Claremont Colleges
Scholarship @ Claremont

CGU Theses & Dissertations

CGU Student Scholarship

Fall 2022

Data-Driven Methods for Low-Energy Nuclear Theory

Jordan M.R. Fox
Claremont Graduate University

Follow this and additional works at: https://scholarship.claremont.edu/cgu_etd

Part of the Nuclear Commons

Recommended Citation
Fox, Jordan M.R.. (2022). *Data-Driven Methods for Low-Energy Nuclear Theory*. CGU Theses & Dissertations, 445. https://scholarship.claremont.edu/cgu_etd/445.

This Open Access Dissertation is brought to you for free and open access by the CGU Student Scholarship at Scholarship @ Claremont. It has been accepted for inclusion in CGU Theses & Dissertations by an authorized administrator of Scholarship @ Claremont. For more information, please contact scholarship@cuc.claremont.edu.

DATA-DRIVEN METHODS FOR LOW-ENERGY NUCLEAR THEORY

by

Jordan M. R. Fox

Claremont Graduate University and San Diego State University

Fall 2022

Copyright © 2022

by

Jordan M. R. Fox

Approval of the Dissertation Committee

This dissertation has been duly read, reviewed, and critiqued by the Committee listed below, which hereby approves the manuscript of Jordan Fox as fulfilling the scope and quality requirements for meriting the degree of Doctor of Philosophy in Computational Science.

Calvin W. Johnson, Chair
San Diego State University
Professor of Physics

Kenneth Nollet
San Diego State University
Associate Professor of Physics

Rodrigo Navarro Perez
San Diego State University
Assistant Professor of Physics

Marina Chugunova
Claremont Graduate University
Professor of Mathematics

Ali Nadim
Claremont Graduate University
Professor of Mathematics

ABSTRACT

Data-driven methods for
low-energy nuclear theory
by
Jordan M. R. Fox

San Diego State University and Claremont Graduate University: 2022

The term *data-driven* describes computational methods for numerical problem solving which have been developed by the field of data science; these are at the intersection of computer science, mathematics, and statistics. When applied to a domain science like nuclear physics, especially with the goal of deepening scientific insight, data-driven methods form a core pillar of the computational science endeavor. In this dissertation I explore two problems related to theoretical nuclear physics: one in the framework of numerical statistics, and the other in the framework of machine learning.

I) Historically our understanding of the structure of the atomic nucleus, the quantum many-body problem, has been built upon many layers of approximation, since the computational complexity of the problems is so large. One of the most flexible and enduring models, the configuration-interaction shell model, allows for detailed calculations of arbitrary scope. I lay out a simple framework for uncertainty quantification in empirical shell model calculations, thus providing not only error bars for large-scale calculations, but also insight for theory optimization and experimental design.

II) Nuclear cross sections are an integral component in many different applications including astrophysics and nuclear medicine, but descriptions of cross sections are often very "data-heavy". Huge libraries consisting of cross section evaluations, a combination of experimental measurements and theoretical results, are dense with information and thus ripe for data-driven methods. I have developed a deep learning system to learn trends in cross sections across the nuclear landscape. This system can predict cross sections for any nuclide and also can be used as an ensemble predictor. This is to my knowledge the first generative adversarial model developed for analyzing trends in nuclear data libraries.

DEDICATION

To Dad and Suzy.

ACKNOWLEDGMENTS

My utmost gratitude and respect goes to Calvin Johnson for so thoughtfully guiding me for the last six years. Through his care and tutelage I have grown from a student into a young scientist, confident in my own abilities and excited to continue research by my own direction. Over these years I have learned from him not only science and mathematics, but though his example how to meet adversity with tact, sincerity, and respect for every person.

Thanks to my committee members for their support. To Rodrigo Navarro-Perez for always taking the time to listen to my difficulties and point me precisely in the right direction. To Ken Nollett, whose catalog of contributions to nuclear theory and astrophysics I respect immensely. To Ali Nadim, the most precise and clearest lecturer I have had the pleasure of learning from. To Marina Chugunova, for helping us simple "engineers" through some of the most challenging mathematics I have encountered, and helping me to overcome my imposter syndrome. I am thankful to be supported by such an excellent group.

Thanks to the Computational Science Research Center, our director Jose Castillo, Satchi Venkataraman, and Parisa Plant. Thanks to Charlotte Ballesteros at CGU.

Thanks to my lab mentor Kyle Wendt for his guidance and trust over these years. Thanks to the nuclear theory group at Lawrence Livermore National Laboratory for their accommodation during my internships.

Thanks to my friends in the SDSU computational nuclear physics research group, in particular Ryan Zbikowski, Stephanie Lauber, and Oliver Gorton for their camaraderie and support over the years.

Lastly thanks to my family for their unwavering support and for encouraging me to follow my passions.

If people never did silly things, nothing intelligent would ever get done.

– Ludwig Wittgenstein

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

0.1 OVERVIEW

In this dissertation I explain two research problems I have worked on during my doctoral research.

Chapters 1, 2, and 3 introduce the relevant aspects of nuclear theory, uncertainty quantification, and machine learning respectively. It is my intent that these chapters give sufficient background and framing to the later chapters which detail my work.

Chapters 4 and 5 cover my work on uncertainty quantification of nuclear shell model calculations. Much of the contents from these chapters are taken from two papers [42] and [43]. At the time of writing, the first paper has been published in Physical Review C, and the second is in peer review for the same journal.

Chapter 6 covers my work on a deep learning application for nuclear reaction data. This research began during an internship at Lawrence Livermore National Laboratory where I was working with Kyle Wendt, and continued into a USDOE Office of Science Graduate Fellowship in 2021 and after. At the time of writing we are preparing a paper on this research for publication[1]

Finally, Chapter 7 summarizes the conclusions of each project and describes my plans for future research.

---

[1] I cannot provide a citation for this work but a manuscript will be posted to arXiv.org in the weeks following publication of this dissertation.

CHAPTER 1

INTRODUCTION TO LOW-ENERGY NUCLEAR THEORY

1.1 Foundations

The physics of atomic nuclei informs some of our deepest scientific questions, like the origins of visible matter, the physics of stars, and the cosmological dark matter problem. More urgently though, some science problems are crucial for the security and future of humanity: safe energy generation, radioisotope use in medicine, and nonproliferation of nuclear weapons will shape our world through the 21st century and beyond. Today's nuclear physicists have a responsibility to build a robust understanding of our science. We are presented with a unique challenge: while the Standard Model of particle physics is the most precise physics model ever developed, there are still crucial gaps in our understanding. Many promising avenues are being pursued in concert, for example robust *ab initio* methods constructed from fundamental physics, and many-body simulation on quantum computers. My work takes another approach, so-called **data-driven methods**, which extract insight from the abundance of experimental and theoretical data using statistical methods and machine learning.

An atomic nucleus is a bound state of Z positively charged protons and N uncharged neutrons, which together as constituent particles we call **nucleons**. The pair of numbers (Z, N) defines a **nuclide** (a.k.a. species), and each nuclide exhibits particular properties. For example, Carbon-13 is a nuclide with $(Z, N) = (6, 7)$, and it has a particular set of excitations it may exhibit, with different attributes like angular momentum and radius. Understanding these low-energy excitations and their properties is referred to as the study of nuclear structure.

The **chart of nuclides**, shown in Fig. 1.1, shows the nuclides arranged on a grid with neutron number N on the horizontal axis and proton number Z on the vertical. The diagonal trajectory is dubbed the **valley of stability**: nuclides at the center of the valley are stable, meaning they will never decay spontaneously, and most surrounding nuclides are unstable. At present there are about 2900 known nuclides [10], the majority of which are unstable. Unstable nuclei have several ways they can decay (beta, alpha, etc.), and in most cases this results in a different nuclide. Nuclides are considered nonexistent if the nucleons do not form a bound state, and on the chart this occurs at the **driplines**: the edges of the valley of stability. At the dripline, adding another nucleon to the nucleus does not resulting in a bound system, and the new particle spontaneously "drips" off.

One of the fundamental questions in nuclear theory is the nature of the nuclear forces [84]. If nucleons only felt the electromagnetic force, nuclei would fly apart due to the repulsive nature of like charges. Since nuclei are bound, however, we infer there is an overall attractive force felt between nucleons, and that it must be stronger than the electromagnetic force at least on scales of 1 fm = 10^{-15} meters (a.k.a. a distance of one *fermi*). Unlike the forces of electromagnetism and gravity, which do work at any length scale thanks to massless force-carrying particles, the nuclear forces are relegated to *only* small scales. Experimental avenues for probing low-energy structure can be complicated and expensive, and in those cases theoretical guidance of experimental design is valuable. Comprehensive UQ for nuclear theory can help inform experimental design for the nuclear forces, but also for neutrinos and physics beyond the Standard Model.

Nuclei are many-body quantum mechanical systems, so not all variables can be measured simultaneously and to arbitrary accuracy. In the theory, this is represented by non-commutativity of operators: measuring position first and momentum second is famously *not* the same as measuring them the other way around. Model validation for nuclear theory quite often comes down to comparing a handful of numbers: binding energies, low-lying excitation energies, decay half-lives, etc. Unlike classical physics the number of simultaneously observable variables is restricted, and of that set even fewer serve as truly practical and informative probes[1].

Nuclear structure refers to properties of an individual nucleus at low energy: spatial density, angular momentum, collective motion of nucleons, etc. Nuclear structure is thus dominated by the nature of the strong nuclear force and quantum mechanics itself; in a sense we are excluding more complicated situations (i.e. relativistic) and just focus on the bare necessities of a nucleus at low energy. The first chapter of Walecka's book [128] gives a brisk review of the strong force; roughly summarized, the strong force is: attractive, short-range (~2 fm), spin-dependent (and *spin-orbit* dependent, $\hat{s} \cdot \hat{\ell}$), non-central (not only s-wave), charge independent, and hard-core (strongly repulsive at small distance $\lesssim 1/2$ fm). Nucleons are fermions, which means they are subject to the Pauli exclusion principle: no two particles can occupy the same state. Many complicated phenomena result from the combination of these attributes. A complete theory of structure must explain excitation spectra, spin properties, vibrational/rotational modes, transition properties (electromagnetic and electroweak decays), and other properties like electromagnetic moments, including charge radius, etc. The nuclear structure problem can only manifest naturally in a finite number of ways: of the 2900 known nuclides, only a subset of those are possible to obtain and study experimentally.

The modern theoretical formulation of nuclear structure is the quantum many-body problem; we construct a model of the nuclear force with the above properties satisfied and then calculate solutions for which the total energy of the system is well-defined. These solutions can then be used to further compute properties of the nucleus, interactions of the nucleus with external phenomena, etc.

---

[1] A good example of a useful observable is the magnetic dipole moment, which happens to have a wonderfully simple operator at leading order: a sum of spin and orbital angular momentum. We can use this to probe the spin degrees of freedom alone and thus it is relatively easy to interpret.

1.2 The many-body Schrödinger equation for nuclei

In order to describe the many-body quantum state of the nucleus, the problem we want to solve is the time-independent Schrödinger equation in eigenvalue form:

$$\hat{H}|\Psi\rangle = E|\Psi\rangle, \tag{1.1}$$

where $\hat{H}$ is the Hamiltonian, $|\Psi\rangle$ is the nuclear (many-body) wavefunction, and E is the energy of the nuclear wavefunction. I use hats to denote operators, and usually the same letter without the hat is the matrix representation, that operator in a discrete basis, unless otherwise noted. Once we have the nuclear wavefunctions $|\Psi\rangle$ in hand we can compute observables like attributes of the nucleus (radius, densities, etc.) and decay probabilities.

The familiar Hamiltonian form is $\hat{H} = \hat{T} + \hat{V}$. For multiple particles of equal mass, we have

$$\hat{H} = \sum_{i=1}^{N} -\frac{\hbar^2}{2m}\nabla_i^2 + U(\vec{r}) + \sum_{i<j} V(\vec{r}_i - \vec{r}_j), \tag{1.2}$$

where U is an external field which is necessary for certain problems (e.g. a Coulomb field, or a mean-field approximation of forces), and V is a two-body interaction. Now, I will present a reformulation of this Hamiltonian, in so-called *second quantization*, specifically for fermions (which protons and neutrons are). For the whole story on second quantization one may turn to most textbooks on field theory; Chapter 3 of Klauber's QFT book [74] has a good pedagogical presentation. Our Hamiltonian becomes

$$\hat{H} = \sum_{rs} T_{rs}\hat{a}_r^\dagger \hat{a}_s + \frac{1}{4}\sum_{rstu} V_{rstu}\hat{a}_r^\dagger \hat{a}_s^\dagger \hat{a}_u \hat{a}_t = \sum_i \lambda_i \hat{o}_i \tag{1.3}$$

where $\hat{a}^\dagger, \hat{a}$ are the fermion creation and annihilation operators respectively ($\{\hat{a}_i^\dagger, \hat{a}_j\} = \hat{a}_i^\dagger \hat{a}_j + \hat{a}_j \hat{a}_i^\dagger = \delta_{ij}$). They act on the many-body wavefunction and add or remove particles in the state denoted by their subscripts. This is very handy because any many-fermion state can be represented as a product of creation operators acting on the vacuum state: $\prod_i \hat{a}_i^\dagger |0\rangle$. The result will be properly antisymmetrized, and we can fully describe the state just by the indices of the operators; this is known as the **occupation representation**. The coefficients T, V are real numbers, and the letters here are chosen because these terms play roughly the same roles as kinetic energy and potential did before. As input to nuclear configuration-interaction codes [22, 21, 24, 123], the two-body matrix elements are always coupled up to an angular momentum scalar so that the many-body angular momentum J is a good quantum number of eigenstates. (To be specific, the two-body matrix elements are $V_{JT}(ab, cd) = \langle ab; JT|\hat{V}|cd; JT\rangle$, where $\hat{V}$ is the nuclear two-body interaction and $|ab; JT\rangle$ is a normalized two-body state with nucleons in single-particle orbits labeled by a, b coupled up to total angular momentum J and total isospin T.) The right-most expression in Eq. 1.3 is simply pointing out that the Hamiltonian is just a linear combination of density operators $\hat{o}$. This is a handy representation when doing parameter estimation, such as fitting the interaction to reproduce some experimental observation, because we can deal directly with the vector $\boldsymbol{\lambda}$.

The term **density operator** is used for operators of the form $\hat{a}_i^\dagger \hat{a}_j$ (one-body) , $\hat{a}_r^\dagger \hat{a}_s^\dagger \hat{a}_u \hat{a}_t$ (two-body), $\hat{a}_q^\dagger \hat{a}_r^\dagger \hat{a}_s^\dagger \hat{a}_v \hat{a}_u \hat{a}_t$ (three-body), and so on. The expectation value of $\hat{a}_i^\dagger \hat{a}_j$, for instance, measures the density of the wavefunction from state j and i. Likewise $\hat{a}_r^\dagger \hat{a}_s^\dagger \hat{a}_u \hat{a}_t$ measures the density of the wavefunction from the pair of states t, u to the pair r, s. Building the Hamiltonian from density operators conveniently allows us to represent many different theoretical force models in a single format.

Importantly, in Eq. 1.3, the sums run over single-particle *states*, not over the particles themselves, because the particles are indistinguishable. We will decide on what single-particle states to include in the model, and that is separate from the total number of constituent particles we are modeling. We must make two decisions now: first, what are the single-particle states? and second, what are the coefficients $\boldsymbol{\lambda}$?

1.3 The nuclear shell model

The nuclear shell model (NSM) is fundamentally based on two empirical observations: the Pauli exclusion principle and the magic number phenomena. Regarding the former, particles with half-integer spin ($^1/_2, ^3/_2$, etc.) obey the Pauli exclusion principal, meaning no two particles in a many-body system can occupy the same state. These systems are called **fermionic** and they include the electrons around an atom and the protons and neutrons in a nucleus. Many-body systems tend to prefer low-energy states, but the Pauli principle means fermions will "stack" on top of one another, filling up states like blocks in a game of *Tetris*. **Bosonic** systems by contrast are not subject to exclusion and thus are free to fit many particles into the same state; a famous example of which is the superfluidity of helium-4. Regarding magic number phenomena, nuclides with certain magic numbers (2, 8, 20, 28, 50, etc.) of constituent nucleons are observed to have a greater binding energy (i.e. are bound tighter) than others similar to them. Although binding energy per nucleon is almost constant for the majority of nuclides, as shown in Fig. 1.2, shell structure is clearly visible when looking at the whole chart: Fig. 1.3 shows approximate second derivatives in binding energy with respect to N, Z, so deviations away from linearity show up very clearly and are generally located at these magic numbers. One powerful aspect of the shell model is the ability to set up calculations *between* shell boundaries, thereby alleviating the need to reproduce complex shell structure, and instead allowing us to focus on modeling trends in between. As such, so-called empirical shell model Hamiltonians are designed for specific closed regions of model space.

Prior to the development of the NSM, a similar magic number phenomena had also been observed in electron structure of atoms, although with different magic numbers (2, 10, 18, 36, 54, etc.). Much of the theory used in atomic structure was carried over to nuclear structure; of course the constituent particles and governing forces are very different, but some theoretical constructions persist. Namely, N-body basis states are almost always considered to be products of N single-particle wavefunctions, thus solutions to the problem are a linear combination of those. The theoretical approach to single-particle wavefunctions in NSM adopts some conventions from the electron shell model, including the use of **spectroscopic notation**. Oftentimes out individual particles inhabit eigenstates of the 3D harmonic oscillator labelled by quantum numbers: the nodal number n, the orbital number l, spin s (almost always omitted, since $|s| = 1/2$ for all fermions), total angular momentum $j = l + s$ (where $s = \pm 1/2$), and the magnetic number $m = j_z$. Spectroscopic notation is simply a shorthand for describing the **orbit** of a particle: that is, the state(s) specified by n, l, j : typically written nl_j. The magnetic number is omitted because in general our systems are energy-degenerate in m, meaning the Hamiltoninan does not depend on m. To sufficiently confuse the uninitiated, each l is assigned a letter corresponding to an arcane naming convention: $l = 0$ is assigned s (sharp), $l = 1$ gets p (principal), $l = 2$ gets d (diffuse), $l = 3$ gets f (fundamental/fine structure), and then the naming continues alphabetically with g, h, etc. So, our single-particle orbits beginning with $n = 0$ are written as $0s_{1/2}, 0p_{3/2}, 0p_{1/2}, 1s_{1/2}, 0d_{5/2}, 0d_{3/2}$, and so on. Model spaces in the NSM are described as set of orbits: the "p-shell" is the set $\{0p_{3/2}, 0p_{1/2}\}$, the sd-shell is the set $\{1s_{1/2}, 0d_{5/2}, 0d_{3/2}\}$, and so on.

