two important observations can be made. First, note that the approximation becomes exact if all respective entries of $\boldsymbol{A}$ are selected by choosing all rows and columns. Second, even if not all rows and columns of $\boldsymbol{A}$ can be retained (this would correspond to storing the full tensor), the approximation becomes exact at the midpoints of the crosses. Therefore, the approximation can also be called interpolation at these points.

A crucial part of the algorithm is the choice of the rows and columns. Both [4] and [17] chose a heuristic approach that maximizes the volume of the pivot matrix, defined by the absolute value of its determinant $\operatorname{vol}\boldsymbol{P} = |\det \boldsymbol{P}|$. It has been proven that this approach yields a good heuristic whilst keeping the pivot matrix nondegenerate, which is important for inverting it in Equation 2.8. The algorithm chooses new sampling points in each step, effectively yielding a locally optimal pivot value.

MCI can be generalized to tensors, similar to the tensor to MPS decomposition discussed in Section 2.6.4.

2.7 Baseline Methods

In order to evaluate a method, one benefits from comparing it to another method. In our case, we resorted to comparing the TCI to a more traditional machine learning method. The next paragraph will introduce the mathematical framework that we used to benchmark and compare the methods.

2.7.1 Machine Learning

In the traditional regression machine learning framework, one is seeking a probability density $p^\star$ by only having access to n pairs of features $\boldsymbol{x}_i \in \mathbb{R}^d$ and target values y_i collected in a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \sim p^\star(y_i|\boldsymbol{x}_i)\}_{i=1}^n \subseteq \mathbb{R}^d \times \mathbb{R}$. In general, $p^\star$ is unknown but can be written as the solution to the optimization problem of maximizing the likelihood of the data

$$p^\star = \arg\max_p p(\mathcal{D}),$$

where the maximum is taken over all probability distributions. In general, it is not possible to optimize over distributions and one commonly restricts the optimization to a set of parametric distributions $\{p_\theta\}_{\theta \in \mathcal{M}}$ where θ collects all p parameters of a distribution on a parametric manifold $\mathcal{M} \subseteq \mathbb{R}^p$. Then, one can maximize the likelihood of observing the data $\mathcal{D}$ on the parametric manifold to find the optimal parameter vector

$$\theta^\star = \arg\max_{\theta \in \mathcal{M}} p_\theta(\mathcal{D}). \tag{2.9}$$

After finding the optimal parameter, $p_{\theta^\star}$ approximates $p^\star$ as close as possible.

By making further assumptions, we can reformulate the optimization goal. First of all, we make the common assumption that our samples stem from the same distribution $p^\star$ and are statistically independent. This allows us to rewrite the joint distribution $p_\theta(\mathcal{D})$ as a product $\prod\limits_{x \in \mathcal{D}} p_\theta(\boldsymbol{x}, y)$ of individual independent factors $p_\theta(\boldsymbol{x}, y)$. Using Bayes' rule, these factors can be written as

$$p_\theta(y|\boldsymbol{x}) = \frac{p_\theta(\boldsymbol{x}, y)}{p(\boldsymbol{x})} \iff p_\theta(\boldsymbol{x}, y) = p_\theta(y|\boldsymbol{x}) \cdot \overset{\nabla_\theta p(\boldsymbol{x}) = 0}{\cancel{p(\boldsymbol{x})}} \propto p_\theta(y|\boldsymbol{x}).$$

Note that we can drop the feature prior term $p(\boldsymbol{x})$ when seeking an extremum on $\mathcal{M}$ since the prior distribution does not depend on the parameter vector θ. Thus, it only acts like a constant multiplicative factor that does not shift the location of the extremum. Second, we chose a normal distribution with unit covariance as our parametric family and parametrize it by a neural network that collects the parameters f_θ,

$$p_\theta(y|\boldsymbol{x}) = \mathcal{N}(f_\theta(\boldsymbol{x})|y, I) \propto \exp\left(-\frac{1}{2}\|f_\theta(\boldsymbol{x}) - y\|_2^2\right),$$

where $\|\cdot\|_2$ denotes the L_2-norm. With these tools and acknowledging that the logarithm is monotonic, we can simplify Equation 2.9 into

$$\begin{aligned}
\theta^\star &= \underset{\theta\in\mathcal{M}}{\arg\max} \prod_{(\boldsymbol{x},y)\in\mathcal{D}} p_\theta(y|\boldsymbol{x}) \\
&= \underset{\theta\in\mathcal{M}}{\arg\max} \left\{\log\left[\prod_{(\boldsymbol{x},y)\in\mathcal{D}} p_\theta(y|\boldsymbol{x})\right]\right\} = \underset{\theta\in\mathcal{M}}{\arg\max} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \log p_\theta(y|\boldsymbol{x}) \\
&= \underset{\theta\in\mathcal{M}}{\arg\min} \left[-\sum_{(\boldsymbol{x},y)\in\mathcal{D}} \log p_\theta(y|\boldsymbol{x})\right].
\end{aligned}$$

This common form indicates that maximizing the likelihood is equivalent to minimizing the negative log-likelihood. We can now substitute the term involving $\log p_\theta(\cdot)$ and drop all the constant multiplicative factors because they do not affect the optimization:

$$\begin{aligned}
&\underset{\theta\in\mathcal{M}}{\arg\min} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} -\log p_\theta(y|\boldsymbol{x}) \\
=&\underset{\theta\in\mathcal{M}}{\arg\min} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} -\underbrace{\log\exp}_{\text{identity}}\left(-\frac{1}{2}\|f_\theta(\boldsymbol{x}) - y\|_2^2\right) \\
=&\underset{\theta\in\mathcal{M}}{\arg\min} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \frac{1}{2}\|f_\theta(\boldsymbol{x}) - y\|_2^2 = \underset{\theta\in\mathcal{M}}{\arg\min} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \mathsf{MSE}(f_\theta(\boldsymbol{x}), y) \overset{\text{def}}{=} \underset{\theta\in\mathcal{M}}{\arg\min}\, \mathcal{L}(\theta).
\end{aligned} \tag{2.10}$$

This resembles the form of the well-known mean-squared loss function used in regression analysis. The last line holds because of the symmetry of the L_2-norm. In the next section, we describe how this optimization problem can be solved numerically, and a set of parameters that minimizes the objective function $\mathcal{L}$ can be found. A similar derivation can be found in many standard machine learning textbooks, like [13].

2.7.2 Numerical Optimization

A local extreme point of Equation 2.10 can be found by setting the gradient of the loss function $\nabla_\theta \mathcal{L}(\theta)$ to zero and solving for the parameter vector θ. Unfortunately, this is not feasible in an analytical fashion for many machine learning models. Therefore, one usually resorts to numeric optimization techniques. Since the objective function $\mathcal{L}$ is differentiable and the model f_θ as well, we can use the traditional gradient descent scheme of iteratively updating a set of parameters $\theta^{(k)}$ using the well-known rule

$$\theta^{(k+1)} \leftarrow \theta^{(k)} - \alpha \cdot \nabla_\theta \mathcal{L}(\theta)|_{\theta=\theta^{(k)}},$$

where α is the step size. By repetitively following the steepest descent direction, $\theta^{(k)}$ is assumed to have converged to a local minimum. Today's machine learning libraries like JAX [2] have builtin support for computing the descent direction $\nabla_\theta \mathcal{L}(\theta)|_{\theta=\theta^{(k)}}$.

3 Interpolating Potential Energy Surfaces

Our experiments combine traditional quantum chemistry tools with an interpolation technique. This allows the PES to be evaluated only in a necessary number of evaluations. The following part describes the overall experimental setup, the studied systems, and the inner workings of our experimental and benchmarking pipeline.

3.1 Motivation

Quantum chemistry has been tasked with finding, representing, and approximating energy surfaces for all of its history. In this part, we present selected approaches and highlight differences in our approach.

PESs have already been studied by applying interpolation techniques. Ischtwan and Collins [11] introduced a method for constructing PES using a moving interpolation technique. The approach integrates ab initio energy, gradient, and second derivatives calculations at dynamically significant configurations. Their algorithm ensures convergence with an increasing number of data points and emphasizes efficient placement, guided by trajectory sampling in relevant regions of the configuration space. The authors report that their method works well for reaction dynamics. They apply the approach to a six-dimensional PES for a reaction that shows the framework's potential. For this method to work, a reaction path must be known before optimization. From this work, we conclude that an effective choice of sample points is crucial for interpolating a PES.

QTCI provides such a sampling scheme. Therefore, in this thesis, we explore the application of an interpolation technique to quantum chemistry simulations. We hypothesize that QTCI excels at finding an accurate representation of a PES by cleverly choosing regions to sample from. The existing quantum chemistry tools are used to compute the energy at the specific location. With that, we aim to find an accurate representation of the PES, stored in an MPS tensor combined with a quantics representation.

We are extending the set of tools and exploring a novel combination of tools that we believe will yield promising results in accurately representing molecular PESs while remaining sample-efficient.

3.2 Querying Potential Energy Surfaces

Our initial goal was to obtain a model of the ground state energy surface and the surface of the first excited state. This would allow for finding regions of interest where the two surfaces touch. In photochemistry, this corresponds to a conical intersection as introduced before. In our exploration, we experimented with a variety of different quantum chemistry tools from `pyscf` [20]. A brief overview of the methods has already been given in Chapter 2. For the experiments, it is important to note that we use the quantum chemistry tools to generate reference points on the ground state surface $E_0^\star$ and the first excited state surface $E_1^\star$. These are not particularly special surfaces but can be obtained from most quantum chemistry packages without special tricks. Because of the variational limitations discussed in Section 2.1.2, the methods themselves cannot compute the electronic states without error. Nevertheless, they are the best estimate available and they will therefore serve as a ground truth.

In the following, we will briefly summarize the quantum chemistry tools used in our experiments. Note that all our experiments were expressed as `Python` code.