Within a usual shell model calculation, we begin with the model inputs and, with the help of computer code, end with nuclear wavefunctions. From nuclear wavefuctions we can compute observables and compare to experiment (or study them in their own right, though technically they are not observables). The model inputs include the one-body wavefunctions from which we form the many-body basis, the nuclear force expressed in that basis, and proton number and neutron number to specify the nuclide.

1.3.1 Interactions

Modern nuclear interactions typically fall into two categories: phenomenological and *ab initio*. My work mainly concerns the former, but I include a mention of the latter because they are very important for modern shell model calculations. The difference of the two has to do with parameterizations of the nuclear force: phenomenological forces may be parameterized to fit some data and may not have an easy physical interpretation, whereas *ab initio* ones are tuned by physically "low-level" parameters, 2-nucleon and 3-nucleon scattering data, and thus may be more robust but generally require more work to afford the same accuracy.

A **phenomenological** or **empirical interaction** is a nuclear Hamiltonian which is constructed for a specific model space, typically a small number of contiguous shell model orbitals. The interactions relevant to my research all have the form of 1.3 but in general they do not need to; we say this Hamiltonian is a sum over one- and two-body density operators. The model assumes that some of the nucleons are inactive, or *frozen*, and those within the model space interact and form the different configurations. Here are some examples of model spaces.

- p-shell calculations. We assume the core is a frozen ^{4}He nucleus, and there are $(N-2)$ neutrons and $(Z-2)$ protons forming configurations in the $0p_{3/2}+0p_{1/2}$ space (up to 6 neutrons and 6 protons active).
- sd-shell calculations. We assume the core is a frozen ^{16}O nucleus, and there are $(N-8)$ neutrons and $(Z-8)$ protons forming configurations in the $0d_{5/2}+0d_{3/2}+1s_{1/2}$ space (up to 12 neutrons and 12 protons active).
- pf-shell calculations. We assume the core is a frozen ^{40}Ca nucleus, and there are $(N-20)$ neutrons and $(Z-20)$ protons forming configurations in the $1p_{1/2}+1p_{3/2}+0f_{5/2}+0f_{7/2}$ space (up to 20 neutrons and 20 protons active).

Using a Hamiltonian made of one- and two-body density operators, choosing the model space fully determines the number of parameters: each orbit is assigned a **single-particle energy** and every four orbits (which do not break rotational symmetry or isospin) is assigned a **two-body matrix element**. Together these form our interaction parameters $\boldsymbol{\lambda}$. These parameters can be determined from first principles [79], but the usual procedure is to begin with the so-called Brueckner G-matrix method [61]. The G-matrix has historically been an indispensable component of nuclear structure calculations, since it is constructed by wrapping many different orders of interaction into a single matrix. The G-matrix provides a good starting point for parameter values; typically one then iteratively optimizes $\boldsymbol{\lambda}$ to fit a set of experimental energies and occasionally some other structure observables.

The phenomenological interaction used in my work is the universal *sd*-shell interaction, version B; **USDB** for short [18]. It is one of several interactions made for the *sd*-shell by B. Alex Brown (Michigan State University) and collaborators.

Finally, ***ab initio*** **interactions** are part of a larger effort by nuclear theorists to create computationally tractable descriptions of the nucleus without some of large-scale approximations typical traditional shell-model calculations. *Ab initio*, meaning "from first principles", is generally a label for models which construct the system as an ensemble of small well-understood pieces rather than empirically fitting within a fixed energy regime. In nuclear physics this takes several forms but my experience is with the no-core shell model (NCSM) [97, 9]. The NCSM Hamiltonians are derived primarily from two- and few-nucleon data, often in the framework of chiral effective field theory [127]; because they work in large single-particle spaces and converge as the model space increases, such calculations are considered more robust than empirical calculations, but are limited to *p*-shell nuclides and the lightest *sd*-shell nuclides. Part of the motivation for such calculations is that one can identify the errors in the theory from leaving out higher order terms [131, 49, 23, 100, 87, 133], but in practice quantifying uncertainties in *ab initio* many-body calculations is far from trivial. As the name suggests, the NCSM does not assume a frozen core of nucleons, but rather all nucleons are considered as degrees of freedom. This means that even the very lowest-energy single-particle states must be included in the basis calculation, and ultimately the basis dimension is much larger in NCSM than the same nuclide in an empirical calculation. (A promising alternative to empirical interactions are so-called *microscopic* effective interactions [121] wherein *ab initio* methods are used in a fixed model space, but these are still very much under development.)

1.4 Shell model calculations with `BIGSTICK`

Our research group at San Diego State University uses the `BIGSTICK` code [68] for many different research applications[2]. `BIGSTICK` is a flexible configuration-interaction open-source shell-model code for the many-fermion problem. The primary operation of `BIGSTICK` is to solve the time-independent Schrödinger equation for a nuclear Hamiltonian, but has many other capabilities as well. I suggest the technically interested reader to look at [68] for a thorough overview. **Configuration-interaction** (CI) is a method for numerically solving the Schrödinger equation for a many-fermion Hamiltonian in an orthonormal basis of anti-symmetrized products of single particle states, called Slater determinants. The Hamiltonian matrix must be computed in this basis, then a diagonalization algorithm is used to obtain eigenvalues and eigenvectors (most often a subset of the total). CI is exact in principle; that is, there is nothing within the theoretical formulation of CI that would introduce errors. However in practice, computational resources are finite and so one must use finite bases to represent the problem; this inevitably leaves out some physical contributions and introduces errors. As computing systems have become more advanced, larger CI problems can be constructed and solved with decreasing errors.

To get the matrix version of Eq. 1.1, we strategically insert some completeness relations and simplify. Consider a finite orthonormal basis $\{|\Phi_i\rangle\}_{i=1,\ldots,M}$;

$$\sum_i |\Phi_i\rangle\langle\Phi_i|\hat{H}\sum_j |\Phi_j\rangle\langle\Phi_j|\Psi\rangle = E\sum_i |\Phi_i\rangle\langle\Phi_i|\Psi\rangle \tag{1.4}$$

and with some rearranging we can obtain

---

[2] The code is available for public download here.

$$\sum_{ij} \langle \Phi_i | \hat{H} | \Phi_j \rangle \langle \Phi_j | \Psi \rangle | \Phi_i \rangle = E \sum_i \langle \Phi_i | \Psi \rangle | \Phi_i \rangle. \tag{1.5}$$

The expression $\langle \Phi_i | \hat{H} | \Phi_j \rangle$ is the matrix element of the Hamiltonian, H_{ij}, and the inner products $\langle \Phi_i | \Psi \rangle$ and $\langle \Phi_j | \Psi \rangle$ are coefficients of the vector Ψ in this basis, c_i and c_j respectively. The eigenvalue problem can now expressed in terms of sums and products,

$$\sum_{ij} H_{ij} c_j | \Phi_i \rangle = E \sum_i c_i | \Phi_i \rangle \tag{1.6}$$

where

$$H_{ij} = \langle \Phi_i | \hat{H} | \Phi_j \rangle; \tag{1.7}$$

or, in typical algebra notation,

$$H \boldsymbol{\psi} = E \boldsymbol{\psi} \tag{1.8}$$

where $\psi_i = \langle \Phi_i | \Psi \rangle$ is the wavefunction represented as a vector in this basis. Now, Eq. 1.8 looks an awful lot like Eq. 1.1, but what we have constructed is actually much simpler. We have expressed the Hamiltonian operator and the formally "representationless" solution $|\Psi\rangle$ in a basis $\{|\Phi_i\rangle\}$. The many-body Schrodinger equation is now a typical eigenvalue problem – nothing more mysterious than what is seen in a first-year linear algebra course. However, computing the Hamiltonian matrix elements and solving this equation for very large dimensions is far from trivial.

Since we have complete control over our choice of basis, we should take the opportunity to bake in some physics. The basis used in `BIGSTICK`, for a calculation involving N particles, is a set of antisymmetrized products of N single particle states. Although not mandatory, it is convenient to choose single-particle states with nice symmetries; we almost always use 3D harmonic oscillator eigenstates defined by quantum numbers n (nodal), l (orbital), j (total spin), m (magnetic/orientation). Usually in CI, one chooses a number to be invariant for all basis states: in the

M-scheme we fix $J_z = M$, but there are other codes which fix J instead. These are called M-scheme and J-scheme, respectively, and both are equally valid, but there are practical tradeoffs. Taking M to be invariant means we set its value for all basis vectors and thus any solution will have $J \geq M$. This is convenient because experimental states are typically identified by the magnitude of J and excitation energy, so we can either set M to the minimum to include many states, or raise it to exclude states with $J < M$. Because nucleons have intrinsic spin $s = 1/2$, the minimum M is 0 for even A nuclides and 1/2 for odd A nuclides.

The single-particle space (SPS) is a set of one-body wavefunctions $|\phi\rangle$ from which the many-body basis is constructed. If all basis vectors $|\Phi\rangle$ are properly antisymmetrized many-fermion wavefunctions, then the solution to the many-body problem, $|\Psi\rangle$, will be a linear combination of properly antisymmetrized many-fermion wavefunctions, and thus antisymmetric itself. Conveniently, we can express this using a *determinant* structure: for N particles, we construct

$$\Phi(x_1, \ldots, x_N) = \frac{1}{\sqrt{N!}} \det \begin{bmatrix} \phi_1(x_1) & \phi_1(x_2) & \cdots & \phi_1(x_N) \\ \phi_2(x_1) & \phi_2(x_2) & \cdots & \phi_2(x_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N(x_1) & \phi_N(x_2) & \cdots & \phi_N(x_N) \end{bmatrix} \tag{1.9}$$

known as a **Slater determinant**. Here, the symbols ϕ_i mean the single particle wavefunction *occupied* by particle i. So, the matrix inside the determinant has dimension of the number of particles N; the whole basis is constructed by taking all possible combinations of single particle states for N particles, a.k.a. all possible *configurations* of the system.

In practice, our basis dimensions can far exceed what modern computers are capable of. For a matrix with dimension 10^8 and sparsity $1/10^6$, the number of nonzero matrix elements is about $(10^8)^2/10^6 = 10^{10}$. Even with some tricks (for instance, binary representation of Slater determinants) the number of nonzero matrix elements can quickly surpass hundreds of gigabytes. However, one of `BIGSTICK`'s features is a very efficient factorization of the Hamiltonian: we store not matrix elements themselves, but instead factors of them. This allows for reduced memory storage and good scalability on supercomputers. The M-scheme in general leads to large basis dimension, compared to other methods, but `BIGSTICK` remedies this by scaling efficiently to many hundreds or thousands of compute nodes (and across even more CPUs with multi-threading).

1.5 Electromagnetic and weak transitions

In this section I will first describe the theory of transition strengths, then explain the calculations we do to obtain them. The experimental property of interest for any transition is the half-life $T_{1/2}$, which is the amount of time for half of the sample to decay. The half-life is related to decay rate by

$$\frac{N_0}{2} = N_0 \exp(-W T_{1/2}) \rightarrow T_{1/2} = \frac{\ln 2}{W}. \tag{1.10}$$

The connection to nuclear theory is made by **Fermi's golden rule**,

$$W_{i\rightarrow f} = \frac{2\pi}{\hbar} |\langle \psi_f | \hat{O} | \psi_i \rangle|^2 \rho(E_f), \tag{1.11}$$

which describes how we compute the decay rate from energy eigenstates of a nucleus $|i\rangle, |f\rangle$. Here, $\rho(E_f)$ is the density of final states, a.k.a. the phase-space factor. The rule considers a single initial state decaying to multiple final states. Experimentalists however can measure rates between single initial and final states, so the phase-space factor is not relevant for us. We focus on the quantity $|\langle f|\hat{O}|i\rangle|^2$, which is called the **transition strength** ,

$$B(\hat{O})_{i\rightarrow f} = \frac{|\langle \psi_f : J_f || \hat{O} || \psi_i : J_i \rangle|^2}{2J_i + 1} \tag{1.12}$$

wherein the factor of $2J_i + 1$ is the number of orientations and accounts for isotropy of the strength. The double vertical lines around $\hat{O}$ indicate this is a *reduced* matrix element, which I explain later. Comparing theory to experiment is either done in terms of B-values or transition matrix elements $M(\hat{O})$, where $B(\hat{O}) = |M(\hat{O})|^2$. B-values are always positive real numbers, but there are some special cases where we need the phase information and thus work in terms of M.

We are interested in computing the transition matrix element $M(\hat{O})$ by which state $|i\rangle$ turns into state $|f\rangle$ via the operator $\hat{O}$, and to do so we need the numerical representations of these vectors and operator. The wavefunctions are produced as vectors by solving the many-body Schrödinger equation, which is discussed in Section 1.4. The operators we want are those for electric transitions $E\lambda$, magnetic transitions $M\lambda$, and beta decay.

The fundamental electromagnetic and weak interactions with nuclei are unified under **electroweak** theory: in a transition, the nucleus interacts with external fields mediated by force-carriers (gauge bosons) and in doing so will change from one state to another. For the electromagnetic force the gauge boson is the photon, and for the weak force the gauge bosons are the two charged $W^{\pm}$ and neutral Z^0. Properties of the bosons give us some hints as to the nature of the interaction: unlike the uncharged photon, the $W^{\pm}$ bosons can change charge of the particle it interacts with, and thus leave us with a different nuclide than what we started with (meaning, the electric charge of the nucleus changes with the decay). For example, β^- decay will change a neutron into a proton and release an electron and electron antineutrino: $(Z, N) \rightarrow (Z+1, N-1) + e^- + \bar{\nu}_e$.

The transition operators can be derived from first principles of relativistic quantum field theory, and a detailed explanation may be found in Walecka's book [128], Chapters 7 (EM) and 42 (EW). Due to their complexity, I will not include the derivations here, but I will point out some relevant facts from that analysis. The electromagnetic operators can be derived from the QED (quantum electrodynamics) Hamiltonian. The beta decay operators can be derived from the electroweak portion of the Standard Model Lagrangian. In both cases, one takes the **long-wavelength limit** (i.e. low-energy limit) and get the effective one-body operators for nuclei. The long-wavelength limit means our modeling of these physical processes is biased toward low-energy solutions. For the energy regimes relevant to my research, this is appropriate, but for physics in higher energy-density environments the appropriate descriptions of operators may change.

1.5.1 Electromagnetic transitions

Physically, an electromagnetic transition is characterized by the nucleus interacting with an external electromagnetic field, or rather the quanta of that field, the photon. The interaction is organized in the **multipole expansion** in terms of a positive integer multipolarity λ, which is the angular momentum carried by the operator which represents that external field, and the total transition is the sum over all values:

$$\hat{O}(\text{total EM}) = \sum_{\lambda} \hat{O}(E\lambda) + \hat{O}(M\lambda). \tag{1.13}$$

In this way, we can theoretically isolate each channel of the interaction: $\lambda = 0$ is the *monopole*, $\lambda = 1$ is the *dipole*, $\lambda = 2$ is the *quadrupole*, etc. The orientation of the external field also matters, so we keep track of that with μ which can take integer values from $-\lambda$ to $+\lambda$. Fermi's golden rule from Eq. 1.11 can be re-written a bit more explicitly for an EM transition:

$$T_{fi}^{\Sigma\lambda\mu} = \frac{2}{\epsilon_0 \hbar} \frac{\lambda + 1}{\lambda \left[(2\lambda + 1)!!\right]^2} \left(\frac{E_\gamma}{\hbar c}\right)^{2\lambda+1} |\langle J_f M_f | \hat{O}(\Sigma\lambda) | J_i M_i \rangle|^2, \tag{1.14}$$

where E_γ is the energy of the photon, and Σ is either E or M. Since the individual orientations (a.k.a. magnetic substates) are not typically observed, we sum over μ, M_i, M_f to get the complete transition probability

$$T_{fi}^{\Sigma\lambda} = \frac{1}{2J_i + 1} \sum_{M_i, \mu, M_f} T_{fi}^{\Sigma\lambda\mu}. \tag{1.15}$$

Note the factor in front with $(2J_i + 1)$; this averages over initial orientations, and we will actually wrap it and the summation into our definition of the transition strength $B(\Sigma\lambda)$.

$$T_{fi}^{\Sigma\lambda} = \frac{2}{\epsilon_0 \hbar} \frac{\lambda + 1}{\lambda \left[(2\lambda + 1)!!\right]^2} \left(\frac{E_\gamma}{\hbar c}\right)^{2\lambda+1} B(\Sigma\lambda : J_i \to J_f) \tag{1.16}$$

The reduced transition strength (called *B-value* for short) now contains the average over initial state orientations and summation over photon and final state orientations. The formula in Eq. 1.16 also allows us to convert from experimental half-lives to experimental B-values, which is usually how we compare model predictions to nature.