3.2.1 Ground State Solvers

Many quantum chemistry tools use a Hartree-Fock solver to estimate a ground state. Note that in our experiments, we are fortunate not to have to be concerned with the absolute accuracy of the Hartree-Fock solution. In general, this solution is only a crude approximation of the actual ground state and, thus, its energy. It is common to take the Hartree-Fock solution as a starting point for finer solutions [16] or to find excited states on top of its solution, like TD-HF [5]. The solvers serve as initial evidence that the computational framework is doing something reasonable. In our experiments, we used the Hartree-Fock method and the DFT solvers available for ground state computations.

3.2.2 Excited State Solvers

For our experiments, we were studying the energy gap between PES. Therefore, we used quantum chemistry methods that either compute orthogonal solutions of the Schrödinger equation at once or extend a ground state solution to include excited states.

3.3 Surface Interpolation

As described before, our interpolation technique of choice is the QTCI. We use the `xfac` package [7] to run the TCI algorithm in combination with a built-in quantics representation. The code is written in `C++`, but the package provides bindings for using the QTCI in Python code. Furthermore, the authors also provide an implementation in Julia. Since our experimental framework uses the pyscf quantum chemistry package, it was a natural choice to use the Python bindings and integrate the QTCI module into our code.

The xfac package expects the function to be interpolated, in our case, the energy functions, as a callback returning a single scalar value. This makes it necessary to wrap all the calls to the computational chemistry package in a single function. Our code makes it possible to generate such functions solely by providing a system definition, a solver, and an experimental configuration. The solver can be any function that maps a quantum chemical system configuration to a scalar value. In our case, these were the energy approximation programs provided by pyscf. The experimental configuration includes parameters of the quantum chemistry algorithms, like tolerances or convergence thresholds. The code developed during this thesis can be found in the accompanying repository. [1]

3.4 Studied Systems

System	DoF	Parameter Range	Basis Set
Hydrogen chain H_n	$n-1$	$[0.5, 8]^{n-1}$Å	sto6g
LiF	1	$[0.5, 8]^{n-1}$Å	augccpvtz
Formaldimine $H_2C{=}NH$	2	$(\alpha, \phi) \in ([0, \pi] \times [0, 2\pi])$	sto3g

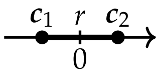
Table 3.1: An overview of the quantum chemical systems showing the degrees of freedom (DoF), the parametrization, and the quantum chemical basis set.

Our experiments were conducted using different quantum chemistry tools and different quantum chemical systems. For the experiments, it was necessary to express them as a function mapping a D-dimensional input coordinate to an energy value. These are the functions $E_i : \mathbb{R}^D \to \mathbb{R}$ introduced in the previous sections. In order to compute the energy one has to map the coordinate to an actual instance of a quantum chemical system. In general, we created a code that maps a D-dimensional input coordinate to a molecule representation suitable to the treatment with pyscf. In our experiments, we either parametrized bond lengths or angles based on previous findings, which will be discussed in the following by talking about specific systems. A short summary of the systems is given in Table 3.1.

[1] It is available at https://github.com/neftlon/masters-thesis-code.

3.4.1 Dimer Molecules

The first system that we considered is a molecule dimer, which consists of two atoms. The two atoms comprising the system can be joined by a shared bond, which is parametrized by a single *bond-length* r which is a positive scalar. Therefore, these systems belong to a class having an energy function taking a one-dimensional argument. The dimer can exist anywhere in $\mathbb{R}^3$, but because of the rotational and translational invariance of the energy operator, we restrict ourselves to the simple instances on a single coordinate axis. One can map r to two coordinates, for instance $\boldsymbol{c}_1 = \begin{bmatrix} 0 & 0 & -\frac{1}{2}r \end{bmatrix}^\top$ and $\boldsymbol{c}_2 = \begin{bmatrix} 0 & 0 & \frac{1}{2}r \end{bmatrix}^\top$, that are exactly $r = \|\boldsymbol{c}_2 - \boldsymbol{c}_1\|_2$ apart. With this definition, the coordinates for both atoms are in a straight line, as shown in the following scheme.



This allows the approximation of the ground state using a QTCI representation.

For the experiments, we chose to use the H_2 and LiF dimer molecules. The first one was chosen because it is easy to setup and model with quantum chemistry methods. The second molecule is a bit harder to model because of its increased number of electrons, but it possesses a conical intersection. The conical intersection in this system was reported, for instance, in [12].

3.4.2 Hydrogen Chains

A natural extension of the dimer molecules was to consider hydrogen chains because they have two important properties despite not being physically very plausible. First, they allow easy control of the number of degrees of freedom. Second, hydrogen atoms are comparably easy to model and include in calculations, making scaling to a higher-dimensional system computationally feasible. We modeled a hydrogen chain of n hydrogen atoms to lie on a straight line, similar to our dimer configuration. There are common parameterizations available for the hydrogen chain, for instance in [18], where the chain is parametrized by a bond length and a spacing parameter, or uniformly separated atoms controlled by a single parameter like in [16]. Our parametrization yield a molecule configurable by $n-1$ degrees of freedom where n is the number of atoms in the chain.

3.4.3 Formaldimine

As alluded to in Chapter 1, formaldimine is a model of a photosensitive protein in the human eye. The key part that makes this system so interesting for studying photochemical reactions is its behavior around the conical intersection in its potential energy surface as reported in [1]. Because of the touching point between the ground state and excited state surfaces, the molecule can move from one conformation into another solely by excitation. This excitation can be caused by the interaction with a photon, which is the driving force for processes in the human eye as well. Because of its reasonable simplicity, we study this molecule in this thesis.

Formaldimine is a favorable system to study because it can be expressed using two degrees of freedom as reported in [1]: a dihedral angle ϕ and a bending angle α. The core of the molecule is formed by a double bond between a carbon atom a nitrogen atom. The carbon atom contains two additional hydrogen atoms and the nitrogen atom one. Our parametrization keeps the overall arrangement of the molecule fixed. The only atom that is able to change by the two-dimensional conformation is the hydrogen atom attached to nitrogen. The bending angle α essentially dictates how much the hydrogen atom is lifted from the plane spanned by the remaining molecules. The dihedral angle indicates how the atom is rotated around the molecule. The two angles allow the convenient depiction of the molecules by an image where the color indicated the energy for the respective conformation.

3.4.4 System Preparation

One important problem was the range conversion from our chosen coordinate frame to ranges suitable for the QTCI. In general, a QTCI can interpolate any function that is defined on a half-open interval $[a, b)^D$. Note that every coordinate has to be in the same range, which was problematic for systems with different parametrizations. For instance, formaldimine's parametrization has two angles, but both have different angular values that are admissible. Therefore, it is impossible to create a single bijective mapping that converts both angles into the range $[0, 1)$. Since our value ranges are vastly different among systems, we added a system-dependent preprocessing step that maps a coordinate vector from a molecule-specific range to a D-dimensional vector in the interval between zero and one, namely $[0, 1)^D$. More concretely, this allows mapping values from, for instance, angular ranges like $[0, 2\pi)$ to the required range $[0, 1)$ for the QTCI. The same applies to the bond length. Using this system, the QTCI only needs to be informed of the system's number of degrees of freedom because the base range $[a, b)^D = [0, 1)^D$ is always the same.

3.5 Inspecting Energy Gaps

In our first approach, we use a QTCI to approximate either of these surfaces separately. Concretely, we obtain an interpolation for both surfaces E_0 and E_1. This entails that two interpolations must be fitted separately to have reasonable representations of both electronic states. Many quantum chemistry tools compute the energies of multiple electronic states at the same time without any additional computational overhead. Thus, the first approach is not optimal because it essentially wastes the excited state information when fitting the ground state surface and vice versa. It is possible to halve the computational overhead by directly modeling the energy difference and using the full information given by the respective quantum chemistry tool. For that, we only learn a single interpolation that represents the difference $E_1 - E_0$.

4 Experiments and Results

Here, we show the results that were obtained throughout the project phase of the thesis using the methods described in the previous section. We start by presenting one-dimensional and example systems. Step by step, we will explore more complex methods and examine the capabilities of QTCI.

4.1 Smoke Test

The first experiments were conducted by reproducing examples from the publication introduction of the QTCI method [17]. These examples serve as very illustrative examples of the potential of QTCI. Figure 4.1 shows a plot of the function

$$f(x) = \cos\left(\frac{x}{B}\right)\cos\left(\frac{x}{4\sqrt{5}B}\right)\exp\left(-x^2\right) + 2\exp(-x)$$

for $B = 2^{-30}$. The two cosine terms in the beginning account for the high-frequency oscillations that are visible only on a scale as tiny as $\left[0, 2^{-23}\right]$. The second part of the sum describes an exponential fall-off that is responsible for the global behavior on a bigger scale for $x \in [0, 3]$. Both behaviors are captured using QTCI, as shown in the plot. This experiment shows that QTCI is capable of interpolating functions that exhibit scale separation.

4.2 One-Dimensional Ground State

The subsequent experiment's goal is to transfer the quality of the one-dimensional example function to a simple PES without excited states. For a chemical dimer that consists of two atoms that are joined through a bond, the system's only degree of freedom is the distance of the two nuclei. Therefore, one can interpret the dimer's PES as a one-dimensional function mapping the atomic separation r to an energy value. We used different dimers to experiment with one-dimensional systems. Among these were H_2 and LiF, of which we discuss the corresponding experimental results in the following. The simplest quantum chemical computation is using the Hartree-Fock method to approximate the ground state energy. The Hartree-Fock method returns an

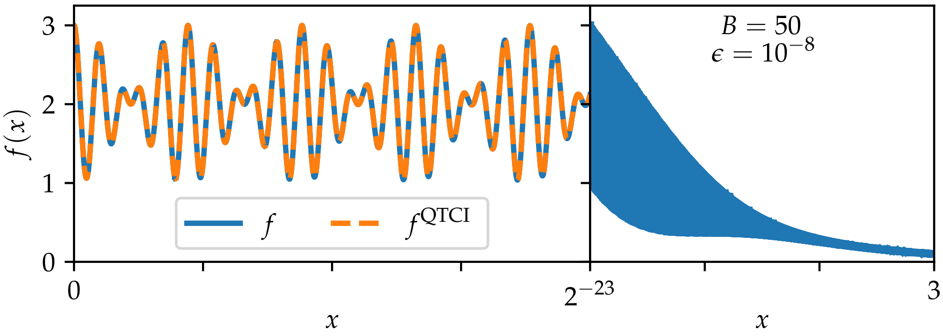


Figure 4.1: A plot comparing a QTCI with the interpolated original function where the QTCI shines because of scale-separation. The left part shows the local behavior of f with rapid oscillations on a tiny interval. On the right side, the global behavior is shown for values of a bigger length scale on a much larger interval. On a global scale, rapid oscillations are no longer noticeable. (The figure is reproduced from [17] using our computational framework.)

energy value given a set of nuclear coordinates. Therefore, our first goal was to model the ground state surface of a dimer.