We now must turn our attention to the electromagnetic operators themselves. As stated, I will not derive these expressions, since such derivations are available in a number of nuclear physics textbooks, and also the details are not relevant to this dissertation. The operators are

$$\hat{O}(E\lambda\mu) = \sum_{j=1}^{A} e_j r_j^\lambda Y_\lambda^\mu(\Omega_j) \tag{1.17}$$

for electric, and

$$\hat{O}(M\lambda\mu) = \frac{\mu_N}{\hbar c} \sum_{j=1}^{A} \left[\frac{2}{\lambda + 1} g_{l,j} \hat{l}_j + g_{s,j} \hat{s}_j\right] \cdot \nabla_j \left[r_j^\lambda Y_\lambda^\mu(\Omega_j)\right] \tag{1.18}$$

for magnetic. Here, j is an index over nucleons and so values dependent on j really only depend on whether nucleon j is a proton or neutron. Note that these operators are dependent on μ, and our wavefunctions will be eigenstates of M as well; we will undo this later with the Wigner-Eckart theorem. The function $Y_\lambda^\mu(\theta, \phi)$ is the **spherical harmonic** and are the eigenfunctions of the usual orbital angular momentum operators:

$$\hat{\boldsymbol{L}}^2 Y_\lambda^\mu = \lambda(1+\lambda) Y_\lambda^\mu \tag{1.19a}$$

$$\hat{L}_z Y_\lambda^\mu = \mu Y_\lambda^\mu \tag{1.19b}$$

which are the square of orbital angular momentum and the z-component of orbital angular momentum respectively. We often split these operators into two parts: one which only acts on the proton wavefunction and the other which only acts on the neutron wavefunction, denoted by subscript p, n. The electric operator is then expressed

$$\hat{O}(E\lambda\mu) = e_p \left[r^\lambda Y_\lambda^\mu(\theta,\phi) \right]_p + e_n \left[r^\lambda Y_\lambda^\mu(\theta,\phi) \right]_n . \tag{1.20}$$

The constants e_p, e_n are called **effective charges**. For the free nucleon, these would just be the bare proton and neutron charges: 1 and 0 in units of e, the elementary charge. However, since these operators will work in a limited model space (i.e. the finite many-body basis) we will inevitably leave out some high-energy contributions. This truncation introduces errors, and one common way to partially alleviate the errors is to empirically fit effective charges. In other words, we allow the proton and neutron charges to inflate a little to accommodate some model shortcomings. These values are typically around $e_p = 1.4$ and $e_n = 0.4$, but differ depending on the application.

When using harmonic oscillator eigenstates for single-particle wavefunctions, we have another parameter that appears in matrix elements: the **oscillator length** b. This parameter is actually not a length, it is dimensionless, but it is related to the harmonic oscillator frequency ω by the relation $b^2 = \hbar/m\omega$. Increasing the oscillator length widens the oscillator potential and decreases the energy separation between states. Basically, the r^λ in the operator will contribute a factor of b^λ to the total matrix element. I mention this here because when doing UQ for the E2 transitions we consider b^2 as a parameter of the transition along with e_p, e_n, even though technically it is a parameter of the single particle wavefunction.

The electric operator I am most interested in is where $\lambda = 2$, called the **quadrupole**,

$$\hat{O}(E2) = e_p \left[r^2 Y_2^{\mu}(\theta, \phi) \right]_p + e_n \left[r^2 Y_2^{\mu}(\theta, \phi) \right]_n . \tag{1.21}$$

When computing single particle matrix elements, μ ends up being zero, but it is not obvious why this is. First, the spherical harmonic is complex-valued for $m > 0$. This is no problem in theory, but there is strong empirical evidence that $m > 0$ contributions are zero. Namely, imaginary components in matrix elements break time-reversal symmetry, i.e. creating or destroying energy. We see empirically that electromagnetic transitions are time-reversal symmetric, and thus deduce than any imaginary component of the electromagnetic transition must ultimately be zero. This is convenient for us, because (as discussed in section 1.3) we work in bases of many-body states with fixed M (that is, $M = J_z$, the z-component of angular momentum). In order for two of these states to be connected by an operator with $m > 0$, the initial and final M must differ. With $m = 0$, connecting states with the same M is trivial[3].

The magnetic dipole operator (M1) is simpler than the E2 in the sense that it has no spatial dependence:

$$\hat{O}(M1) = g_{sp}\vec{s}_p + g_{sn}\vec{s}_n + g_{\ell p}\vec{\ell}_p + g_{\ell n}\vec{\ell}_n . \tag{1.22}$$

Protons and neutrons each contribute a spin term and an orbital angular momentum term. The coupling constants $g_{sp}, g_{lp}, g_{sn}, g_{ln}$ may be treated as parameters (as I will show in UQ later), but they have default values for bare nucleons. The spin constants, also known as *gyromagnetic factors*, have bare values $g_{sp} = 5.586$ and $g_{sn} = -3.826$, and the orbital factors have bare values $g_{lp} = 1$ and $g_{ln} = 0$. In practice though, one may replace these with effective values, which I address in the UQ section.

---

[3] By *connecting* states here I mean that, for states i and f, that $\langle f|\hat{O}|i\rangle$ is not trivially zero.

1.5.2 Beta decay

Beta decay is an **electroweak** phenomenon, meaning it involves the electric force and the weak nuclear force; since nuclear structure is governed by the strong force, a complete understanding of beta decay of the nucleus cannot be had without wrapping quarks, leptons, and the force-carrying bosons into a single unified framework. The current incarnation of this framework, and arguably the most successful scientific theory ever developed, is the **Standard Model** of particle physics. Beta decay operators are derived electroweak sector of the Standard Model Lagrangian.

There are three distinct phenomena classified as beta decay: β^-, β^+, and electron capture (EC). They are described formulaically as follows. (Positron capture is relevant for some studies but only in sufficiently high energy densities.)

$$\begin{aligned}\beta^- &: A(Z,N) \rightarrow A(Z+1,N-1) + e^- + \overline{\nu}_e \\ \beta^+ &: A(Z,N) \rightarrow A(Z-1,N+1) + e^+ + \nu_e \\ EC &: A(Z,N) + e^- \rightarrow A(Z-1,N+1) + \nu_e\end{aligned} \tag{1.23}$$

Depending on how the operators change under the Laplace transform we give them special names: scalar (S), pseudoscalar (P), vector (V), axial-vector (A), and tensor (T). In principle, each of these components could be an important part of the weak interaction, however experimental tests have consistently showed that the vector and axial-vector parts dominate. The result is given the name $V - A$ theory, since the resulting form of the interaction is the vector (V) part minus the axial-vector (A) part. The other three couplings may not be strictly zero, but they are small by comparison[4].

For my work, we accept the $V - A$ theory and work in terms of two operators: the vector part, also known as the **Fermi** part, is simply an isospin (T) raising/lowering $\hat{\tau}_\pm$, while the axial-vector part, also known as the **Gamow-Teller** (GT), is a isospin raising/lowering and a spin flip $\vec{\sigma}\hat{\tau}_\pm$. That is, the Fermi component only changes T_z,

---

[4] Some work has been done to measure the S, P, and T contributions, which may be nonzero via electromagnetic induction, but such is beyond the scope of this dissertation.

and the GT changes T_z and can change total spin but up to one unit: $\Delta J = 0, \pm 1$. The Fermi component to any decay can be determined analytically, due to its simplicity, but the presence of spin operator in the GT means it is dependent on the wavefunction. Since experimental results contain a mix of Fermi and GT contributions, we usually remove the Fermi contribution from the experimental measurement and compare the result directly to $B(GT)$.

1.5.3 Computing transition strengths

The transition strength from state ψ_i to ψ_f is

$$B_{if}(\hat{O}) = \frac{|\langle J_f \| \hat{O}_K \| J_i \rangle|^2}{2J_i + 1}. \tag{1.24}$$

We often think of matrix elements as integrals, but in this case we do not compute it by evaluating the integral itself. Instead, we can move forward using the concept of **reduced matrix elements** and ultimately use **density matrices** to simplify the calculation. First, we note that the matrix element in Eq. 1.24, the quantity $\langle J_f \| \hat{O}_{J_t} \| J_i \rangle$ is already reduced in total angular momentum, meaning the states involved are independent of spatial orientation of the systems given by M_i, M_f, and M_t. I am using J_t and M_t to be the angular momentum and magnetic number of the transition operator. Many texts use K for angular momentum of the transition, but I prefer to be explicit here: a subscript t means that number belongs to the transition operator. The notation for reduced matrix elements is to use $\|$ inside the matrix element instead of $|$. This concept of "reduction" comes from the Wigner-Eckart theorem:

$$\langle J_f M_f | \hat{O}_{J_t M_t} | J_i M_i \rangle = \langle J_i M_i, J_t M_t | J_f M_f \rangle \frac{\langle J_f \| \hat{O}_{J_t} \| J_i \rangle}{\sqrt{2J_i + 1}}. \tag{1.25}$$

This tells us how to couple and uncouple angular momenta: a matrix element in the coupled basis (where $|J_i - J_t| \leq J_f \leq |J_i + J_t|$ and $M_i + M_t = M_f$) can be expressed as a product of a matrix element in a uncoupled basis and a Clebsch-Gordon coefficient. (Beware: some authors have other conventions for reduced matrix elements.)

The B-value thus is equivalent to a linear combination of non-reduced matrix elements,

$$\frac{1}{2J_i+1}\sum_{M_i}\sum_{M_f}\sum_{M_t}|\langle J_f M_f|\hat{O}_{J_t M_t}|J_i M_i\rangle|^2 = \frac{1}{2J_i+1}|\langle J_f||\hat{O}_{J_t}||J_i\rangle|^2, \tag{1.26}$$

where we average over initial state orientations, sum over final state orientations, and sum over external field orientations. The B-value however does depend on orientation in isospin space, because the z-component of isospin is a property of the nuclide, $T_z = (N - Z)/2$, which keeps track of charge. Our shell model calculations using BIGSTICK require a single fixed T_z for the basis. Because of this, we must "undo" the reduction in T using the Wigner-Eckart theorem; that is, multiply by the Clebsh-Gordon coefficient to couple isospin. For γ-transitions, charge is conserved so there is no issue, and the B-value is

$$B_{if}(\hat{O}_{J_t M_t T_t T_{z,t}}) = |\langle \psi_f : J_f T_f T_{z,f}|||\hat{O}_{J_t M_t T_t T_{z,t}}|||\psi_i : J_i T_i T_{z,i}\rangle|^2 \frac{|\langle T_i T_{z,i}, T_t\, T_{z,t}|T_f T_{z,f}\rangle|^2}{2T_f+1}. \tag{1.27}$$

However, if the transition changes the charge of the nucleus from T_{zi} to T_{zf} , as is the case with β-decay (where $|T_i - T_t| \le T_f \le |T_i + T_t|$ and $T_{z,i} + T_{z,t} = T_{z,f}$), then this presents a problem: how can we compute a matrix element $\langle J_f, T_f T_{zf}||\hat{O}_{J_t T_t T_{z,t}}||J_i, T_i T_{zi}\rangle$ using a single basis with fixed T_z? We can resolve this via a so-called **isospin rotation**; luckily as long as the states have good isospin, the rotation is completely handled by the Clebsch-Gordon. Since we already need to introduce the isospin Clebsch-Gordon coefficient in our expression, the only requirement for the rotation is that we choose a T_z for the basis which is less than or equal to both T_{zi} and T_{zf}. Then, we update the expression for the B-value as

$$B_{if}(\hat{O}_{J_t M_t T_t T_{z,t}}) = \frac{1}{2J_i+1}|\langle \psi_f : J_f T_f T_z|||\hat{O}_{J_t M_t T_t T_z}|||\psi_i : J_i T_i T_z\rangle|^2 \frac{|\langle T_i T_z, T_t\, T_{z,t}|T_f T_z\rangle|^2}{2T_f+1}, \tag{1.28}$$

where T_z is at least the charge *difference* carried by the operator (for β-decay it is ±1).

Lastly, we compute the **doubly reduced matrix element** (denoted with $|||$ around the operator) using density matrices:

$$\rho(a,b) = \langle\Psi|\hat{c}_a^\dagger \hat{c}_b|\Psi\rangle. \tag{1.29}$$

Specifically the **one-body density matrix** for a normalized state Ψ is as follows, using $\hat{c}$ as the annihilation operator. The interpretation of the density matrix is quite intuitive: the density operator removes a particle from state a and then creates a particle in state b, so it is a measure of which single-particle states are occupied in Ψ. Without accounting for angular momentum, the one-body matrix elements of $\hat{O}$ would be computed like

$$\langle\psi_f|\hat{O}|\psi_i\rangle = \sum_{ab} \rho(a,b)\langle a|\hat{O}|b\rangle. \tag{1.30}$$

However, we want the doubly reduced density matrices from state ψ_i to state ψ_f, so we introduce a slight modification by giving the density operator a well defined angular momentum J_t and writing it as reduced in angular momentum (per our convention) and reduced in isospin as well.

$$\rho^{fi}_{J_t T_t}(a,b) = \frac{\langle\psi_f : J_f T_f||| \left[\hat{c}_a^\dagger \otimes \tilde{c}_b\right]_{J_t T_t} |||\psi_i\rangle}{2J_t + 1} \tag{1.31}$$

The notation for coupling the product of operators to angular momentum K and magnetic number M is $\left[\hat{c}_a^\dagger \otimes \tilde{c}_b\right]_{J_t M_t}$ which is equal to a linear combination of uncoupled operators with Clebsh-Gordon coefficients:

$$\left[\hat{c}_a^\dagger \otimes \tilde{c}_b\right]_{J_t M_t} = \sum_{m_a m_b} \langle j_a m_a, j_b m_b|J_t M_t\rangle \hat{c}^\dagger_{j_a m_a} \tilde{c}_{j_b m_b} \tag{1.32}$$

where the time-reversed annihilation operator is $\tilde{c}_{j,m} = (-1)^{j+m} c_{j,m}$. The subtlety here is that the annihilation operator is actually not a spherical tensor, and in order to use the SU(2) rules of angular momentum coupling we must be working with spherical tensors. Luckily, the time-reversed version is a spherical tensor.

With the doubly reduced density matrix in hand, we can compute the doubly reduced matrix element,

$$\langle \psi_f : J_f T_f T_z ||| \hat{O}_{J_t M_t T_t T_z} ||| \psi_i : J_i T_i T_z \rangle = \sum_{ab} \rho^{fi}_{J_t M_t}(a,b) \langle a ||| \hat{O}_{J_t M_t T_t T_z} ||| b \rangle \tag{1.33}$$

which we can plug into the B-value in Eq. 1.28. The doubly-reduced one-body matrix elements of the transition operator, $\langle a ||| \hat{O}_{J_t M_t T_t T_z} ||| b \rangle$, depend on the operator in question but are usually not difficult to compute. They are computed once, stored, then referred to for many calculations of many-body matrix elements.

1.6 Nuclear reactions and scattering

This dissertation is not about scattering theory, but in order to do proper machine learning on reaction cross section evaluations we must understand the physical importance of relevant quantities. A **nuclear reaction** is a process which begins with an initial nucleus, ends with a final nucleus, and is described by a interaction with some other nuclei or particles. The initial and final nuclide may be the same, for example $^{24}\text{Mg} + n \rightarrow ^{24}\text{Mg} + n'$, which we can imagine as a energetic neutron colliding with a magnesium-24 nucleus, depositing some energy, and bouncing off. The initial and final nuclides can also be different, for example one of Ernest Rutherford's scattering experiments tested the reaction $^{14}\text{N} + \alpha \rightarrow ^{17}\text{O} + p$: a nitrogen-14 target is hit with an alpha particle (a.k.a. a ^{4}He nucleus, made of two protons and two neutrons), and produces oxygen-17 and an extra proton. In both these examples we have a target and a projectile; we know the general attributes of their initial and final states, and what happens in between is governed by some complex scattering phenomena.

The simplest theoretical picture of quantum scattering is to represent the incoming particle with a plane wave incident upon the nucleus, then represent the outgoing particle as a spherical wave travelling away from the nucleus. So we would look for solutions to the Schrodinger equation with the form

$$\Psi \approx \left[e^{ikz} + f(\theta)\frac{e^{ikr}}{r} \right]. \tag{1.34}$$

An illustration of the scattering event is shown in Fig .1.4. Note than the wavenumber, k, related to energy by $k = \sqrt{2mE}/\hbar$, is the same for the incoming particle and outgoing particle, and we call this situation *elastic* scattering. For *inelastic* scattering, wherein some energy is deposited in or withdrawn from the target, one must account for different wavenumbers. The proper handling of that typically involves a partial wave analysis and is not particularly relevant for this dissertation. Under these assumptions, the quantity of interest is the **scattering amplitude**, $f(\theta)$, which describes the polar distribution of the scattered particles. Treating this like a probability amplitude, we can square it to get the so-called **differential cross section**

$$d\sigma = |f(\theta)|^2 d\Omega, \quad \text{or} \quad \frac{d\sigma}{d\Omega} = |f(\theta)|^2. \tag{1.35}$$

The solid angle element $d\Omega$ is a small patch of the detector through which outgoing particles pass, and $d\sigma$ is a small increase in the cross section of particles scattered at polar angle θ and reaching the $d\Omega$-sized area. The differential cross section as I wrote it is defined for a projectile energy $E = k^2\hbar^2/2m$, but we can easily imagine our hand on the knob controlling energy of the projectile beam. As we turn up the energy, $d\sigma/d\Omega$ is going to change and allow us to probe the physics underlying the scattering process. Integrating over the polar angle we can get the **total cross section**, as a function of the energy of the incoming particle that is

$$\sigma_{\text{tot}}(E) = \int d\Omega \sigma(\theta, E). \tag{1.36}$$

This is the quantity being studied in my ML research: the cross section as a function of energy. Relevant scattering energies are on the order of 10 MeV, which is on the low end of possible nucleon-nucleus scattering experiments.