The results of interpolating the PES of the dimer systems are shown in Figure 4.2. For the experiment, a QTCI with a resolution of 16 bits was used and fit upon a relative tolerance of 10^{-8}. All bond dimensions stayed well within the predefined upper bound of $L_{\max} = 15$. The energy plot was created by evaluating the fitted QTCI on a uniformly spaced grid of 100 points. It shows that the QTCI is capable of representing the PES.

In order to evaluate the accuracy of the approximation, the second row in Figure 4.2 shows the absolute errors comparing the predicted energy E^{QTCI} with the reference energy E of the Hartree-Fock method by taking the absolute difference. Hence, the plot shows the squared-error $\left\| E^{\mathrm{QTCI}}(r) - E(r) \right\|_2^2$ evaluated for r on the same grid that was used to plot the energy function. Most of the errors are considerably low for both systems, hovering around 10^{-12}. This demonstrates that the QTCI is capable of accurately representing the PES of these two dimers.

Interestingly, the interpolation algorithm lacks confidence about energy values close to the borders. This can be seen in the frequency plot in the last row of Figure 4.2. For both systems, the borders are sampled with very high frequency. Similar to the borders, the area around the minimum of the ground state surface is densely populated as well. This is to be expected because this is the area with the most sudden change when going from one conformation to another. The QTCI needs a lot of samples to

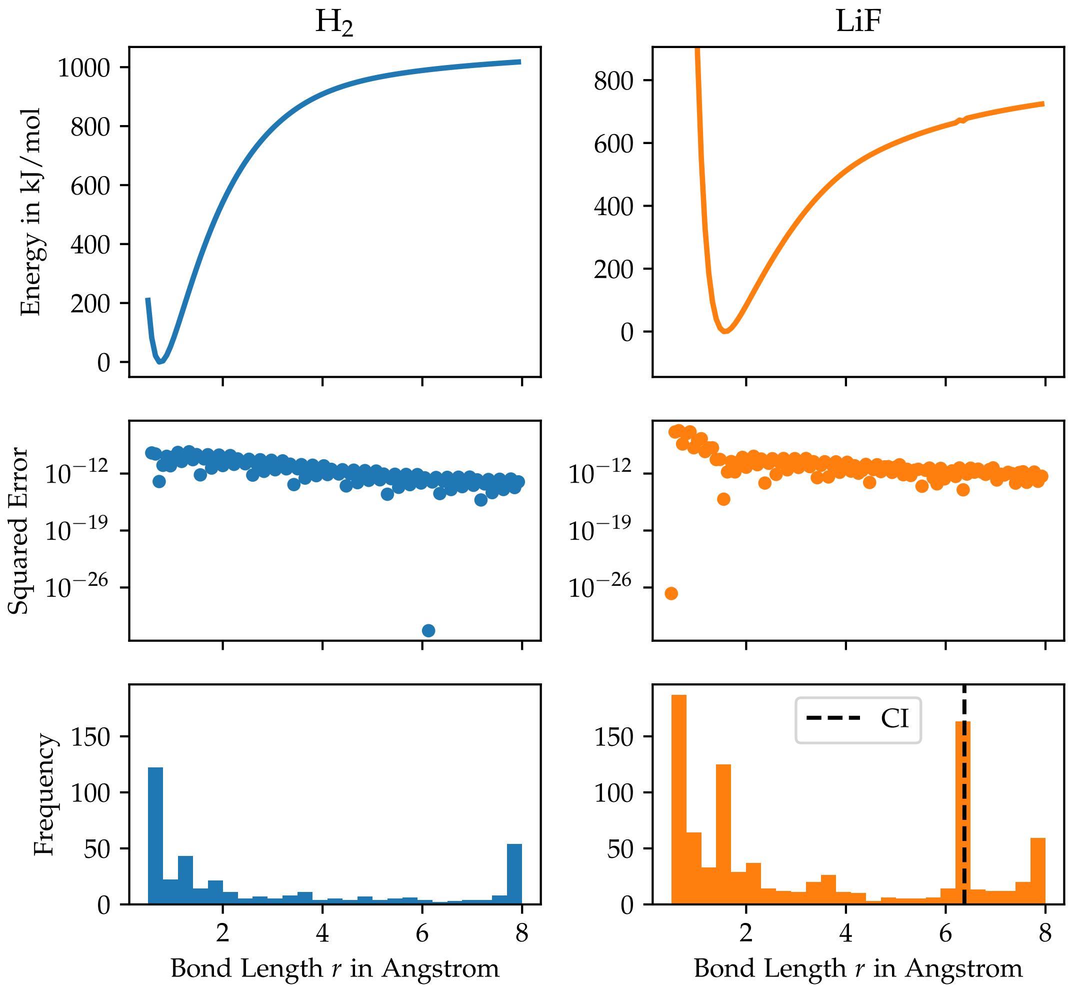


Figure 4.2: A plot showing different aspects of H_2 (left column, blue data) and LiF (right column, orange data). For both systems, different results depending on the bond length $r \in [0.5, 8)$ in Angstrom are shown: The interpolated energy in the first row, the absolute error on a logarithmic scale in the second row, and the frequency of requested QTCI samples in the last row shown as a histogram. The frequency plot in the bottom right corner contains a marker at $r = 6.375$ for the location of the conical intersection (CI) in the LiF system.

reduce the prediction error in this region. Another observation is that close to the conical intersection of LiF at around 6 Å (as described by [12]), the QTCI requests a lot of samples. This manifests in a peak in the histogram for the corresponding bond length, which we marked in the plot.

4.3 Excited State Exploration

For our approach to modeling conical intersections, we aimed to model the gap between the ground and excited states. Therefore, the next step was to investigate modeling excited states. Ideally, we can model the PES difference of a molecule at sufficiently many locations such that QTCI provides a cheap representation of the difference surface. Unfortunately, this demand cannot be met, and we will discuss our path in the following section. In Section 3.2.2, we already presented the various solvers we experimented with. The following part shows our experimental path. We first started to use common quantum chemistry tools and then extended our range of methods with more sophisticated tools. We found strong differences in the quality and performance of the methods.

4.3.1 Natural Extension

After obtaining a ground state with Hartree-Fock, extending the solution to include one or more excited states using the TD methods described in Section 3.2.2 is very easy. To get started, we again looked at LiF because of its one-dimensional PES and the conical intersection it possesses.

Figure 4.3 shows the first attempts to study both the PES and the interpolations generated of LiF with QTCI. The PES looks well represented on a large scale, but close inspections reveal subtle discrepancies. These cause the problem to be harder to solve than necessary for QTCI. The plot highlights two regions of the interpolated PES, both showing physically unrealistic behavior, which will be discussed in more depth in the following.

The red region shows that the ground state suddenly jumps to an excited state. This unphysical break of continuity in the potential energy surface can be traced back to the method that generates the ground state surface generated by the Hartree-Fock method. The choice of methods can be questioned, but it cannot be argued against the QTCI. Quite the opposite is the case: The QTCI is able to accurately represent the function given as its input.

Figure 4.4 plot shows the samples requested by the algorithm. Note that the discontinuity is present in the sampled surfaces as well. Therefore, the QTCI can accurately