The current corpus of nuclear cross section data has contributions from many groups in many countries around the world. Groups in China, Russia, Japan, Europe (under the Organization for Economic Co-operation and Development), and the USA have issued up-to-date comprehensive libraries, not to mention the data maintained and issued by the International Atomic Energy Agency or other libraries with a special purpose (i.e. fusion research, etc.). Thanks to the enourmous efforts of nuclear data scientists, much of this data can be accessed through the evaluated nuclear data files (ENDF) database [25]. ENDF is the name of a reaction evaluation library created by groups in the USA, but the online ENDF database allows access to many other libraries as well. Fig. 1.5 shows an example of the ENDF data retrieval tool.

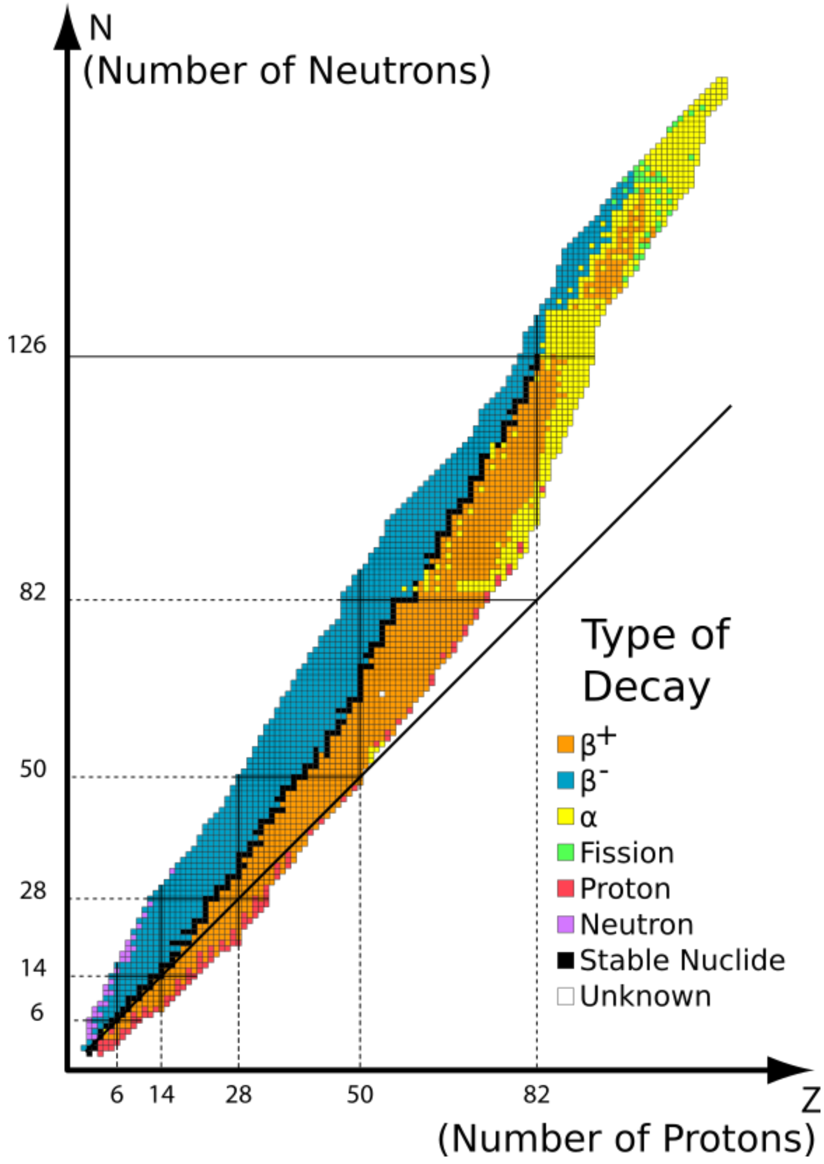


Figure 1.1. The chart of nuclides.

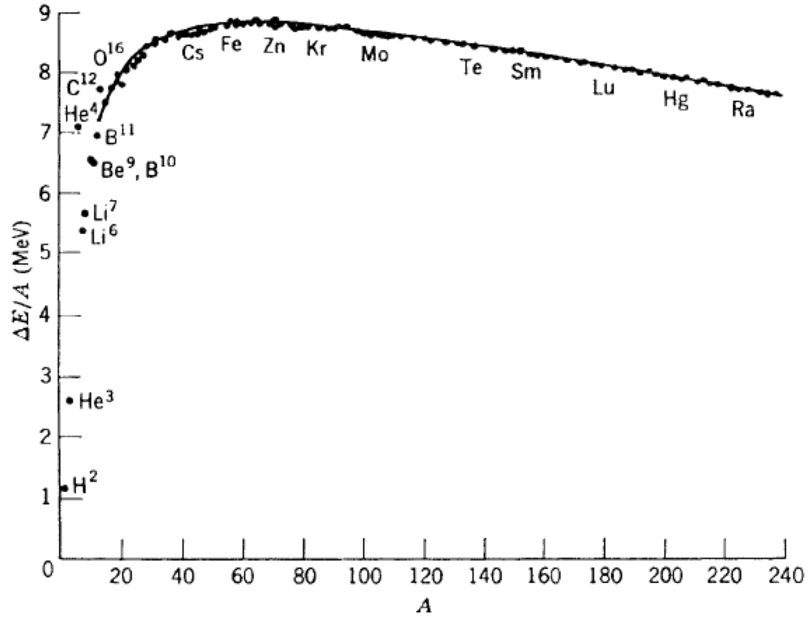


Figure 1.2. A plot of binding energy per nucleon (MeV) as a function of mass number A. With the exception of light nuclei, $\Delta E/A$ is almost a constant.

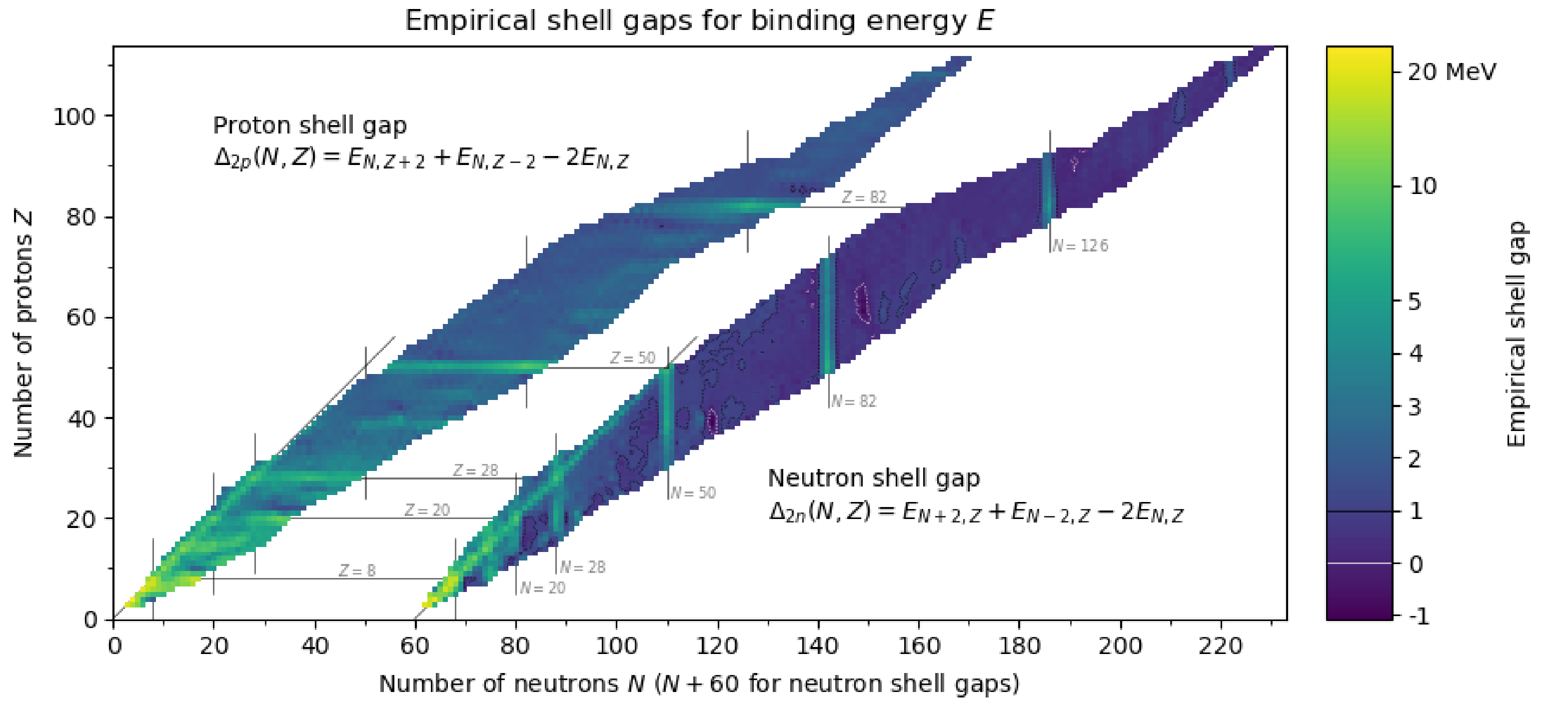


Figure 1.3. The empirical shell gap across the nuclear chart. The "shell gap" measure $\Delta_{2p,2n}(N,Z)$ is proportional to the second derivative (finite difference) approximation of binding energy with respect to proton/neutron number (with a step size of 2). Binding energies, as shown in Fig. 1.2, increase nearly linearly for much of the chart, so the second derivative measure should be small on average. Regions with a larger second derivative (concave-up) in the landscape of binding energy indicate shell structure because shell phenomena cause particles to be more tightly bound. This is visible along lines where N, Z are one of the magic numbers: 2, 8, 20, 28, 50, 82, 126, etc. as labelled on the chart. [Figure created by Mia Dobson, CC BY-SA 4.0, link.]

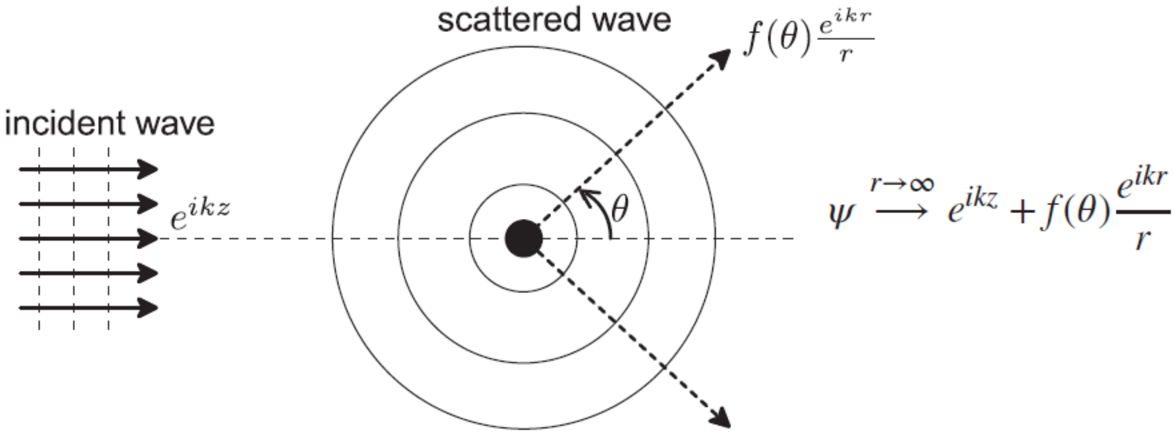
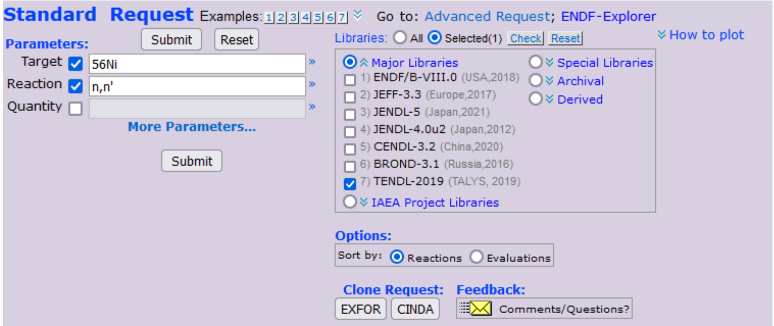


Figure 1.4. Illustration of the basic (elastic) quantum scattering event. (Figure by Jay Wang, 2020)


Standard Request Examples: 1 2 3 4 5 6 7 Go to: Advanced Request; ENDF-Explorer

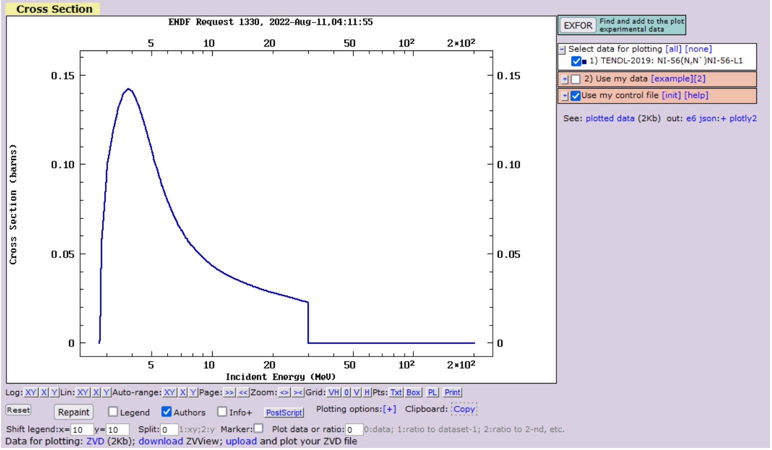
Parameters:
Target ☑ 56Ni
Reaction ☑ n,n'
Quantity ☐

Libraries: ○ All ● Selected(1)

● Major Libraries
☐ 1) ENDF/B-VIII.0 (USA,2018)
☐ 2) JEFF-3.3 (Europe,2017)
☐ 3) JENDL-5 (Japan,2021)
☐ 4) JENDL-4.0u2 (Japan,2012)
☐ 5) CENDL-3.2 (China,2020)
☐ 6) BROND-3.1 (Russia,2016)
☑ 7) TENDL-2019 (TALYS, 2019)
○ IAEA Project Libraries
○ Special Libraries
○ Archival
○ Derived

Options:
Sort by: ● Reactions ○ Evaluations

(a) The ENDF retrieval options. Note the various libraries accessible.


ENDF Request 1330, 2022-Aug-11,04:11:55

Select data for plotting [all] [none]
☑ 1) TENDL-2019: NI-56(N,N`)NI-56-L1
☐ 2) Use my data [example][2]
☑ Use my control file [init] [help]

See: plotted data (2Kb) out: e6 json:+ plotly2

☐ Legend ☑ Authors ☐ Info+

Shift legend:x= 10 y= 10 Split: 0 1:xy;2:y Marker: ☐ Plot data or ratio: 0 0:data; 1:ratio to dataset-1; 2:ratio to 2-nd, etc.

Data for plotting: ZVD (2Kb); download ZVView; upload and plot your ZVD file

(b) The retrieved plot of the inelastic neutron scattering cross section evaluation for ^{56}Ni in the TENDL library.

Figure 1.5. Example usage of the ENDF data retrieval service.

CHAPTER 2

UNCERTAINTY QUANTIFICATION IN PHYSICS

The fundamental idea of uncertainty quantification (UQ) for theoretical calculations is that no physical theory or model exactly represents nature, and that we can gain insight by studying discrepancy between our simulator and experimental observation. Providing theoretical predictions with error bars that reflect the true limits in our knowledge of a physical system allows for meaningful comparisons between different theoretical models. Even if the assigned uncertainty is an approximation, as it often is in practice, it is useful to theorists and experimentalists alike. As such, UQ can be considered a driver of the experiment-theory feedback loop. Proper UQ on the theoretical side can be used to identify the experimental measurements that will have the largest impact in reducing such uncertainty. New measurements, with their own error bars, can then be used to improve current theoretical models.

Gelman's book *Bayesian Data Analysis* [51] describes the data analysis process in three steps, which I paraphrase here.

1. Set up a probability model for the quantities of interest. This may include experiment, computer simulation, or parameters to a computer model. The goal is to correctly identify what your target variables depend on. We may need to learn some probability theory, but this step is not too difficult.

2. Put the formal probability model in a form you can compute. We construct the *posterior* distribution for the quantity of interest, usually dependent on some experimental data. This step is the most challenging and likely the most contentious. Here is where we may introduce approximations, choice of emulators, etc. (Any mistakes made here will certainly propagate to the outcome, so be careful.)

3. Evaluate the model. How this is done depends on the form of the model itself, but the result is a description of the posterior distribution and thus we can

> analyze model fit, implications, etc. In the Bayesian paradigm, the posterior distribution informs our belief about the variables in question. Before much computational development had gone into UQ, this part would have been quite challenging as well, but recent advances in computational methods and machine learning have made evaluation easier than ever.

We describe an experimental measurement y as

$$y(A) = m(A, \theta) + \delta(A, \theta) + \epsilon(A), \tag{2.1}$$

where m is our simulator for the quantity y, A denotes attributes of the physical system being modeled, θ is a set of simulator parameters, δ is the systematic discrepancy in the simulator, and finally ϵ is the error in the experiment, including statistical and systematic errors [71, 115]. We use the term **simulator** to mean the mathematical model as well as the code implementation; although errors and uncertainty can technically stem from code implementation itself, we combine these factors under the assumption that code errors and algorithmic/numerical errors are smaller than all other sources of error. The experimental values $y(A)$ are tabulated with corresponding uncertainties $\epsilon(A)$ being reported by the experimentalist. The discrepancy $\delta(A, \theta)$ is entirely dependent on the problem at hand. For some problems we have very little knowledge of systematic structures and so modeling δ in its own right is valuable for progressing theory. However, in much physics research, there is a general understanding of discrepancy and as such modeling δ directly is not very fruitful. A good example of this in physics is anywhere perturbation theory is used to find a leading-order contribution: doing UQ only to determine the known higher-order corrections is clearly a redundant effort. Rather than attempting to improve the model by explicitly reducing discrepancy, we want to quantify the uncertainty of model parameters θ. Because of this, we consider the total error $\delta(A, \theta) + \epsilon(A) = y(A) - m(A, \theta)$, and uncertainty on θ can be determined by studying this relationship for many different cases or systems A.