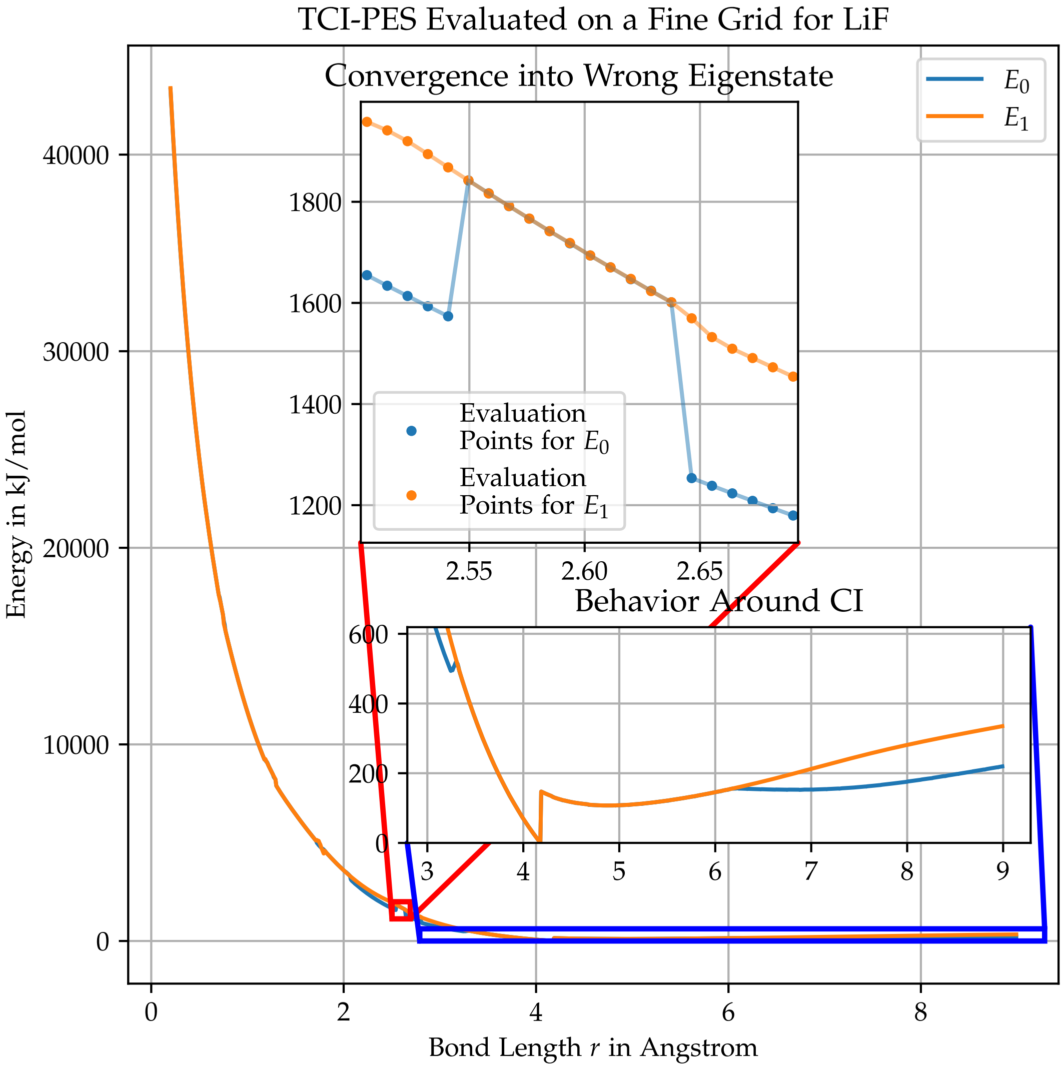


Figure 4.3: A plot showing the surface learned by a QTCI approximation parametrized by the bond length of the dimer. The inset axes shown in red and blue indicated areas where the quantum chemistry method exhibits unphysical behavior due to unstabilities.

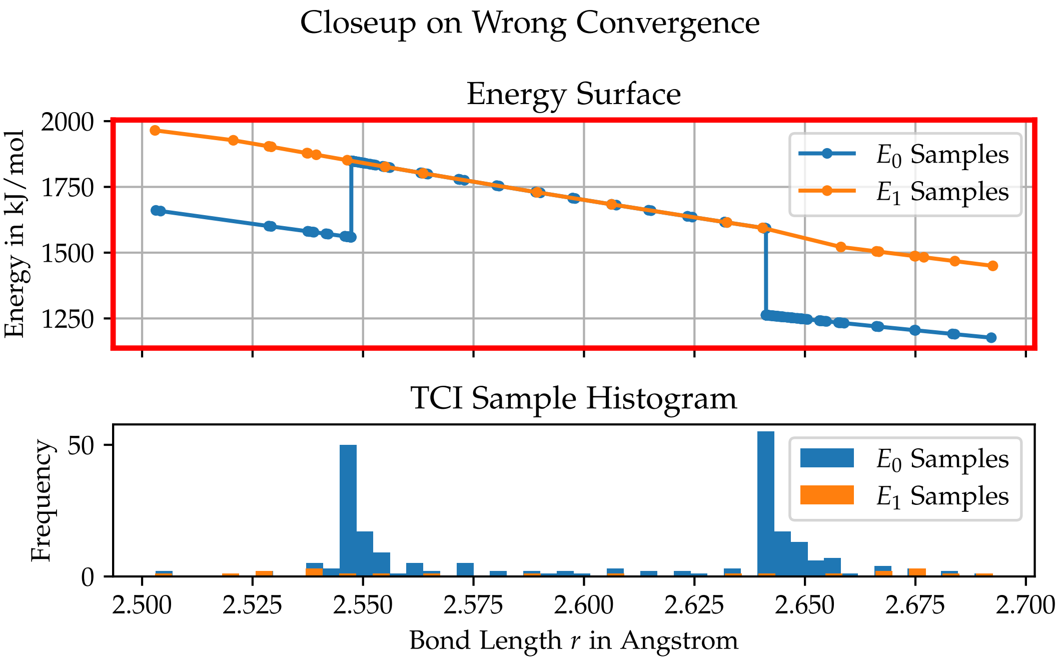


Figure 4.4: A plot showing the samples requested by QTCI in the red region from Figure 4.3. The plot on the top shows the energy surface where the sampled locations are marked with dots. The bottom plot shows the frequency at which samples were requested at the respective bond length. Note that a large number of samples have been requested around the discontinuity.

represent the PES, but the underlying method is incapable of providing high-quality data.

Another important point that can be extracted from the red region is that the QTCI requests samples close to the edge of the jump, visible in Figure 4.4 as well. The TCI always samples points that contribute to high pivot error. The rationale behind this is to pick places that have the most impact on the error, and thus have the ability to reduce it substantially when being corrected by TCI. A discontinuity in the modeled surface is a source of high pivot error where the QTCI has to first learn the kink induced by the underlying model. Thus, the TCI tries to reduce the error by sampling values as close as possible to the jump in order to find the discontinuity. Not only does this correspond to unphysical behavior, but it also wastes a lot of samples for modeling a problem caused by the underlying model's quality.

We show another unphysical behavior in the blue part of Figure 4.4. Here, a cusp in the ground state surface is shown. Furthermore, the ground state and the excited state share the same curve over various amounts of bond lengths. In theory, the conical intersection should be only one point, or a small touching region. This could not be reproduced by TD-DFT, which fails to converge into the correct eigenstate. Instead, both states are represented by the same eigenstate, which corresponds to unphysical behavior. For large bond lengths, at around 6 Å, the curves of E_1 and E_0 separate again, which is desired.

4.3.2 Comparison to Stable Method

The previous part proved that the discrepancies of the underlying method manifest in the interpolation as well. Next, we inspected the overall quality of the interpolation. Figure 4.5 compares the samples generated with TD-DFT with those generated by a method that is expected to perform better in terms of stability. In particular, the sampling locations in this plot are not requested by a TCI algorithm, they are uniformly spaced across the values on the axis of the bond length. This allows for the direct comparison of the two methods used, TD-DFT and DMRG-SCF.

First of all, it becomes evident that the energies generated by TD-DFT are not very stable. Especially in the region around 4 Å, TD-DFT generates different energy values over a range of values. This is not in agreement with the continuous nature of the potential energy surface. This is especially hard when considering that QTCI is an interpolation. This means that it is able to memorize the values of samples and approximate the surface between the samples. Because of the unpredictable nature of TD-DFT, this makes it very hard for a QTCI to converge, as described previously in Section 4.3.1 where the unphysical jumps were supervised in a highly accurate but ultimately wasteful manner. The instability of TD-DFT has been reported previously in

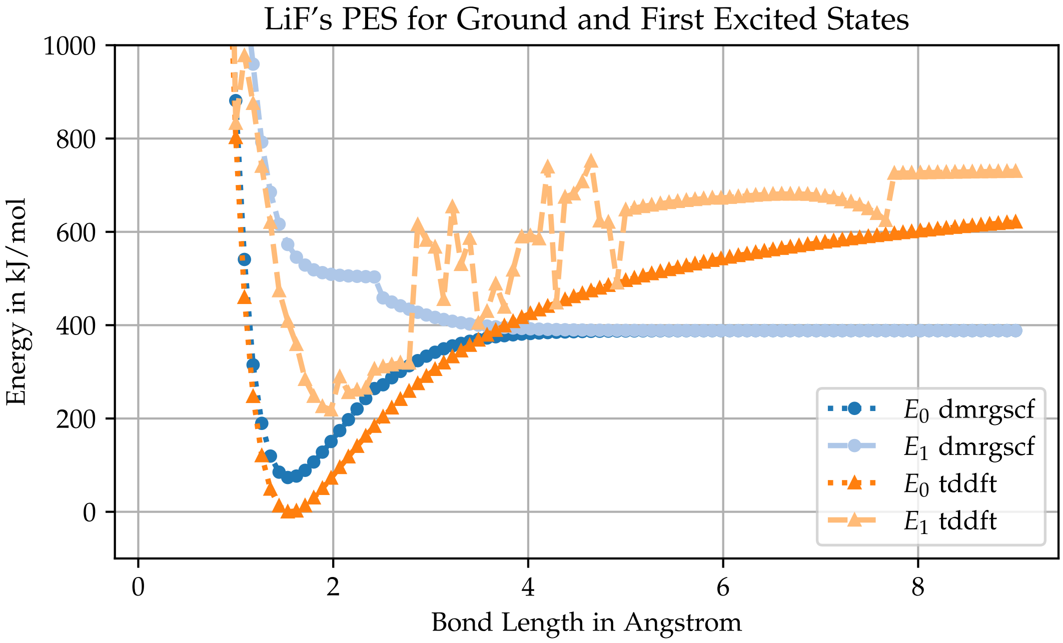


Figure 4.5: A plot showing the samples generated by the TD-DFT and DMRG-SCF for LiF at various bond lengths. It becomes evident that TD-DFT is very unstable for computing the excited state surface.

[9].

Furthermore, it is interesting to see that the energy surfaces are very different. Both curves seem to diverge as expected for bond lengths close to zero. Both curves seem to have different extrema and behaviors for large bond lengths. Whilst the surfaces of DMRG-SCF attain each other for bond lengths that are large enough, the curves generated with TD-DFT are largely separated. Nevertheless, we believe that methodological differences can explain this and should not be too concerning for this thesis.

Both surfaces exhibit continuous but not differentiable bends in the surfaces. For DMRG-SCF, the excited state curve changes rapidly at around 2 Å, and the same happens for TD-DFT's respective curve at around 8 Å. We believe that is due to a rapid change in the used orbitals in the basis set. Note that these are regions, where the QTCI will have problems approximating the continuous surface.

Lastly, what is uncommon is that the energy curve of TD-DFT is below the DMRG-SCF. The latter is a method that should, upon the right configuration, yield a better result. In terms of quantum chemical minimization, this corresponds to finding lower-lying energy eigenvalues that are closer to the true ground state. Since the curve of DMRG-SCF is above that of TD-DFT, we theorize that the algorithmic configuration of our instance of DMRG-SCF is not optimal. We used DMRG-SCF in a configuration where it focused on the optimization of only some orbitals in a smaller active space. This is a common technique to simplify the computational overhead in quantum chemistry methods. We believe that the higher energy result with DMRG-SCF could be due to picking a $(2, 2)$ active space instead of solving the full system. Nevertheless, the surface generated by DMRG-SCF is smoother. Therefore, we opt for using DMRG-SCF in subsequent experiments with a two-dimensional molecule.

After inspecting the performance of TD-DFT, as discussed in this part, we investigated more complex methods closely. It was very important to us to use a method that is computationally feasible. Therefore, we first measured the wallclock durations of various methods, which will be discussed in the next section.

4.4 Trade-Off with the Experiment Duration

We found, that even for the simplest systems, the experiment durations vary drastically. Given that the QTCI does not give an estimate of how many samples it requests in a given run, the experiment duration is rather unpredictable. We tried fencing in the duration as much as possible. We took the average time per sample as a metric, as shown in Figure 4.6. The plot gives an idea of which methods can be expected to yield a feasible runtime. The data was obtained by running the different methods

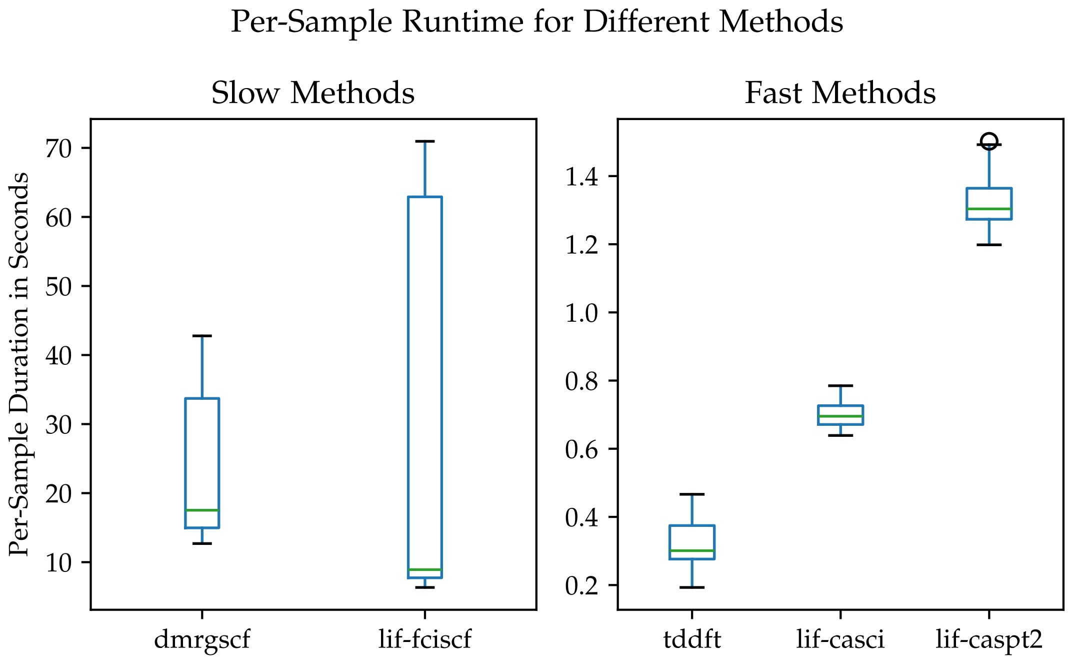


Figure 4.6: A plot showing the per-sample time for various methods. The plot is divided into two groups based on the length scale of the runtime. The fast methods all have runtime lengths below 2 s while the slow methods take times on the scale of minutes.

with 100 random samples on the LiF system. We chose this system because it is right in the middle of our example systems in terms of complexity for computational modeling. Note that we added additional methods for comparison that we tweaked to be performant with LiF. In the figure, these are prefixed by `lif`. These methods use a $(6, 6)$ active space configuration that is selected just for this system.

In the bar plot, we compare different quantum chemistry methods and their respective sample times. The methods can be divided into two categories. In the plot, these categories are labeled as either "fast" or "slow". It becomes evident, that the methods with a heavier computational overhead need to be considered slow. This was to be expected, but the measurements make it possible to at least estimate the runtime duration of experiments with these methods.

In the speed comparison, we also included methods that incorporate unfavorable scaling, such as full configuration interaction (FCI). As expected, TD-DFT is the fastest method and a FCI approach takes the longest. Interestingly, the DMRG-based SCF computation is almost as slow as running FCI.

4.5 Experiments with Formaldimine

In this part, we will describe our experiments with formaldimine. First, we document the initial exploration, which is when we compute a ground truth dataset. Then, we explain the different instances of QTCI, which we used to interpolate the surface.

4.5.1 PES Evaluation

As stated in Section 3.4.3, we also explored the gap between the ground state surface and the first excited state of formaldimine. Because of the limitations of TD-DFT explained in the previous part, we had to use a more stable approach for computing the PES of formaldimine.

Figure 4.7 was generated using DMRG-SCF evaluated on a regular grid of 50×50 pixels. For formaldimine, we picked an $(3, 4)$ active space similar to [1]. Each pixel represents an angle pair, which is a parameter vector for a unique conformation of formaldimine. Embedded into our framework, DMRG-SCF outputs two energies for each pixel: The ground state energy and the first excited state's energy. In Figure 4.7, the ground state surface is shown in the image on the left, and the excited state surface is shown in the middle. From the plot, it becomes evident that the ground state surface contains two wells. This corresponds to two minima in the depicted region of the image. Since the angular configurations are cyclic, more configurations exist for which the energy surface has minima. The difference between the two energy values is shown

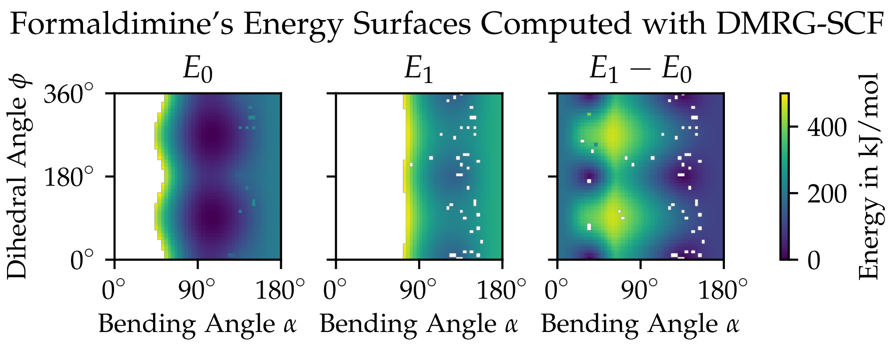


Figure 4.7: A plot showing the PES of formaldimine computed with DMRG-SCF evaluated on a two-dimensional grid. Different energies are indicated by different colors. Samples in white indicate energy values above 500 kJ mol^{-1}. These large values are excluded because their inclusion would distort the regions of interest, hence rendering them unrecognizable.

in the last plot. The minima of the difference allude to a touching point of the two surfaces.

After verifying the quality of the generated PES, we found it qualified to advance to the next step in which the difference between the energies is interpolated using QTCI. The hope for QTCI is to require fewer samples than to evaluate the full grid as done for the plot. Essentially, the QTCI should need less data than the method evaluated on the entire grid.

4.5.2 Interpolation with QTCI

After establishing and inspecting an experimental framework for the formaldimine molecule, we interpolate the PES using QTCI. In order to compare different versions, we use different algorithmic configurations. In traditional machine learning parlance, this is called hyperparameter tuning. Note that we did not spend much time optimizing the quantum chemistry toolchain since our focus lies on the interpolation with QTCI. The following part documents different configurations of QTCI and their impact on the interpolated surface.

Interpolation Training

As described in Section 2.6.5, the algorithm to obtain a QTCI can be split into different sweeps. The algorithmic framework we used allows for investigating the state of the interpolation within the sweeps. This can be thought of as looking at the interpolation

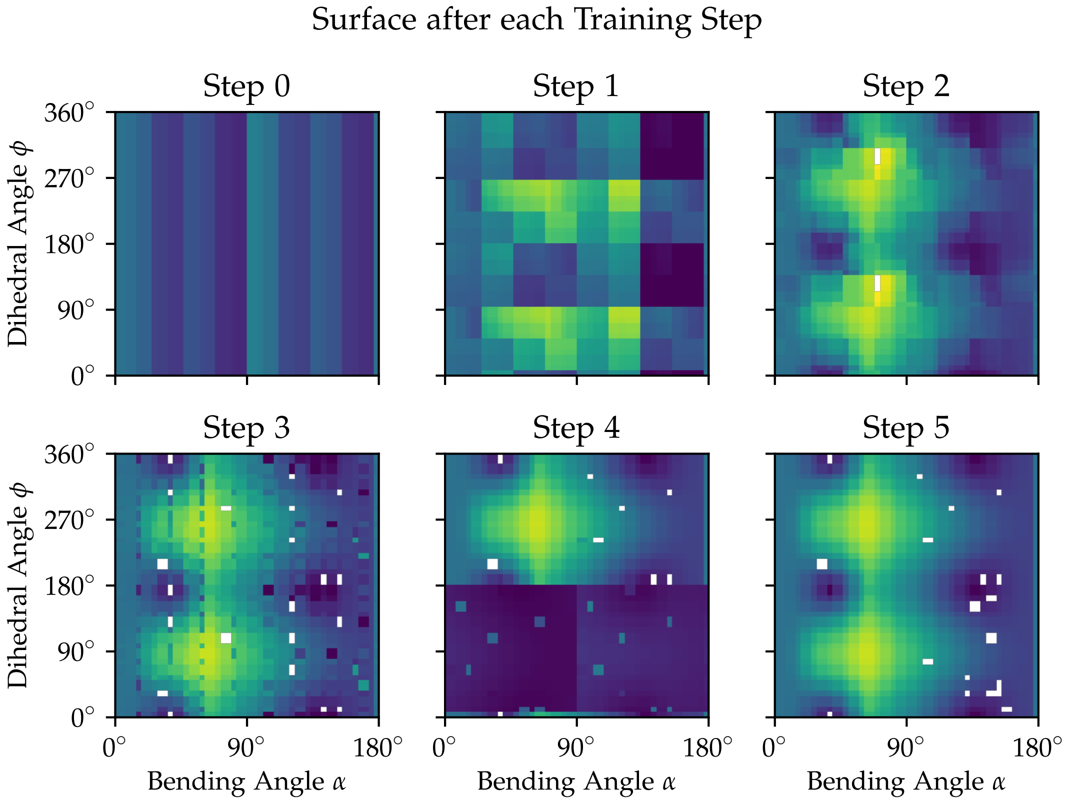


Figure 4.8: This plot contains an example of a training run where a QTCI was fitted to formaldimine's PES. Step number zero shows the QTCI before training. It is only exact for a few initial pivot points. The subsequent steps yield an interpolation of the actual PES by iterative refinement.