We begin by asking a question: given experimental observations, how do we describe behaviour of the variable (parameters) of interest θ? The first step in answering this question is formulating **Bayes' rule**,

$$P(\theta|y) = \frac{P(y|\theta)P(\theta)}{P(y)}. \tag{2.2}$$

The **posterior** $P(\theta|y)$ is the ultimate goal of our analysis: a distribution describing the parameters of interest θ given observations y. Bayes' rule says the posterior can be computed from existing (or at least, attainable) knowledge of these variables. The **likelihood** distribution $P(y|\theta)$ describes the converse of the posterior: how observations y behave given an instance of θ. The **prior** $P(\theta)$ is the probability distribution of θ according to our prior knowledge. Finally, the **evidence** $P(y)$ describes the distribution of y according to our knowledge.

A short aside, in the effort to develop some intuition – we are using particular notation to describe variables conditioned on other variables: $A|B$ means A *given* B. This is related to intersection by

$$P(A|B) = P(A \cap B)/P(B) \qquad\qquad P(A \cap B) = P(A|B)P(B). \tag{2.3}$$

Since probabilities are always between zero and 1, we know that $P(A \cap B) \leq P(A|B)$: the probability of two events occurring together (intersecting) is always less than or equal to one occurring given the other, and always less than or equal to either of the individual probabilities $P(A)$ and $P(B)$.

Our next step is to decide the likelihood, and in the context of comparing a numerical model to experiment, the most common choice for $P(y|\theta)$ is defined in terms of the χ^2 ("chi-squared"), which is a simple measure of model error. The sensible assumption in using this form of likelihood is that the probability of $P(y|\theta)$ is equal to the product of individual probabilities $P(y_\alpha|\theta)$, so

$$\chi^2(\theta) = \sum_\alpha \left(\frac{y_\alpha - m_\alpha(\theta)}{\Delta y_\alpha} \right)^2 . \tag{2.4}$$

The index α denotes each observation, and the factor Δy_α is the experimental uncertainty (one standard deviation). While the latter is often equal to ϵ in my formulation above, that is not always the case and computing it may be a bit subtle; in that case one must remember that separate (uncorrelated) contributions to uncertainty add in quadrature. The likelihood is conventionally defined $P(y|\theta) = \exp\left[-\frac{1}{2}\chi^2(\theta)\right]$. This distribution is Gaussian in terms of $m_\alpha(\theta)$ but is generally *not* Gaussian in terms of θ.

Our next step is to define the prior $P(\theta)$. Assigning probability distributions to variables is nontrivial, and very much research has been done regarding choices of priors [119, 51]. In the Bayesian context, the prior encodes our existing *a priori* knowledge of θ, and as such is entirely problem dependent. We do have guidance, however, in the form of the **principle of maximum entropy** (MaxEnt): the best choice of probability density $p(x)$ is one that maximizes entropy $S = -\int dx p(x) \log p(x)$ subject to some constraints (probabilities sum to 1, etc.). The simplest result gives the *principle of insufficient reason*: when *nothing* is known about θ in advance, let all outcomes of θ have the same probability. However, with physics models, this is rarely the case. One may argue that using a uniform prior is warranted for practical reasons, possibly based on the difficulty of designing an informative prior, but physics models are rarely designed with arbitrary parameterizations. Some results of MaxEnt are given in Table 2, depending on the *a priori* knowledge.

a priori knowledge	MaxEnt distribution
Nothing	Uniform
Mean	Exponential
Mean and standard deviation	Gaussian

One can probably see now why Gaussian priors are so common: physicists will generally be able to estimate mean and standard deviation of the parameter, but rarely bother with higher moments. This is also why the Gaussian likelihood is so common for experimental data which include errors. In the larger picture, we maintain that scientists should always work to remove subjectivity: no two researchers should do the exact same statistical analysis and come up with different answers. This in itself is not a very controversial point. However, it is my belief that use of an informative prior, even when it is technically an inaccurate representation of beliefs due to the parameterization being wrong, is still very useful and warranted. One simply must make it explicitly clear what the assumptions were, as is always the case. With advances in numerical evaluation of probability distributions (e.g. fast Markov Chain Monte Carlo), one may include several choices of prior distribution and present them side-by-side in the research.

A common trick to deal with the evidence $P(y)$ in the denominator of Eq.2.2 is to wrap it into a proportionality constant. The evidence has no θ dependence, thus can only contribute to the overall scaling of the posterior. In a typical UQ analysis, a normalized posterior need not be obtained anyway, since it is relative probabilities that are informative. So, we seldom keep track of proportionality constants,

$$P(\theta|y) \propto P(y|\theta)P(\theta). \tag{2.5}$$

Bayes' rule now looks deceivingly simple; evaluating this numerically can be difficult and require considerable computation power. Luckily MCMC evaluation is easier and more powerful than ever; in Python my preferred tool is the `emcee` library [40], an affine-invariant ensemble sampler, which parallelizes and scales efficiently. Matlab and R have their own built-in MCMC libraries, and many open-source implementations can be found for C and Fortran. There are even programming languages developed specifically for Bayesian statistical analyses, with statistical principles built into their design; a popular one is Stan [52], but there are many more.

CHAPTER 3
INTRODUCTION TO DEEP LEARNING

3.1 A review of artificial neural networks

Any deep learning system has three key components: the neural network architecture M, the data (X, Y), and the loss function $\mathcal{L}$. The objective is to find the function mapping $M : X \to Y$ by training the network on data. The neural network is a highly nonlinear composite function with parameters (weights) W, mapping input data $x \in X$ to $y \in Y$, so we may write $M(W; x) = y'$ where y' is ideally close to y. The accuracy is determined by training and the loss function which very often compares the model **prediction** y' to the **target** data y and returns a number $\mathcal{L}(y, y')$ which becomes smaller as $y' \to y$. The ML practitioner uses an optimization algorithm to iteratively change the networks wights W as to minimize (on average) $\mathcal{L}(y, M(W; x))$ for all $x \in X$ and $y \in Y$. This optimization process typically uses a stochastic gradient descent (SGD) algorithm, which uses **backpropagation** (automatic differentiation) to determine the gradient of the loss function at each iteration and find a new parameterization $W \to W'$ with some randomness involved. In general this is a global optimization problem, and there may be many local minima which give a poor solution. The random properties of SGD allow for local minima to be reached and overcome as training continues.

Often the first task of a ML practitioner is to look at the data, clean it, normalize it, and/or generally put it into a proper form for machine learning. This takes many forms and is entirely dependent on the problem, but there are a few guiding principles. First, data must be as clean as possible: all data should be of the same type and shape, with no missing values. Second, the data should probably be normalized; there are a number of common ways to normalize data and this likely will depend on network architecture as well. Of course, some problems may require the model to learn different scales and thus normalization would not be appropriate, so this depends on the situation. Third, it may be optimal to transform or compress the data. For example, consider high-dimensional image data: will the ML system perform better on the images themselves, or on a truncated Fourier transform of the images? Usually this requires some experimentation, but simplifying data representation can make a big difference in model accuracy.

The simplest deep neural network design is a densely-connected feed-forward network, shown in Fig. 3.1, called the **multi-layer perceptron** (MLP). The network receives input x with dimension p and outputs y (1-dimensional in this example). The middle layer, called a **hidden layer**, takes inputs and performs the operation $f(W_0x) = v$ where f is a nonlinear **activation function**, W_0 is the weight matrix of that layer, and v is the resulting vector. Notice that every input node connects to every node of the hidden layer; this is called a **dense** connection. The output layer does the operation $g(W_1v) = y$ where W_1 is another set of weights, g is a function mapping onto the output range.

The function f is an **activation function**. Probably the most common *modern* activation function is the rectified linear unit, **ReLU**, which is zero for all negative numbers and identity for all positive numbers. The ReLU is very powerful and there have been many studies exploring its efficacy [32, 136]. I will mention two other activation functions, but there are many more. First, the **Leaky ReLU** [83] was invented to fix the vanishing gradient problem for negative ReLU inputs: rather than zero for negative numbers the Leaky ReLU becomes negative with a very small positive slope, hence it is allowed to "leak" a little energy from the activations. Second the **sigmoid linear unit**, **SiLU/Swish** [106], defined $x/(1-e^{-x})$, is an everywhere-smooth function with a single minimum near -1. Which activation works best is particularly dependent on features in the training data: despite not being smooth, the kink in ReLU may perform best in learning data with sharp changes. Likewise the SiLU may be an optimal choice when data contains smoother trends. This is mostly heuristic though, and in practice one simply tries several and uses the best for the present problem. For example, the SiLU may outperform the ReLU in many cases, but it is also more expensive to compute; for very large networks the extra time required to compute SiLU for many nodes might not be justified by those gains.

The function g maps from the internal representation $f(W_0x) = v$ to the output $g(W_1v) = y$. Some popular choices are *linear* (identity), the same function used for activation (if appropriate), and a **sigmoid**. A sigmoid is not one function but a family with similar properties; they accept a real number input on $(-\infty, +\infty)$, are monotonically increasing, differentiable everywhere, and have exactly one inflexion point (i.e. the second derivative has one root). Some examples are shown in Fig. 3.2. Sigmoids are good for mapping the real numbers to probabilities. Sigmoids have historically been used as activation functions as well, but are not used as such in modern deep learning applications because they suffer from some issues such as vanishing gradients.

In order to train the MLP model $M(W;x)$ to emulate a *function* $\phi(x)$, we need many evaluations of that function for training and validation, and we must define a differentiable **loss function**, $\mathcal{L}(\phi(x), M(W;x))$. In general, the form of the loss function is dependent on the task, but we have a few standard options [15]. First, mean-squared error (MSE) is a common choice for regression problems because it grows quadratically with error thereby strongly emphasizing poor predictions. However, this also means that MSE is susceptible to being saturated by outliers. If outliers are a problem, one might use mean absolute error (MAE) which only grows linearly with error; however, MAE is not differentiable everywhere which may cause problems. Lastly, binary cross-entropy (BCE) is useful particularly for classification problems where both ϕ and M are probabilities (that is, $\phi, M \in [0,1]$). All three of these loss

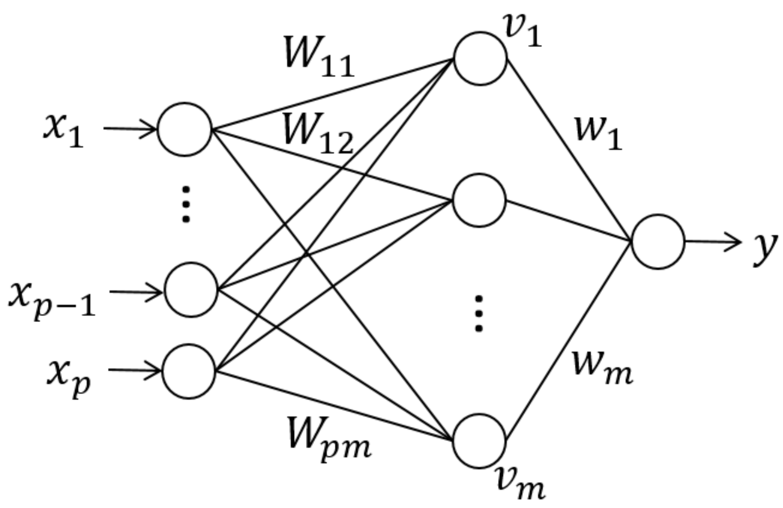


Figure 3.1. The multi-layer perceptron, the simplest useful deep neural network. Information flows from left to right, mapping the input $\boldsymbol{x}$ to output y. The multiplicative weights W and w are optimized during training.

functions have the property that every contribution to the loss function is positive, so we need not be concerned about cancellations.

$$\begin{aligned}
\mathcal{L}_{MSE}(\boldsymbol{x}) &= \frac{1}{N}\sum_{i=1}^{N}(y(\boldsymbol{x}) - \phi(\boldsymbol{x}))^2 \\
\mathcal{L}_{MAE}(\boldsymbol{x}) &= \frac{1}{N}\sum_{i=1}^{N}|y(\boldsymbol{x}) - \phi(\boldsymbol{x})| \\
\mathcal{L}_{BCE}(\boldsymbol{x}) &= -\frac{1}{N}\sum_{i=1}^{N}\phi(\boldsymbol{x})\log(y(\boldsymbol{x})) + (1-\phi(\boldsymbol{x}))\log(1-y(\boldsymbol{x}))
\end{aligned} \tag{3.1}$$

The MLP and other networks like it may be used for two kinds of problems: regression and classification. Before going into more complicated tasks, we must understand these two. Generally speaking, if our model is mapping to a **continuous** variable then we have a regression problem, and otherwise if our model is mapping to a **discrete** variable then we have a classification problem. Input variables, which can be continuous or discrete, are considered independent.

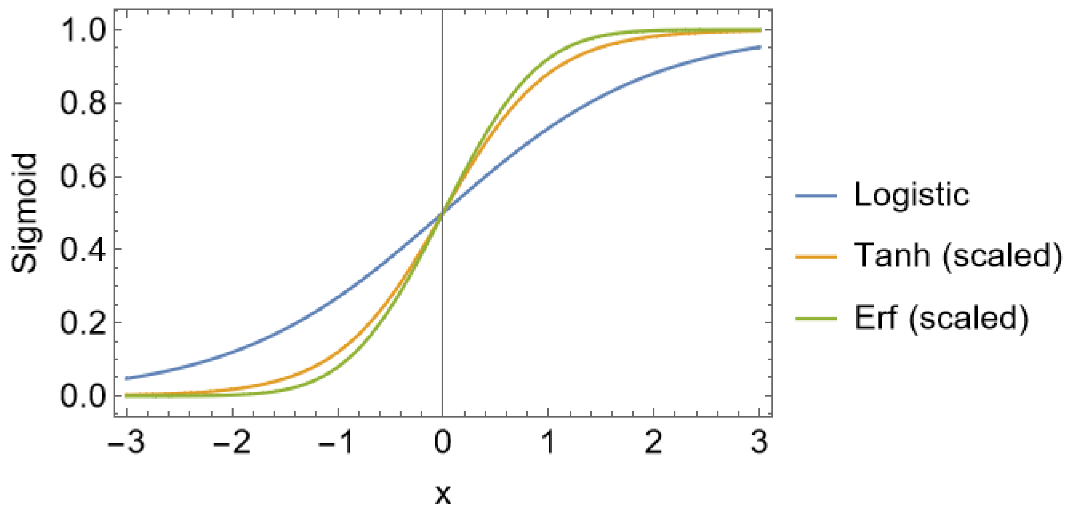


Figure 3.2. Some examples of sigmoid functions: the logistic function $(1 - e^{-x})^{-1}$, **a scaled hyperbolic tangent** $(\tanh(x) + 1)/2$ **where** $\tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$, **and a scaled error function** $(\mathrm{erf}(x) + 1)/2$ **where** $\mathrm{erf}(x) = (2/\sqrt{\pi})\int_0^x \exp(-t^2)dt$. **Sigmoids are useful for mapping the real numbers to probabilities.**

3.2 A brief history of deep learning

Machine learning (ML) is a broad field encompassing many different methods and applications. It is intimately related with the field of artificial intelligence, and as illustrated in Fig: 3.4, the overlap of the two contains so-called **deep learning** which is relevant for this dissertation. ML is often described strictly as a subset of AI, but I believe that view is a bit outdated. Some aspects of AI research have very little to do with computer implementations, like the politics and sociology of robots. On the other side, plenty of useful ML algorithms can hardly be considered intelligent, such as regression algorithms, decision trees, and Gaussian processes. Certainly those can be considered the basis for contemporary deep learning, but our definitions of AI will naturally change as technology develops. Some machine learning tasks are easy for humans to perform, but are quite difficult for us to detail the rules for or write a computer code to perform: this is where ML enters into AI. Deep learning is useful for problems wherein detailing steps toward the solution is too complicated, and yet one has plenty of examples of correct execution: these include image

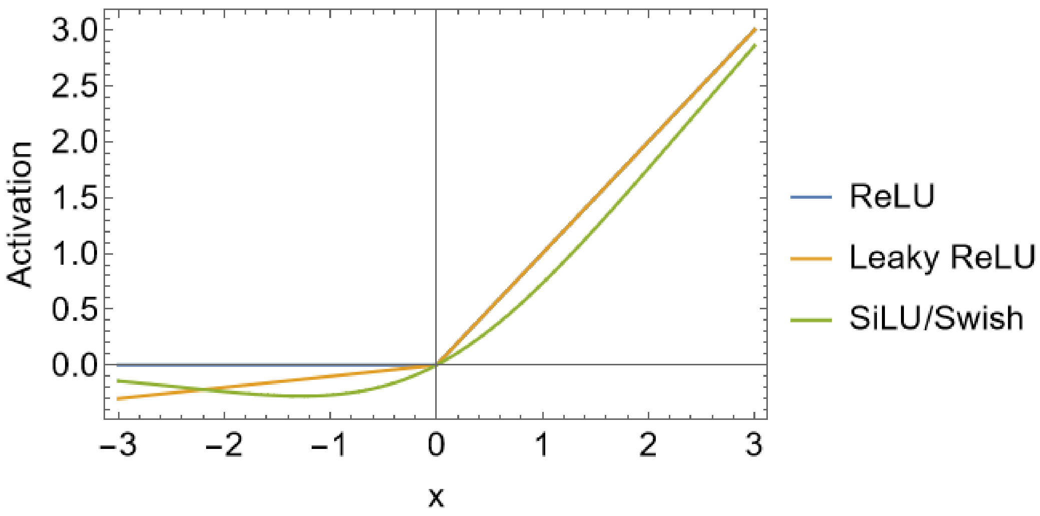


Figure 3.3. Some examples of activation functions.

recognition/classification, speech recognition, generating data/images by example, and controlling complex systems like cars and robots.