at various stages of optimization. Initially, we verified that the QTCI interpolation actually is progressively enhancing an initially bad interpolation over various steps. The initial findings are shown in Figure 4.8, where we fixed one algorithmic configuration and evaluated the QTCI on a regular grid after each step. We obtain similar results for other algorithmic configurations, and this example should only serve as an indicator that can be generalized to other configurations. Note that the method that generates the energy difference values on the PES was always the same across different configurations. We used the DMRG-SCF method that we investigated formaldimine's PES with before in Section 4.5.1. The method was evaluated on the same 50×50 grid that we used previously to generate the dataset. The algorithm itself has at no point been informed about the specific grid that we picked.

By quantitative observations, one can already see that the surface is continuously refined. Starting at a very crude approximation in the leftmost column in the plot, the QTCI converges to a reasonable representation as shown in the column after the last iteration. This shows that the TCI algorithm is able to iteratively refine its representation up to the maximum accuracy possible at the given bit depth.

After this initial verification, we conducted further experiments with different algorithmic configurations. The most important parameters of the QTCI algorithm are the maximum bond dimension, the convergence threshold, and the bit depth of the quantization. In the following, we show the results of changing different algorithmic parameters.

Different Bit Depths

First, we altered the number of bits while fixing the maximum bond dimension. We immediately noticed that a very high number of bits led to the algorithm getting stuck. Therefore, we restricted the number of bits to small numbers. Figure 4.9 shows two trained QTCIs, one with a four-bit representation and one using five bits. The reference dataset from before is shown on the right. For four bits, the possible number of sample locations is 2^4 for every dimension. Therefore, the two-dimensional surface has 256 unique locations after discretizing the angular coordinates.

Note that the functions approximated in Section 4.1 and [17] exhibit oscillations on vastly different length scales. This does not apply to the PES we studied in our experiments. This does not corrupt our claim that QTCI can accurately represent those surfaces. Nevertheless, we hypothesized that the QTCI does not require a large number of bits because the surface can already be represented accurately without scale separation. Therefore, the examples shown in the plot use comparably low numbers of five bits per dimension.

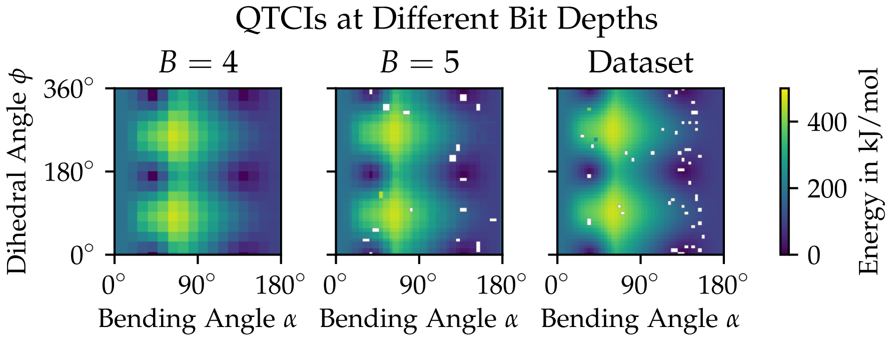


Figure 4.9: A plot showing the trained QTCIs with two different numbers of bits B. The rightmost column contains the reference dataset.

Different Bond Dimensions

Table 4.1: The algorithmic hyperparameters for QTCI as shown in Figure 4.10. In addition, the corresponding number of samples requested by the interpolation algorithm and the MSE on our reference dataset are shown.

Bit Depth	Max. Bond Dim.	Rel. Tol.	Number of Samples	Dataset MSE
5	5	10^{-3}	315	2.3401×10^{-2}
5	20	10^{-2}	532	1.4253×10^{-2}
5	20	10^{-3}	970	1.3825×10^{-2}
5	100	10^{-3}	994	1.4419×10^{-2}

The full hyperparameter specification for the experiments is summarized in Table 4.1. The parameters were chosen rather arbitrarily to get an idea of the algorithm's behavior in the low-bit-depth regime in which we wanted to model the PES. In the following, we present our findings with respect to variations of different parameter configurations.

Despite the reasonably good representation of the PES, we also observe a collapse in the representation before converging. By collapse, we refer to an iteration in which the interpolation behaves worse than in a previous iteration. This can be seen in Figure 4.8, where step number four provides a not yet complete but compelling-looking interpolation. The following step completely discards the well expected for dihedral values $\phi > 180°$. Although the well reappears in the subsequent step, we have to note that the interpolation would behave very badly if the interpolation had terminated one step earlier.

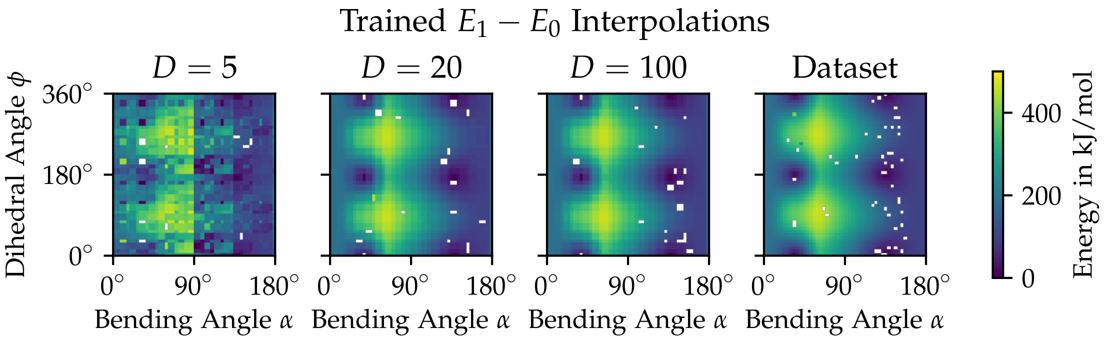


Figure 4.10: A plot showing the trained QTCIs with different algorithmic parameters. Each column corresponds to an algorithmic configuration. We captured the interpolation at the end of each step and evaluated it on the same regular grid. The rightmost column contains the reference dataset.

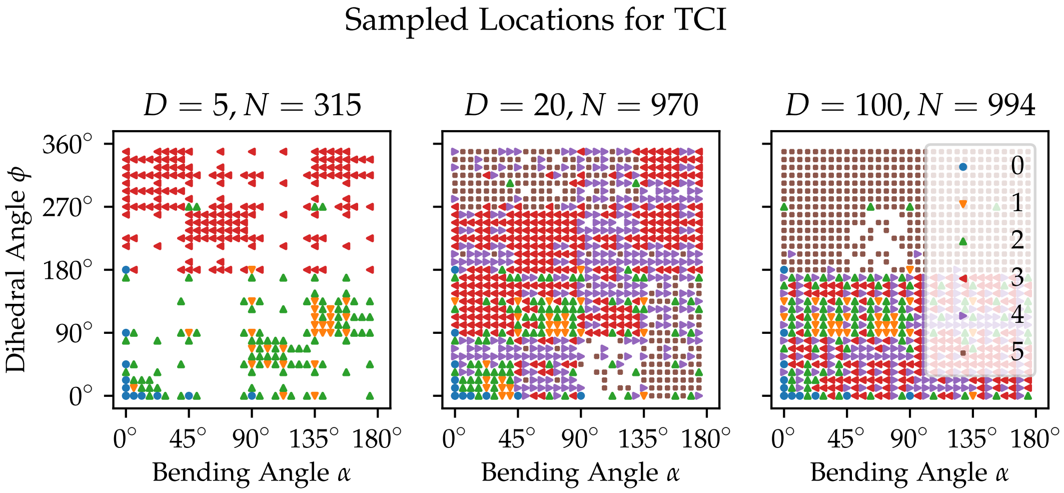


Figure 4.11: This plot shows the quantized locations where the TCI algorithm sampled the formaldimine PES. Different configurations took a different number of iteration steps to converge. The step is indicated by a different shape.

At the selected bit depth of five bits per dimension, the quantified space has a size of $2^5 \cdot 2^5 = 1024$ potential sampling locations. For the two-dimensional surface of formaldimine, we can plot all possible sampling locations in a grid. Figure 4.11 collects the sampled locations for several runs corresponding to different algorithmic configurations of QTCI. For better comparability, the configurations were the same as captured in Table 4.1 that have been used to obtain Figure 4.10.

Figure 4.11 shows that a larger bond dimension leads to a larger number of requested samples. We hypothesize that the reason for this behavior is that the higher bond length causes a bigger pivot error that is reduced in subsequent iterations. Nevertheless, one observation is that, unless the bond length is really small ($D = 5$), the QTCI requests samples from almost all potential sampling locations of the PES. Initially, we hoped for the QTCI to require only a few evaluations of the PES, even for larger bond dimensions. Unfortunately, this goal could not be achieved.

In the next section, we analyze the errors of different interpolations to find out if the error reduces substantially by increasing the bond dimension of the QTCI.

4.5.3 Error Analysis

Table 4.1 contains the errors of the interpolation on our small test dataset. Since the relative tolerance threshold of QTCI was specified in units of squared energy, namely Hartree^2, our errors are specified in Hartree^2 as well. The numbers are higher than we expected. Therefore, we investigated the individual contribution of each sample for the dataset. In Figure 4.12, we show a box plot for the various configurations reported in the table. We can see that many samples produce a low error. Nevertheless, there are many strong outliers that contribute to high error, visible as black circles above the boxes.

After a close inspection, we found that the error occurred in regions where there is a huge discrepancy between individual samples that are close to each other. This manifests the thesis that the interpolation picks up the noise of the underlying quantum chemistry method and gets stuck focusing on the error instead of accurately modeling the surface.

4.6 High-Dimensional Hydrogen Chain

One initial claim of using QTCI was that tensor network methods usually shine when applied to higher-dimensional systems while embarking on the exponential blow-up of the tensor size. We chose a hydrogen chain to serve as an example system for providing a tractable and scalable PES to study. Unlike formaldimine, the n-dimensional nature of our hydrogen chain parametrization makes it impossible to visualize the energy in a

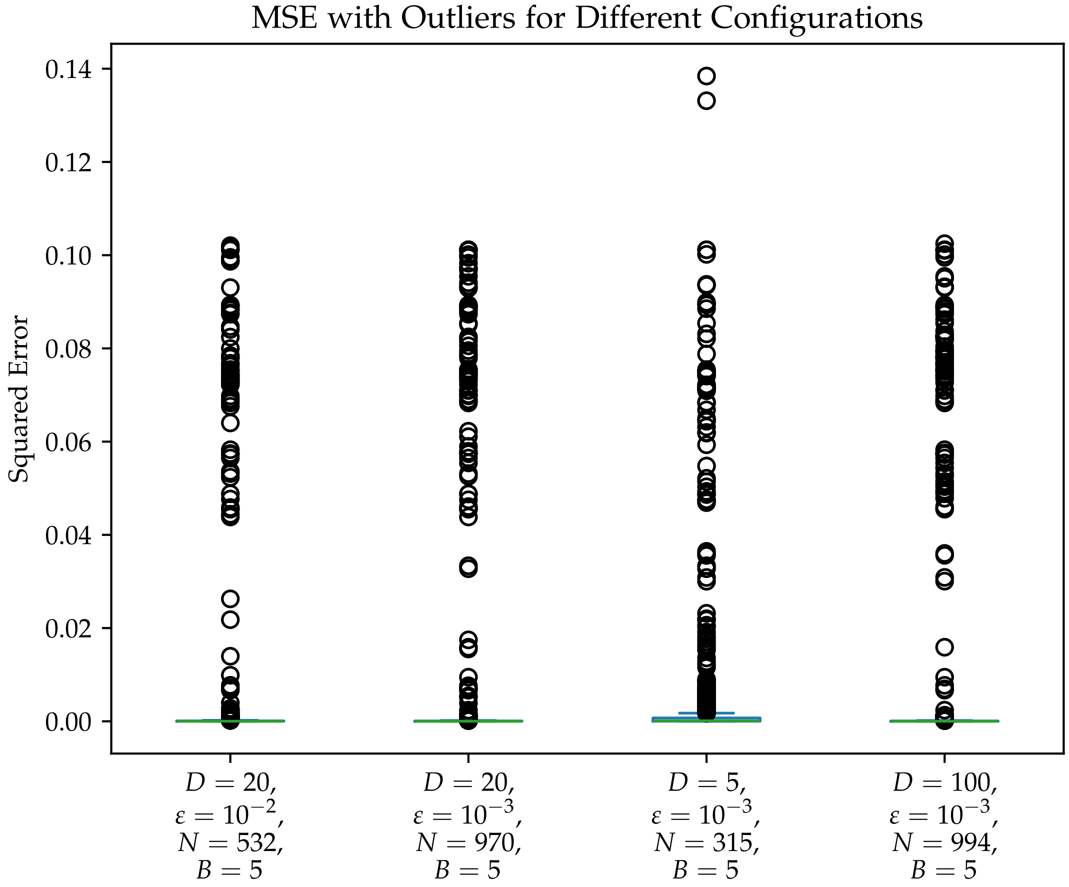


Figure 4.12: A plot showing the errors by different QTCIs on the dataset generated in Section 4.5.1. The values are sorted by the MSE reported in Table 4.1.

single plot for more than three dimensions. Nevertheless, we observed the absolute convergence of the method and subsequently compared it to a simple ML baseline. Because of the limitations of excited state computation, we focus only on representing the ground state PES in this part. Even though this problem is different from modeling the excited states, it still demonstrates the capabilities of modeling a PES with QTCI.

In the machine learning regime, it is common to fit a model to a predefined set of data. We simulated this scenario in order to compare the QTCI fitting approach to a traditional machine learning procedure. For different instances of the hydrogen chain, differing by the number of degrees of freedom, we trained a QTCI and a machine learning model. Both were then evaluated on the same independent test set.

As a machine learning model, we picked a perceptron with two hidden layers with 100 units each. Formally, the model can be expressed as a function mapping a set of coordinates $\boldsymbol{x}$ to an output, i.e., the energy value y. The computation can be expressed formally as

$$\begin{aligned} \boldsymbol{h}_0 &= \boldsymbol{x} \\ \boldsymbol{h}_i &= \mathrm{ReLU}(\boldsymbol{h}_{i-1} \cdot \boldsymbol{W}_i + \boldsymbol{b}_i) \qquad\qquad \text{for } i \in \{1,2\} \\ y &= \boldsymbol{h}_2 \cdot \boldsymbol{W}_3 + \boldsymbol{b}_3 \end{aligned}$$

where the $\mathrm{ReLU}(x) = \max\{0, x\}$ is a common nonlinear activation function applied elementwise. The goal of the optimization is to find a set of parameters $\theta = \{\boldsymbol{W}_i, \boldsymbol{b}_i\}_{i=1}^{3}$ such that the MSE on the validation dataset is minimized. This choice of model has around 10,000 parameters. For these experiments, we used `flax` [10] and the `optax` [3] library for performing gradient descent as described in Section 2.7.2.

For measuring the quality of the obtained representation, we used the MSE as derived in Section 2.7.1 since it can be applied to both the QTCI and the machine learning model. Two further metrics are of key importance for comparison: The number of samples used for training, and the number of parameters each model possesses.

Figure 4.13 contains a plot for each metric. The first one shows the MSE of each method on the same test set. We observe that the machine learning model's error curve diverges from the QTCI for larger systems. This is to be expected since both the model size and the training set size were held constant over all instances of the hydrogen chain. This means that we essentially trained the same model on different systems and the machine learning systems performed gradually worse for larger and larger systems. Using QTCI dampens this behavior and the error stays well below the error of the machine learning system.

We hoped the number of parameters could remain constant for larger hydrogen chains. Unfortunately, this was not possible, which can be seen in the explosion of the number of parameters required for larger chains in Figure 4.13. Simply put, if the