Many methods and concepts we now recognize as essential to deep learning are not new inventions. It is likely the first **artificial neural network** was invented by Frank Rosenblatt [113] in 1958 at Cornell Aeronautical Laboratory. Rosenblatt called his invention the **perceptron** and he was mainly concerned with concepts of storing "memory", representing input data in what we would now call the network parameters or weights. One idea that was developed early in cybernetics, which was the name for neural network research at the time, and remains useful in 2022 is using a **stochastic gradient descent** algorithm to optimize weights of the network. However, early models were linear, so they could only learn functions of the form $f(\boldsymbol{x}, \boldsymbol{w}) = \sum_i w_i x_i$. The inability for linear models like the perceptron to learn even very simple nonlinear functions (famously, the exclusive *or* operation, XOR) led researchers to be skeptical of their promise. In the 70s and 80s however, a renewed interest led to a few breakthroughs including Kunihiko Fukushima's Cognitron and Neocognitron models [47, 48] for image recognition, which formed the basis of modern **convolutional**

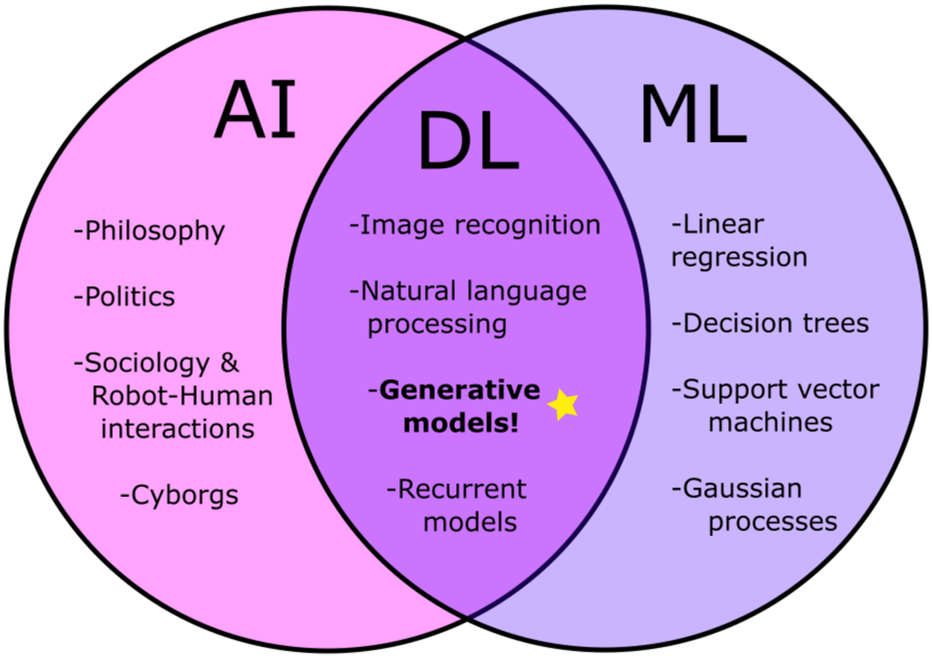


Figure 3.4. Deep learning (DL) exists within (but is not equal to) the intersection of artificial intelligence (AI) and machine learning (ML). In my work I use deep generative models ($\star$) to make predictions about physical systems.

neural networks and included the first use of a **nonlinear activation function**, the rectified linear unit (ReLU).

The so-called deep learning revolution around 2012 had several main factors: unprecedented accuracy in unsupervised learning problems, models outperforming humans at image recognition and games, and significant advances in hardware and software. To name just a few: the now-famous convolutional model ImageNet[78], robust speech recognition with large vocabularies [26], and development of distributed training methods [27] (which would ultimately prompt the construction of TPU architectures). AlphaGo [118, 3] beating high-ranking Go players Fan Hui (2015), Lee Sedol (2016), and #1 Go player in the world Ke Jie (2017) illustrated the superhuman capabilities of deep learning systems on a world stage.

CHAPTER 4

PROJECT 1A: UNCERTAINTY QUANTIFICATION IN SHELL MODEL CALCULATIONS

Published in Phys. Rev. C [42] and available on arXiv.

4.1 Introduction

Recent advancements in nuclear theory have emphasized the importance of theoretical uncertainty quantification (UQ) [28] with applications to, among other things, the nuclear force and effective field theory [131, 49, 23, 100, 87, 133], the optical model [81, 96], density functional theory [116, 86] and the configuration-interaction shell model [64, 139].

The shell model, which provides a useful conceptual framework for nuclear structure, can be approximately divided into *ab initio* and empirical/phenomenological approaches. *Ab initio* calculations, such as the no-core shell model [97, 9], typically use forces built upon chiral effective field theory [127] and thus are arguably more fundamental and also have been subject to considerable UQ [131, 49, 23, 100, 87, 133], but are limited to light nuclei, approximately mass number $A < 16$. Empirical shell model calculations [22, 21, 24] have a long, rich, and successful history, and, importantly, have been applied to a wide range of nuclei far beyond the $0p$ shell, but the theoretical underpinnings are more heuristic: individual interaction matrix elements in the lab frame (single-particle coordinates) are adjusted to reproduce experimental data.

(We will not consider here related but distinct methodologies such as coupled clusters [58], and we note but do not comment further on efforts to construct interactions that ‘look like’ traditional empirical calculations but are derived with significant rigor from *ab initio* forces [122].)

Previous work on UQ in the shell model focused on $0p$-shell calculation: one considered a simple interaction with only seven parameters, examining correlations using a singular-value-decomposition analysis [64]; while the other used 17 parameters but did not consider correlations between parameters [139].

Because of the broad applications and demonstrated utility of the empirical shell model, we carry out a sensitivity analysis on an widely-used, ‘gold standard’ empirical shell-model interaction, Brown and Richter’s universal *sd*-shell interaction, version B, or USDB [19]. Here, ‘*sd*-shell’ means the valence space is limited to $1s$ and $0d$ single-particle orbits, with an inert ^{16}O core.

In fitting their interaction, Brown and Richter followed a standard procedure [22]. They minimized the total error with respect to experiment, defined as the χ^2-function in Eq. (4.5) below, by taking the first derivatives with respect to the parameters, which yield the linear response of calculated energies to perturbations of the parameters, and then carried out gradient descent on the independent parameters, here 63 two-body matrix elements and three single-particle energies. In the fit they found that about five or six linear combinations of parameters, found by singular value decomposition as we do below, were the most important. (Interesting, a similar result was found for random values of the matrix elements [66]). Brown and Richter actually produced two interactions [19], USDA, which was found by fitting the first 30 linear combinations from singular value decomposition, and USDB, found by fitting 56 linear combinations.

For a Bayesian sensitivity analysis, discussed more fully in Chapter **??**, one must characterize the likelihood function for model parameters. In Laplace’s approximation, one assumes the likelihood is well approximated by a Gaussian, which corresponds to a quadratic expansion in the χ^2-function. Even so, the matrix of second derivatives of χ^2

(which, more rigorously, is the log-likelihood), or the *Hessian*, needed is quite demanding to obtain.

We therefore consider a further simplification, approximating the Hessian by the same linear response (first derivatives of the energies), which are efficiently computed by the Feynman-Hellmann theorem [59, 38]. As discussed below, this principal component analysis of the sensitivity is, in this approximation, singular value decomposition of the linear response. Importantly, we find that numerical corrections to the linear response matrix are small, making this approximation appealing for studying larger spaces wherein the full numerical calculation is too costly.

4.2 The empirical configuration-interaction shell model

For details on the shell model, see Section 1.3.

We assume a frozen ^{16}O core and use the $1s$-$0d$ single-particle valence space, also called the *sd*-shell. Assuming both angular momentum J and isospin T are good quantum numbers, one has only three independent single-particle energies and 63 independent two-body matrix element, for a total of 66 parameters. Because each of those parameters appears linearly in the Hamiltonian, we can write

$$\hat{\mathcal{H}} = \sum_i \lambda_i \hat{\mathcal{O}}_i \tag{4.1}$$

where $\hat{\mathcal{O}}_i$ is some dimensionless one- or two-body operator. Thus the parameters $\boldsymbol{\lambda}$ have dimensions of energy.

The set of parameters $\boldsymbol{\lambda} = \{\lambda_i\}$ we use are Brown and Richter's universal *sd*-shell interaction version B (USDB) [19], which, along with its sister interaction USDA, are the current "gold standards" for empirical *sd*-shell calculations. The present study seeks to extend this model by computing theoretical uncertainties on model parameters and shell-model observables [110, 108]. While the parameter vector $\boldsymbol{\lambda}$ is formally considered a random variable, note that all calculations are performed about the USDB values.

An important nuance in using the USDB parameters is that while the single-particle energies are fixed, the two-body matrix elements are scaled by $(A_0/A)^{0.3}$, where A is the mass number of the nucleus, and A_0 is a reference value, here = 18. We account for this by modifying (4.1) as $\hat{\mathcal{H}} = \sum_i \lambda_i (A_0/A)^{0.3} \hat{\mathcal{O}}_i$ (but only for the two-body matrix elements), so that we implicitly varied the parameters fixed at $A = 18$.

Experimental energies in this paper are the same used in the original fit of the USDB Hamiltonian: absolute energies, relative to the ^{16}O core and with Coulomb differences subtracted, of 608 states in 77 nuclei with A = 21 - 40. The data excludes any experimental uncertainties greater than 200 keV, and most are smaller, on the order of 10 keV.

In the rest of this paper, we estimate the uncertainty in the USDB parameters and, from those, estimate uncertainties in observables such as energies, probabilities for selected electromagnetic and weak transitions, and for a matrix element relevant to dark matter direct detection.

4.3 Sensitivity analysis of a nuclear interaction

Our development above is cast in terms of standard sensitivity analysis [82, 88, 103, 28]. To connect with more sophisticated UQ analyses, and to set the stage for future work, we provide a broader, Bayesian context.

To define uncertainty on the USDB parameters, we start with Bayes' theorem. Let D represent data and λ the parameters, then

$$P(\boldsymbol{\lambda}|D) = \frac{P(D|\boldsymbol{\lambda})P(\boldsymbol{\lambda})}{P(D)} \propto P(D|\boldsymbol{\lambda})P(\boldsymbol{\lambda}) \tag{4.2}$$

Bayes' theorem states that the distribution of model parameters given the experimental data (the *posterior* $= P(\boldsymbol{\lambda}|D)$) is proportional to the distribution of data (the *likelihood* $= P(D|\boldsymbol{\lambda})$) given the parameter set, multiplied by the *a priori* distribution of parameters (the *prior* $= P(\boldsymbol{\lambda})$). Bayesian analysis [119] demands that we put some thought into the choice of prior, and a common choice here is a *non-informative* prior, which seeks to minimize the effects of prior knowledge on the posterior distribution. In this case a non-informative prior can simply be uniform and very broad in the limiting case, $P(\boldsymbol{\lambda}) = \text{constant}$ everywhere. This assigns equal probability to all parameter values (the principle of indifference [119]). Although one could also justify using an informative prior, the flat prior it is a sensible first approximation for the scope of this analysis.

With the prior set to constant, Bayes' theorem reduces to:

$$P(\boldsymbol{\lambda}|D) \propto P(D|\boldsymbol{\lambda}) \tag{4.3}$$

The goal now is to evaluate this expression, and we can choose between two methods: Laplace's Approximation (LA), or Markov-Chain Monte Carlo (MCMC). Due to its simplicity, we choose LA, as did a prior shell model study [139]. While MCMC advantageously makes no assumption as to the form of $P(\boldsymbol{\lambda}|D)$, it typically converges slowly for posteriors which are steep around extrema, so the computational cost of LA is comparatively much less.

Laplace's approximation is a second-order Taylor approximation in the log-likelihood, and thus we assume normally distributed errors on energies. Our likelihood function takes the form:

$$P(D|\boldsymbol{\lambda}) = \exp\left[-\frac{1}{2}\chi^2(\boldsymbol{\lambda})\right] \tag{4.4}$$

where χ^2 is the usual sum of squared residuals:

$$\chi^2(\boldsymbol{\lambda}) = \sum_{\alpha=1}^{N}\left(\frac{E_\alpha^{SM}(\boldsymbol{\lambda}) - E_\alpha^{exp}}{\Delta E_\alpha}\right)^2 \tag{4.5}$$

In addition E_α^{exp} is the experimental excitation energy given in the data set and $E_\alpha^{SM}(\boldsymbol{\lambda})$ is the shell model calculation for that energy using the parameters $\boldsymbol{\lambda}$. The total uncertainty ΔE_α on the residual is expressed as experimental uncertainty ΔE_α^{exp} and some *a priori* theoretical uncertainty ΔE^{th} added in quadrature:

$$\Delta E_\alpha^2 = (\Delta E^{th})^2 + (\Delta E_\alpha^{exp})^2 \tag{4.6}$$

Here we introduce ΔE^{th} as an estimated uncertainty on the shell-model predictions of the data. We assume it is independent of the level, that is, of α, and fix it by requiring the reduced sum of squared residuals $\chi^2_\nu = \frac{1}{\nu}\chi^2 \approx 1$ [88], which gives us $\Delta E^{th} \approx 150$ keV. Here ν is the number of degrees of freedom: the number of data points minus the number of parameters. In their original paper, Brown and Richter set σ^{th} (equivalent to our ΔE^{th}) = 0.1 MeV as "close to the rms value" they eventually found, 126 keV [19].

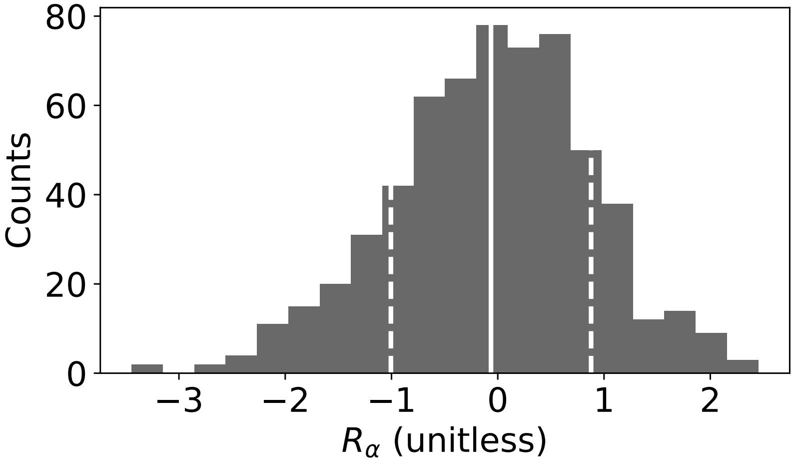


Figure 4.1. Histogram of energy residuals $R_\alpha = (E^{SM}_\alpha(\boldsymbol{\lambda}_{\mathbf{USDB}}) - E^{exp}_\alpha)/\Delta E_\alpha$**.**

Before proceeding with the sensitivity analysis, it is important to test the distribution of residuals $R_\alpha = (E^{SM}_\alpha(\boldsymbol{\lambda}_{\text{USDB}}) - E^{exp}_\alpha)/\Delta E_\alpha$, shown in Fig. 4.1, since we will approximate it to be normally distributed (equivalent to Laplace's approximation). We employ two statistical tests of normality: Kolmogorov-Smirnov [1] (KS-test) and tail-sensitive [93, 6] (TS-test); the former is a typical test of overall normality, while the latter is more sensitive to features in the tails of the distribution. Each test returns a p-value: we adopt the traditional significance threshold of $p > 0.05$ as no significant evidence for deviations from the standard normal distribution. This is sometimes colloquially referred as agreement between the empirical and theoretical distributions. To visualize these tests of normality, we show a rotated quantile-quantile (Q-Q) plot of the residuals R_α in Fig. 4.2. The residuals appear to have a nearly normal distribution, and indeed the KS-test returns a p-value of 0.15. This validates our implementation of Laplace's approximation. However, the more sensitive TS-test returns a p-value of 0.02, indicating that the tails of the residual distribution contain sufficient non-normal features as to warrant a more detailed study in future work.

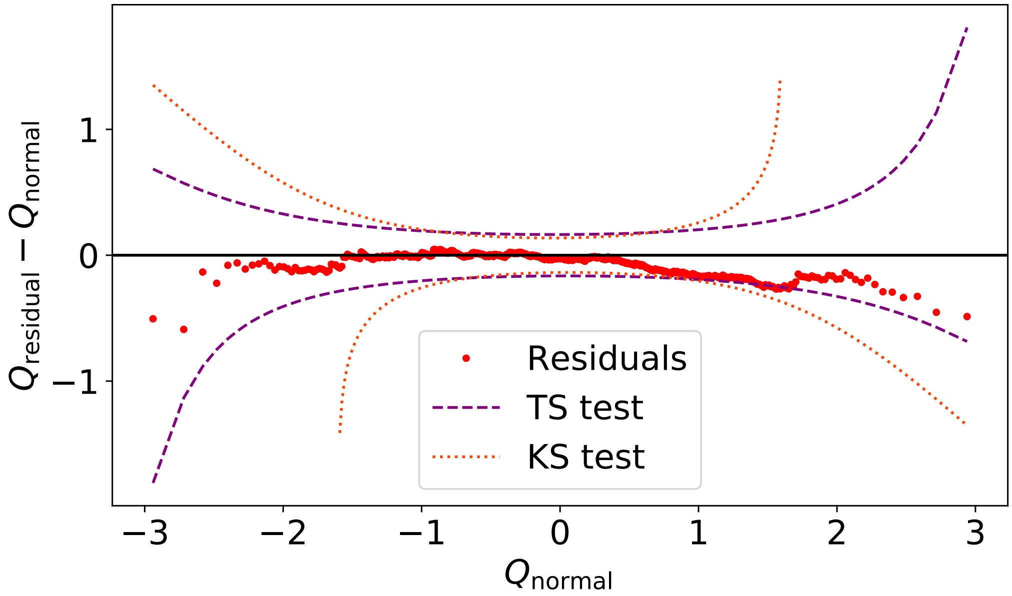


Figure 4.2. Rotated quantile-quantile (Q-Q) plot of energy residuals $(E_\alpha^{SM}(\boldsymbol{\lambda}_{\mathbf{USDB}}) - E_\alpha^{exp})/\Delta E_\alpha$ with respect to standard normal distribution. The dashed and dotted lines in the Q-Q plot show the boundaries of TS and KS-tests respectively. Deviation from the horizontal axis indicates non-normal deviations in the data. The residual points crossing the dashed purple line around $Q_{\mathbf{normal}} \approx 1.5$ corresponds to the low p-value returned by the TS-test.