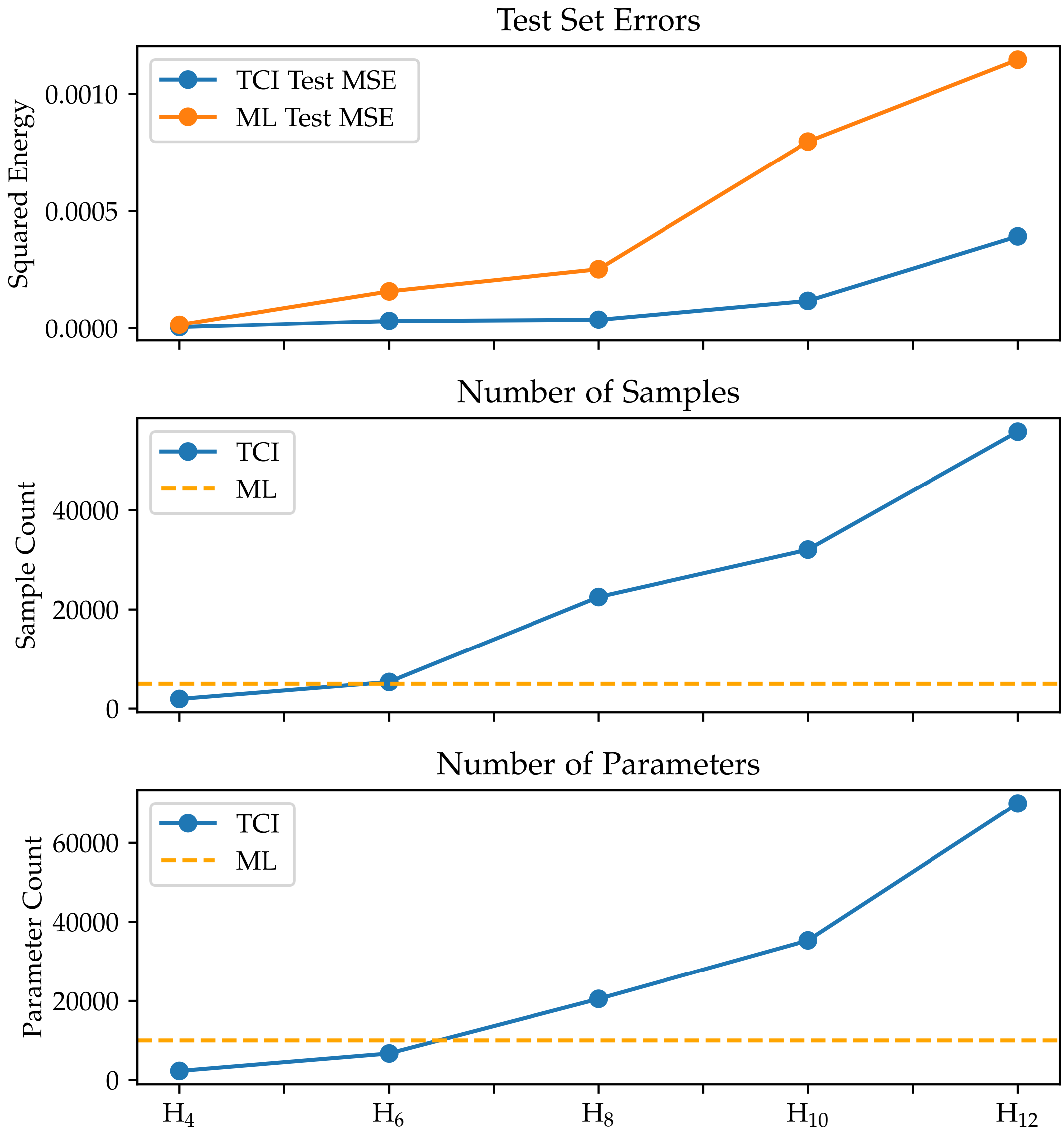


Figure 4.13: A plot comparing the trained QTCI with a simple machine learning model on different instances of the hydrogen chain. Each column corresponds to a different metric. Note that the number of samples and number of parameters is constant across all machine learning configurations.

Table 4.2: The different parameter of QTCI and ML.

Parameter	QTCI	ML
Dataset size	Variable	Fixed
Number of Parameters	Variable	Fixed
Convergence Accuracy	Fixed	Variable

system size rises, the number of parameters to represent the function rises drastically as well. Therefore, the initial claim that the QTCI can generalize the behavior learned on smaller systems to larger systems cannot withstand. The key differences between the machine learning model and the QTCI are summarized in Table 4.2.

4.7 Analysis and Findings

Paired with the scaling experiments in the last section, it is especially evident that the experimental setup initially falls short of the expected results. Even though QTCI does a reasonable job of representing the PES of the corresponding molecule, it is incapable of doing that in a parsimonious way that is sample-efficient. The experiments scaling the bond length especially revealed that the QTCI quickly gathers all the samples it could from the PES. If we had increased the bit depth, the QTCI would have gotten stuck investigating features that might be important for rapidly oscillating functions but that are not present for smooth PESs. Our experiments revealed that the chosen quantum chemistry tools were not stable enough for the interpolation process. The tools became overly specific, and an overzealous amount of time was spent on samples querying already known regions.

We found that machine learning has an easier job at learning from the noisy samples, especially when assuming a prior statistical model as derived in Section 2.7.1. With noisy input in mind, learning from the samples generated by the quantum chemistry methods prevents a representation from trying to get overly confident where it is not possible. An interpolation defined like QTCI does not come with such a necessary error tolerance and, therefore, falls short of the expected results.

5 Conclusion

Conical intersections are an immensely relevant topic in quantum chemistry. Although key in many important processes, finding conical intersections using PES from first principles remains vastly unsolved. We aim to provide insights into this problem by combining an interpolation technique with quantum chemistry solvers. Contrary to existing approaches in which the PES is usually sampled from a uniform grid, our approach should behave like a grid without the need for requiring an exponential amount of samples. The overhead is reduced by only sparsely sampling the PES while inferring the structure of the underlying space. We propose directly modeling the difference in the excited states to pinpoint the location of a conical intersection.

In this thesis, we present the findings of our exploratory project. First, we experimented with various quantum chemistry tools in order to find the most stable version. For our experiments, we used TD-DFT and DMRG-SCF. Despite not being very stable, the former is fast enough to allow for rapid prototyping. Unfortunately, the unstable behavior does not allow for deep investigations, which is why we switched to DMRG-SCF for further experiments. We applied our approach to several molecules, including small dimers and a model for a photochemical reaction. We find that the QTCI can model the PES, but it cannot represent the PES only from a few samples. The interpolation algorithm is not able to ignore the method-inherent noise of the underlying quantum chemistry tools.

Compared to traditional machine learning, the QTCI requires a number of samples that increases with system size. This is in contradiction to the expected benefits of tensor decompositions using MPSs. Even though QTCI has been reported to shine on various tasks like integration, we could not transfer this spirit to PES. We believe that the main limiting factor is the noisy nature of the quantum chemistry tools.

While the current study revealed challenges in approximating PES, the potential for improvement remains vast. Future research could include the development and use of more stable quantum chemistry methods. The ultimate goal is to create stable, accurate, and computationally efficient models for the PES, which would significantly advance our understanding of chemical photoreactions and facilitate progress in various applications, including the design and use of novel materials and catalysts.

Bibliography

[1] V. Bonačić-Kouteckỳ and J. Michl. Photochemical syn-anti isomerization of a schiff base: A two-dimensional description of a conical intersection in formaldimine. *Theoretica chimica acta*, 68:45–55, 1985.

[2] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[3] DeepMind, I. Babuschkin, K. Baumli, A. Bell, S. Bhupatiraju, J. Bruce, P. Buchlovsky, D. Budden, T. Cai, A. Clark, I. Danihelka, A. Dedieu, C. Fantacci, J. Godwin, C. Jones, R. Hemsley, T. Hennigan, M. Hessel, S. Hou, S. Kapturowski, T. Keck, I. Kemaev, M. King, M. Kunesch, L. Martens, H. Merzic, V. Mikulik, T. Norman, G. Papamakarios, J. Quan, R. Ring, F. Ruiz, A. Sanchez, L. Sartran, R. Schneider, E. Sezener, S. Spencer, S. Srinivasan, M. Stanojević, W. Stokowiec, L. Wang, G. Zhou, and F. Viola. The DeepMind JAX Ecosystem, 2020.

[4] S. Dolgov and D. Savostyanov. Parallel cross interpolation for high-precision calculation of high-dimensional integrals. *Computer Physics Communications*, 246:106869, 2020.

[5] A. Dreuw and M. Head-Gordon. Single-reference ab initio methods for the calculation of excited states of large molecules. *Chemical Reviews*, 105(11):4009–4037, 2005. PMID: 16277369.

[6] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[7] Y. N. Fernández, M. K. Ritter, M. Jeannin, J.-W. Li, T. Kloss, T. Louvet, S. Terasaki, O. Parcollet, J. von Delft, H. Shinaoka, and X. Waintal. Learning tensor networks with tensor cross interpolation: new algorithms and libraries, 2024.

[8] P. M. W. Gill. Density functional theory (dft), hartree-fock (hf), and the self-consistent field. *J. Chem. Phys*, 100:5066–5075, 1994.

[9] S. Gozem, F. Melaccio, A. Valentini, M. Filatov, M. Huix-Rotllant, N. Ferre, L. M.

Frutos, C. Angeli, A. I. Krylov, A. A. Granovsky, et al. Shape of multireference, equation-of-motion coupled-cluster, and density functional theory potential energy surfaces at a conical intersection. *Journal of chemical theory and computation*, 10(8):3074–3084, 2014.

[10] J. Heek, A. Levskaya, A. Oliver, M. Ritter, B. Rondepierre, A. Steiner, and M. van Zee. Flax: A neural network library and ecosystem for JAX, 2024.

[11] J. Ischtwan and M. A. Collins. Molecular potential energy surfaces by interpolation. *The Journal of Chemical Physics*, 100(11):8080–8088, 06 1994.

[12] F. Jensen. *Introduction to Computational Chemistry*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2006.

[13] K. P. Murphy. *Machine learning: a probabilistic perspective*. 2012.

[14] Y. Núñez Fernández, M. Jeannin, P. T. Dumitrescu, T. Kloss, J. Kaye, O. Parcollet, and X. Waintal. Learning feynman diagrams with tensor trains. *Physical Review X*, 12(4), Nov. 2022.

[15] R. Penrose. Applications of negative dimensional tensors. *Combinatorial mathematics and its applications*, 1:221–244, 1971.

[16] D. Pfau, J. S. Spencer, A. G. D. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2(3), Sept. 2020.

[17] M. K. Ritter, Y. Núñez Fernández, M. Wallerberger, J. von Delft, H. Shinaoka, and X. Waintal. Quantics tensor cross interpolation for high-resolution parsimonious representations of multivariate functions. *Physical Review Letters*, 132(5), Jan. 2024.

[18] M. Scherbela, R. Reisenhofer, L. Gerard, P. Marquetand, and P. Grohs. Solving the electronic schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks, 2021.

[19] U. Schollwöck. The density-matrix renormalization group. *Reviews of modern physics*, 77(1):259–315, 2005.

[20] Q. Sun, X. Zhang, S. Banerjee, P. Bao, M. Barbry, N. S. Blunt, N. A. Bogdanov, G. H. Booth, J. Chen, Z.-H. Cui, J. J. Eriksen, Y. Gao, S. Guo, J. Hermann, M. R. Hermes, K. Koh, P. Koval, S. Lehtola, Z. Li, J. Liu, N. Mardirossian, J. D. McClain, M. Motta, B. Mussard, H. Q. Pham, A. Pulkin, W. Purwanto, P. J. Robinson, E. Ronca, E. R. Sayfutyarova, M. Scheurer, H. F. Schurkus, J. E. T. Smith, C. Sun,

S.-N. Sun, S. Upadhyay, L. K. Wagner, X. Wang, A. White, J. D. Whitfield, M. J. Williamson, S. Wouters, J. Yang, J. M. Yu, T. Zhu, T. C. Berkelbach, S. Sharma, A. Y. Sokolov, and G. K.-L. Chan. Recent developments in the pyscf program package. *The Journal of Chemical Physics*, 153(2):024109, 07 2020.

[21] A. Szabo and N. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Books on Chemistry. Dover Publications, 1996.