The quantile-quantile (Q-Q) plot [2] is a useful tool for visualizing how well the distribution of a data set matches that of a random variate from a known probability distribution. Our *rotated* Q-Q plot in Fig. 4.2 shows the comparison of energy residuals to a standard normal distribution. A typical Q-Q plot graphs N measured data points $\{x_i^{\text{data}}\}$, sorted from lowest to highest, against N uniformly distributed evaluations $\{x_i^{\text{eval}}\}$ of the *quantile function* (sometimes called a *percent-point function*) of the distribution we wish to compare to. For a random variable X with cumulative distribution function (CDF) $F_X(x) \equiv \Pr(X \leq x)$, the quantile function $Q_X(p)$ returns the value of x such that $F_X(x) = p$; in other words, it is the inverse function of the CDF. For instance if the set of data points follows a normal distribution, that is, $\{x_i^{\text{eval}}\} = \{x_i^{\text{normal}}\}$ then the points $(x_i^{\text{data}}, x_i^{\text{normal}})$ for $i = 1, 2, \ldots, N$ will fall on a straight line with slope of 1. If the data does not follow a normal distribution, then the points will deviate from a straight line, displaying how non-normal the data is. Our Q-Q plot in Fig. 4.2 in this paper has been "rotated" by plotting instead $(x_i^{\text{data}} - x_i^{\text{normal}}, x_i^{\text{normal}})$, where x^{data} are the energy residuals, so that a normal distribution would lie on the horizontal axis at zero. This allows for an easier identification of discrepancies between empirical and theoretical quantiles via visual inspection.

Many statistical tests exist for determining normality of data, and often these can be represented as a curve on the Q-Q plot. The Kolmogorov-Smirnov and tail-sensitive tests used in this work correspond to curves shown in Fig. 4.2; evidence of possible non-normality of the data is indicated by the plotted quantile-quantile points crossing over these curves.

Under the assumption the errors have a normal distribution , χ^2 is well-approximated by quadratic function in $\boldsymbol{\lambda}$, and we can compute the Hessian H, or the second derivative of χ^2, that is,

$$H_{ij} = \frac{1}{2} \frac{\partial^2}{\partial \lambda_i \partial \lambda_j} \chi^2. \tag{4.7}$$

Note that we write the Hessian matrix as H, and the Hamiltonian operator as $\hat{\mathcal{H}}$. We can simplify this expression to put it in terms of energies:

$$\frac{\partial^2 \chi^2}{\partial \lambda_i \partial \lambda_j} = \sum_{\alpha=1}^{N} \frac{2}{(\Delta E_\alpha)^2} \left[\frac{\partial E_\alpha^{SM}}{\partial \lambda_i} \frac{\partial E_\alpha^{SM}}{\partial \lambda_j} + (E_\alpha^{SM} - E_\alpha^{expt}) \frac{\partial^2 E_\alpha^{SM}}{\partial \lambda_i \partial \lambda_j} \right] \tag{4.8}$$

so that

$$H_{ij} = \sum_{\alpha=1}^{N} \frac{1}{(\Delta E_\alpha)^2} \frac{\partial E_\alpha^{SM}}{\partial \lambda_i} \frac{\partial E_\alpha^{SM}}{\partial \lambda_j} + \sum_{\alpha=1}^{N} \frac{(E_\alpha^{SM} - E_\alpha^{expt})}{(\Delta E_\alpha)^2} \frac{\partial^2 E_\alpha^{SM}}{\partial \lambda_i \partial \lambda_j} \tag{4.9}$$

The first term in this expression dominates, so we define the *approximate Hessian* A as follows:

$$H_{ij} \approx \sum_{\alpha=1}^{N} \frac{1}{(\Delta E_\alpha)^2} \frac{\partial E_\alpha^{SM}}{\partial \lambda_i} \frac{\partial E_\alpha^{SM}}{\partial \lambda_j} \equiv A_{ij} \tag{4.10}$$

This approximation is good if the cross-derivative is small, for example if the energies were exactly linear in the parameters, or alternatively if the residual is small (meaning the model is good). Furthermore, the calculation of E_α^{SM} is made with the optimized USDB parameters, therefore the term multiplying the cross-derivative should on average be close to zero. The second term contains the cross-derivative, and this is more challenging to calculate, especially considering the size of the parameter-space.

Note that the energy matrix element is nonlinear in $\boldsymbol{\lambda}$ due to dependence in the wavefunction. If one were to ignore this dependence, we call this the *linear model approximation*

$$E = \langle \psi(\boldsymbol{\lambda}) | \hat{\mathcal{H}}(\boldsymbol{\lambda}) | \psi(\boldsymbol{\lambda}) \rangle \approx \langle \psi | \hat{\mathcal{H}}(\boldsymbol{\lambda}) | \psi \rangle = \sum_{i=1} \lambda_i \langle \psi | \hat{\mathcal{O}}_i | \psi \rangle \tag{4.11}$$

Under the linear model approximation, any cross-derivative term is zero and thus the 'approximate' Hessian above would be equal to the full Hessian: $A = H$.

To compute the derivatives of the energies, in Eq.4.10, we use the Feynman-Hellmann theorem,

$$\frac{\partial E_\alpha^{SM}(\boldsymbol{\lambda})}{\partial \lambda_i} = \left\langle \psi_\alpha \left| \frac{d\hat{\mathcal{H}}}{d\lambda_i} \right| \psi_\alpha \right\rangle = \langle \psi_\alpha | \hat{\mathcal{O}}_i | \psi_\alpha \rangle, \tag{4.12}$$

where the Hamiltonian (4.1) is linear in λ_i. (These first derivatives are Jacobians [28].) Thus, for the first derivatives in (4.10), we can simply evaluate expectation values of the individual 1- and 2-body operators.

While the full numerical calculation of the Hessian is quite costly, we can numerically compute the cross-derivative term in Eq. 4.9 with a simple finite difference approximation of the second derivative, so as to achieve a better approximation to the exact Hessian.

i	$[\Delta]_{ii}$ (MeV^{-2})	σ_i (keV)	$[\Delta_{\text{num}}]_{ii}$ (MeV^{-2})	$[\sigma_{\text{num}}]_i$ (keV)
1	11785000	0.29	11785500	0.29
2	393000	1.6	393600	1.6
3	79100	3.5	78810	3.5
4	71200	3.7	70800	3.7
5	22200	6.7	22220	6.7
6	6357	13	6357	13
7	5200	14	5175	14
8	3600	17	3590	17
9	3270	17	3261	17
10	3050	18	3035	18
...	...	...	...	...
64	10.6	307	< 1	> 1000
65	7.71	360	< 0.1	> 3000
66	3.16	562	< 0.1	> 3000

Table 4.1. Statistics of linear-combinations of USDB matrix elements, or *principle component analysis* (PCA) parameters. The eigenvalues of the approximate Hessian matrix A we denote as $[\Delta]_{ii}$, which is the sensitivity of the ith PCA parameter, and σ_i is the corresponding uncertainty. Thus the most sensitive PCA parameter is constrained to within 290 eV. Likewise, the eigenvalues of the numerically corrected approximate Hessian matrix A_{num} we denote as $[\Delta_{\text{num}}]_{ii}$, and $[\sigma_{\text{num}}]_i$ is the corresponding uncertainty. Note that for the most sensitive PCA parameters, the numerical correction effectively leaves the standard deviations unchanged.

$$\frac{\partial^2 E_\alpha^{SM}}{\partial\lambda_i\partial\lambda_j} = \frac{1}{2\epsilon}\left[\frac{\partial E_\alpha^{SM}(\boldsymbol{\lambda}_j^+)}{\partial\lambda_i} - \frac{\partial E_\alpha^{SM}(\boldsymbol{\lambda}_j^-)}{\partial\lambda_i}\right] + \mathcal{O}(\epsilon^2) \tag{4.13}$$

Here, $E_\alpha^{SM}(\boldsymbol{\lambda}_j^\pm)$ is the α-th energy evaluated using USDB parameters with the j-th value perturbed by $\pm\epsilon$. Inserting into Eq. 4.9, we denote the resulting numerically corrected approximate Hessian matrix as A_{num}.

$$[A_{\text{num}}]_{ij} \equiv [A]_{ij} + \sum_{\alpha=1}^{N}\frac{(E_\alpha^{SM}(\boldsymbol{\lambda}) - E_\alpha^{expt})}{(\Delta E_\alpha)^2}\frac{1}{2\epsilon}\left[\frac{\partial E_\alpha^{SM}(\boldsymbol{\lambda}_j^+)}{\partial\lambda_i} - \frac{\partial E_\alpha^{SM}(\boldsymbol{\lambda}_j^-)}{\partial\lambda_i}\right] \tag{4.14}$$

We tested their importance by evaluating with $\epsilon \approx 0.1$. The the resulting eigenvalues of A and A_{num}, shown in Table 4.1, are very similar, indicating that while the numerical corrections terms are individually nonzero, the total contributions average to very small contributions. Thus A is in fact a very good approximation to the full Hessian matrix and, in what follows, we find that propagation of uncertainties onto observables are almost independent of the numerical correction. This also implies that the linear model approximation (Eq. 4.11) is a good approximation.

4.3.1 PCA Transformation

Transforming the Hessian $UHU^T = D$, where D is diagonal, or its approximation

$$WAW^T = \Delta \tag{4.15}$$

where $\Delta \approx D$ is also diagonal, provides a transformation from the original parameters $\boldsymbol{\lambda}$ to new linear combinations of parameters,

$$\boldsymbol{\mu} = W\boldsymbol{\lambda}. \tag{4.16}$$

This is simply *principal component analysis* (PCA) of the Hessian, and so we call $\boldsymbol{\mu}$ the PCA parameters. In terms of our approximate Hessian, we can also understand this as a singular value decomposition (SVD) of the linear response $J_{\alpha i} = \partial E_\alpha / \partial \lambda_i$. More formally, we approximate $H \approx A = J^T \Sigma^{-2} J$, where Σ is the diagonal matrix of uncertainties on energies, $\Sigma_{\alpha\beta} = \delta_{\alpha\beta} \Delta E_\alpha$; but, as is nearly true, $\Delta E_\alpha \approx \Delta E^{th}$ and hence $A \approx (\Delta E^{th})^{-2} J^T J$; then it should be clear that the eigenvalues of A are proportional to the SVD eigenvalues of J. Thus the eigenvalues found in Δ, presented in Table 4.1 and plotted in Fig. 4.3, allow us to determine the most important linear combinations of parameters to the fit.

4.4 Evaluating uncertainties

The parameter covariance matrix is simply the inverse of the Hessian matrix, which we have approximated as

$$C(\boldsymbol{\lambda}) = H^{-1} \approx A^{-1} = W^T \Delta^{-1} W \tag{4.17}$$

The naive variance of the original parameters λ is given by the diagonals of the covariance matrix, so that $\sigma(\lambda_i) = \sqrt{C_{ii}}$. This, however, ignores correlations between

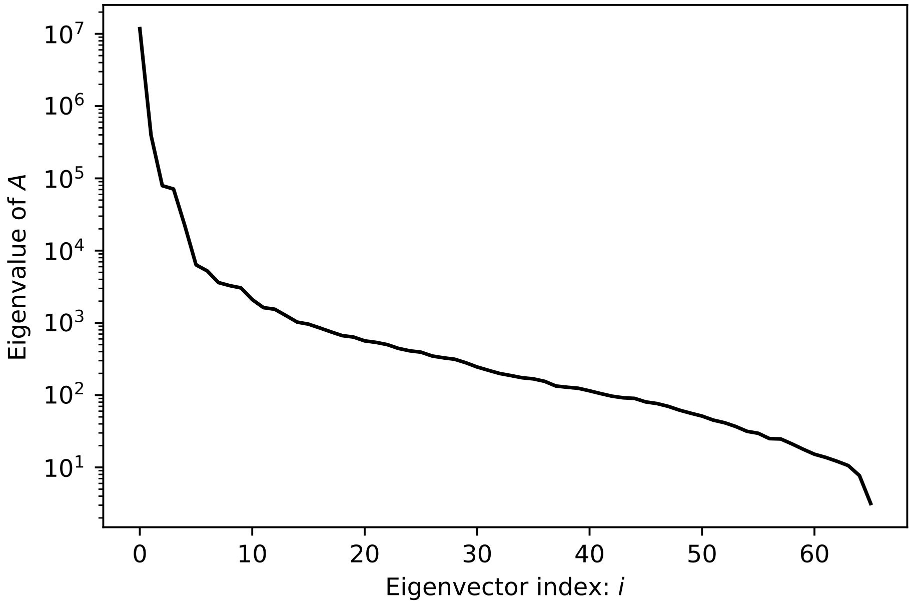


Figure 4.3. Ordered eigenvalues of the approximate Hessian A, which equal the diagonal elements of Δ. The eigenvalues are interpreted as the sensitivity of the corresponding linear combination or principal components of matrix elements (PCA-parameter). The first PCA-parameter carries 95% of the total sensitivity, and the first 5 PCA-parameters carry 99.6% of the sensitivity.

parameters and thus is an incomplete description of parameter uncertainties. A better approach is to compute variances from the diagonalized Hessian matrix, and thus obtaining uncorrelated uncertainties on the PCA parameters, $\sigma(\boldsymbol{\mu}) = 1/\sqrt{\Delta_{ii}}$. These we give in Table 4.1, and plot in Fig. 4.3. Here one sees the first few PCA parameters have very large sensitivity, and indeed the first 10 carry over 99.8% of the total; it is well-known lore in the nuclear shell-model community the fit of USDB and similar empirical interactions are dominated by only a few linear combinations, which here define the PCA parameters. Table 4.1 in fact demonstrates these parameters must be known to within a few keV or better; on the other hand 23 PCA parameters have uncertainties of 100-500 keV. At this point, it is important to remember that these variablities are with respect to experimental data that only includes energies, so these low-variability PCA parameters could in principle be tuned to fit the interaction to various other observables without disrupting the fit to energies.

If the uncertainties in the principal components $\boldsymbol{\mu}$ are independent, then the propagation of uncertainties is straightforward. For any observable X,

$$\sigma^2(X) = \sum_i \left(\frac{\partial X}{\partial \mu_i}\right)^2 \sigma^2(\mu_i) \tag{4.18}$$

Using (4.16),

$$\frac{\partial X}{\partial \mu_i} = \sum_j W_{ij} \frac{\partial X}{\partial \lambda_j} \tag{4.19}$$

and so

$$\sigma^2(X) = \sum_i \sigma^2(\mu_i) \sum_{jk} \frac{\partial X}{\partial \lambda_j} W_{ij} \frac{\partial X}{\partial \lambda_k} W_{ik} = \boldsymbol{g}^T C \boldsymbol{g} \tag{4.20}$$

where $g_i = \partial X / \partial \lambda_i$ is the linear response of any observable to perturbations in the original parameters. This is particularly useful in the case of energies, where we already have the linear response, thanks to the Feynman-Hellmann theorem.

4.4.1 Computed covariance of fitted energies

Here we show that computing the covariance matrix of fit energies C_E by Eq. 4.30 is simply related to a similarity transform of the original uncertainties on fit energies given by Eq. 4.6: $\Sigma_{\alpha\alpha} = \Delta E_\alpha$. The response of the energies to changes in the parameters is an $N_d \times N_p$ Jacobian matrix, $J_{\alpha i} = \partial E_\alpha / \partial \lambda_i$, where N_d is the number of data points and N_p is the number of parameters. The approximate Hessian is

$$A = J^T \Sigma^{-2} J, \tag{4.21}$$

and the parameter covariance is

$$C_\lambda = A^{-1} = (J^T \Sigma^{-2} J)^{-1}. \tag{4.22}$$

Since J is not square, we cannot evaluate this expression in terms of matrix inversion and instead use the pseudoinverse obtained by SVD decomposition. We get the factorization $J = USV^T$ where U is a $N_d \times N_d$ unitary matrix, S is a $N_d \times N_p$ matrix with the only non-zero elements being N_p singular values along the diagonal, and V is a $N_p \times N_p$ unitary matrix. We use this to define a new matrix J^+ which is the pseudoinverse of J.

$$J^+ = VS^+U^T \tag{4.23}$$

Here, S^+ is the pseudoiverse of S, which has the same shape as S^T and the only nonzero elements are such that $S^+_{jj} = 1/S_{jj}$ for $j = 1, 2, ..., N_p$.

Plugging this into the expression for C_λ we have

$$C_\lambda = J^+\Sigma^2[J^T]^+ = (VS^+U^T)\Sigma^2(US^+V^T), \tag{4.24}$$

In turn we insert this into our expression for C_E:

$$C_E = (USV^T)(VS^+U^T)\Sigma^2(US^+V^T)(VSU^T). \tag{4.25}$$

By the orthogonality of U and V we have $U^TU = I_d$ and $U^TU = I_p$, identity matrices in the data-space and parameter-space respectively, so that

$$C_E = USI_pS^+U^T\Sigma^2US^+I_pSU^T. \tag{4.26}$$

To simplify further, we need to pay attention the the rank-deficient property of S. Define $SI_pS^+ = P^p_d$ to be a $N_d \times N_d$ square matrix with N_p 1's on the diagonal, starting from the top, and all zeros otherwise. (This is projection operator from the data-space into the parameter-space, hence this notation.) Then

$$C_E = UP^p_dU^T\Sigma^2UP^p_dU^T. \tag{4.27}$$

Now, notice that since Σ^2 is diagonal, we have $U^T\Sigma^2U = \Sigma^2$. The matrix P^p_d is of course idempotent so $P^p_dP^p_d = P^p_d$, and we get

$$C_E = U\Sigma^2P^p_dU^T, \tag{4.28}$$

or

$$U^TC_EU = \Sigma^2P^p_d. \tag{4.29}$$

Thus, the computed covariance on the energies C_E is equivalent to a similarity transform of the input uncertainties Σ^2, albeit with rank = N_p.

For other observables, we do not use (4.20) directly. Instead, we generate perturbations in USDB by generating perturbations in the PCA parameters $\delta\boldsymbol{\mu}$ with a Gaussian distribution with width $\sigma(\mu_i)$ given by Table I. Because the uncertainties are independent, or nearly so, in the PCA parameter representation, it is safe to generate the perturbations independently. We then transform back to the original representation of the matrix elements and read into a shell-model code [67, 69], find eigenpairs, and evaluate the reduced transition matrix element for one-body transition operators. We sampled at least $1,000$ sets of parameters, which gives sufficient convergence of the resulting set of matrix elements: assuming the transition strengths B_i are normally distributed with respect to small perturbations in the Hamiltonian, we take the theoretical uncertainty $\sigma(B_i)$ as equal to the standard deviation of the set of samples. Previous works have demonstrated convergence with similar approaches and an even smaller number of samples. In [94] the statistical uncertainty in the binding energy of ^{3}H was quantified using 250 samples of an interaction with about 40 parameters, resulting in $\sigma(B) = 15$ keV. The same result was later reproduced in [95] using only 33 samples.

4.4.2 Results

For the energies used in the fit, we already have the elements of $\vec{g}$ saved from computing the approximate Hessian, so this calculation is cheap. We can thus estimate covariance in the computed energies C_E by expanding this expression to a matrix equation.

$$C_E = JC_\lambda J^T \tag{4.30}$$

Results for some of these estimated uncertainties are given in Table 4.2. Using these estimates, 75% of shell-model energies are within 1σ of experiment, and 96% are

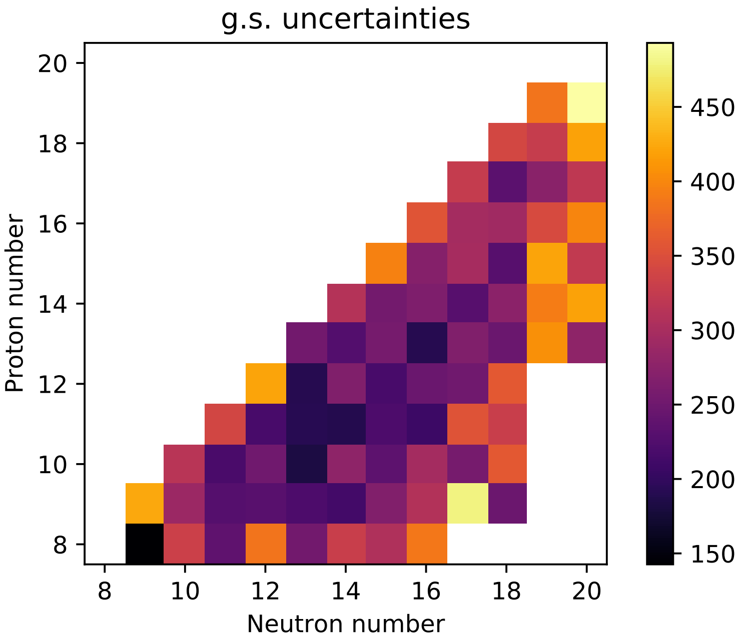


Figure 4.4. Estimated 1σ uncertainties of ground-state fit energies in units of keV.

within 3σ; these are close to the standard normal quantiles of 68% and 99% respectively, so we conclude that these theoretical uncertainties are sensible. Akin to the original sensitivity analysis of fit energies [19], Fig. 4.4 shows theoretical 1σ uncertainties on ground-state binding energies. We refer the reader to [130] for comparison to uncertainty plots, in particular Fig. 10 of that paper. While this description of uncertainties on the fitted energies may be useful, we also note that they are in a sense tautological: the energy covariance C_E is related to the energy uncertainties in Eq. 4.6 by a coordinate transformation.

We also computed the uncertainties in selected transitions. The uncertainty bands presented in all transition strength calculations correspond to the 16th - 84th percentiles; for normal distributions this is precisely the 1σ uncertainty band, but we find many computed transition strengths have asymmetric distributions (especially those with small B-values). This, along with reporting median rather than mean, gives a more accurate description of uncertainty.

Nucleus	J_n^π	T	$E^{exp} - E^{SM}$ (keV)	σ (keV)
^{30}Si	1_1^+	1	-114	851
^{39}K	$1/2_1^+$	1/2	-189	785
^{25}F	$5/2_1^+$	7/2	-312.1	743
^{38}K	1_1^+	0	-355.9	686
^{27}Al	$11/2_1^+$	1/2	-52.9	615
...	...	...	...	...
^{24}Mg	6_1^+	0	156.1	156
^{20}Ne	6_1^+	0	-223.2	154
^{23}Na	$11/2_1^+$	1/2	-15.3	153
^{28}Mg	2_1^+	2	19.3	153
^{17}O	$5/2_1^+$	1/2	218.3	142

Table 4.2. States in experimental energy data, shown in order of descending uncertainty σ (high-variability on top, low-variability on bottom).

Following [108], we compute reduced transition strengths B(E2) for several low-lying transitions in ^{26}Mg and ^{26}Al, shown in Fig. 4.5 and 4.6 respectively. The one-body electric quadrupole operator matrix elements were computed assuming harmonic oscillator radial wave functions with oscillator length $b = 1.802$ [20] and effective charges $e_p = 1.36$, $e_n = 0.45$, which were obtained by a least-squares fit [109]. While some values are close to experiment, others differ significantly. The B(E2) values are quadratically dependent upon both the oscillator length and the effective charges, and can be quite sensitive to small changes in the interaction matrix elements [108].

For ^{26}Mg, in Fig. 4.5, the median values and uncertainty intervals for our selected transitions are $2_1^+ \rightarrow 0_1^+ : 63.7^{+0.78}_{-0.83}$, $2_2^+ \rightarrow 0_1^+ :3.46^{+0.55}_{-0.52}$, $0_2^+ \rightarrow 2_1^+ :1.15^{+0.33}_{-0.29}$, and $2_4^+ \rightarrow 0_1^+ : 0.96^{+0.18}_{-0.18}$, all in units of e^{2}fm^4, while for ^{26}Al, in Fig. 4.6 the median values and uncertainty intervals for our selected transitions are $3_1^+ \rightarrow 5_1^+ : 52.04^{+0.99}_{-1.0}$, $1_2^+ \rightarrow 3_1^+ : 54.47^{+4.19}_{-4.92}$, $2_2^+ \rightarrow 0_1^+ : 56.63^{+1.26}_{-1.16}$, $1_3^+ \rightarrow 3_1^+ : 0.53^{+2.53}_{-0.49}$, $3_2^+ \rightarrow 5_1^+ : 0.017^{+0.041}_{-0.015}$, and $3_3^+ \rightarrow 1_1^+ : 11.38^{+2.82}_{-2.53}$.

Magnetic dipole reduced transition strengths B(M1) distributions for ^{18}F and ^{26}Al are shown in Fig. 4.7 and 4.8 respectively. We used bare gyromagnetic factors, with no corrections for exchange currents. Like the B(E2) values, some of the

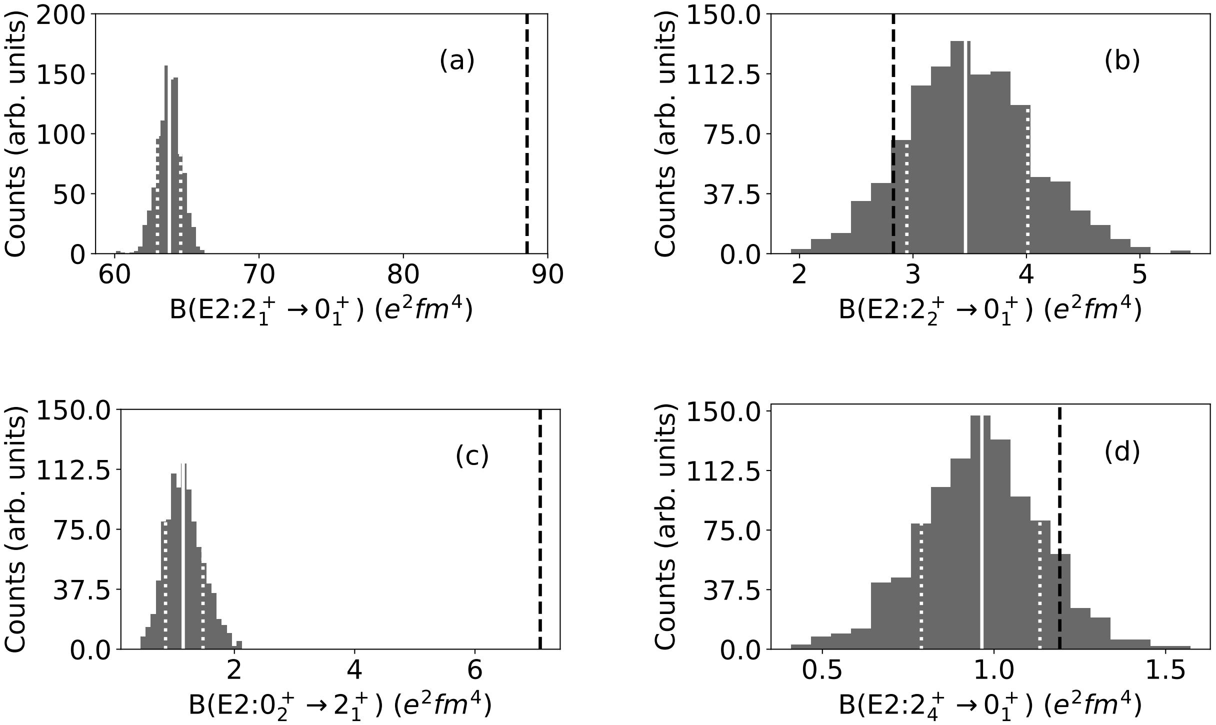


Figure 4.5. Distributions of the electric quadrupole (E2) transition strengths for ^{26}Mg. Black dashed line shows experimental value [129]. The the median values and uncertainty interval are highlighted in white: (a) $2_1^+ \rightarrow 0_1^+ : 63.7^{+0.78}_{-0.83}$, (b) $2_2^+ \rightarrow 0_1^+ : 3.46^{+0.55}_{-0.52}$, (c) $0_2^+ \rightarrow 2_1^+ : 1.15^{+0.33}_{-0.29}$, (d) $2_4^+ \rightarrow 0_1^+ : 0.96^{+0.18}_{-0.18}$, all in units of e^{2}fm^4.

transitions are close to experiment, while the $0_1^+ \rightarrow 1_1^+$ in ^{18}F is quite far away. For ^{18}F, in Fig. 4.7, the median values and uncertainty intervals for our selected transitions are $0_1^+ \rightarrow 1_1^+ : 17.13^{+0.19}_{-0.21}$, $1_2^+ \rightarrow 0_1^+ : 0.31^{+0.076}_{-0.068}$, and $3_2^+ \rightarrow 2_1^+ : 0.57^{+0.087}_{-0.077}$, all in units μ_N^2, where μ_N is the nuclear magneton, while for ^{26}Al, in Fig. 4.8 the median values and uncertainty intervals for our selected transitions are $1_1^+ \rightarrow 0_1^+ : 2.89^{+0.15}_{-0.17}$, $1_2^+ \rightarrow 0_1^+ : 0.55^{+0.18}_{-0.16}$, $1_3^+ \rightarrow 0_1^+ : 0.096^{+0.10}_{-0.07}$, $1_4^+ \rightarrow 0_1^+ : 0.17^{+0.12}_{-0.09}$, and $2_5^+ \rightarrow 1_1^+ : 0.095^{+0.022}_{-0.021}$.

We show Gamow-Teller matrix elements for β^--decays in ^{26}Ne and ^{32}Si in Fig. 4.9 and 4.10 respectively. We have used for the axial-vector coupling constant $g_A/g_V = -1.251$, following [110], and a quenching factor of 0.76 for USDB. For ^{26}Ne, in Fig. 4.9, the median values and uncertainty intervals for our selected transitions are

$0^+_1 \to 1^+_1$: $0.726^{+0.038}_{-0.037}$, $0^+_1 \to 1^+_2$: $0.267^{+0.029}_{-0.030}$,and $0^+_1 \to 1^+_3$: $0.22^{+0.034}_{-0.037}$, all unitless. The ground-state decay of ^{32}Si has a small experimental transition strength, so our sensitivity analysis does not provide a normal distribution for B(GT). Using USDB, our median value and uncertainties are $0.00597^{+0.0071}_{-0.0045}$, but this is quite different than the experimental value is of 0.000038 [130]. This particular transition is very sensitive to the parameters: for the 1985 universal *sd*-shell interaction (USD) interaction [17] we get a value for B(GT) = 0.00005, and if one uses the 2006 universal *sd*-shell interaction version A (USDA), which is a less constrained version of USDB [19], the B(GT) is 0.038.

(Motivated by the non-Gaussian distribution in Fig. 4.10, we increased the number of samples from 1000 to 4000. The results were nearly indistinguishable, with new median value and uncertainties of $0.00624^{+0.0077}_{-0.0047}$.)

One of the biggest questions in physics today is the nature of non-baryonic dark matter [11]. While there are a number of ongoing and planned experiments [114], interpreting experiments, including limits, requires good knowledge of the dark matter-nucleus scattering cross-section, including uncertainties. While historically it was assumed dark matter would couple either to the nucleon density or spin density, more recent work based upon effective field theory showed there should be a large number of low-energy couplings, around 15 [7]. This enlarged landscape of couplings, and the increased need for good theory, is a strong motivation for the current work.

In order to illustrate the application of UQ to nuclear matrix elements for dark matter scattering, Fig. 4.11 shows the uncertainty of an $\vec{l} \cdot \vec{s}$ coupling for ^{36}Ar. ^{36}Ar is a small component (0.3%) of argon dark matter detectors, e.g. [5], but it is within the scope of the current work to compute. Of the EFT operators that do not vanish for a $J^\pi = 0^+$ ground state, most of them depend upon radial wave functions that do not play a role in fitting the USDB parameters; nontrivial operators, however, include $\vec{l} \cdot \vec{s}$, which arises in the long-wavelength (momentum transfer $q \to 0$) limit of the nuclear matrix elements of the operators $\mathcal{O}_{3,12,15}$ [7]

$$\mathcal{O}_3 = i\vec{S}_N \cdot \left(\frac{\vec{q}}{m_N} \times \vec{v}^{\perp} \right),$$
$$\mathcal{O}_{12} = \vec{S}_\chi \cdot \left(\vec{S}_N \times \vec{v}^{\perp} \right),$$
$$\mathcal{O}_{15} = -\left(\vec{S}_\chi \cdot \frac{\vec{q}}{m_N} \right) \left(\left(\vec{S}_N \times \vec{v}^{\perp} \right) \cdot \frac{\vec{q}}{m_N} \right),$$

where m_N is the nucleon mass, $\vec{q}$ is the momentum transfer, $\vec{S}_{N/\chi}$ are the spins of the nucleon/WIMP, and $\vec{v}^{\perp}$ is the component of the nucleon-WIMP relative velocity perpendicular to $\vec{q}$. We chose to study $\langle \vec{l} \cdot \vec{s} \rangle$ for the simple reason of best illustrating a variance due to uncertainty in the USDB parameters. The variance of this particular operator is relatively small, but in larger model spaces there could be greater uncertainty. Knowledge of the variance of the operator is important for interpreting experiments, such as placing upper limits on dark matter-nucleon couplings.

4.5 Conclusions

We have carried out uncertainty quantification of a ‘gold-standard’ empirical interaction for nuclear configuration-interaction calculations in the *sd*-shell valence. Rather than finding the uncertainty in each parameter independently [139], we computed the linear sensitivity of the energies, which is easy to compute using the Feynman-Hellmann theorem, and then constructed an approximate Hessian which we then diagonalized. This is equivalent to a singular-value decomposition of the linear sensitivity, and is also known as principle component analysis. We found evidence this is a good approximation to the full Hessian. From the inverse of the diagonal (in a basis of the PCA linear combination of parameters) approximate Hessian, we obtained approximately independent uncertainties in the PCA parameters. Then, starting from those uncertainties, we generated uncertainties for energies as well as several observables. The distribution of residuals in energies implies statistical agreement, as well as an underlying systematic uncertainty in the shell model of 150 keV. For electromagnetic and weak transitions, which we note are sensitive to effective parameters such as effective charges and assumed oscillator length parameters, our residuals relative to experiment included both good agreement as well as residuals with statistically significant deviations. We also presented as a test case a dark matter-nucleus interaction matrix element and our derived uncertainty.

In future work, in addition to further and more systematic study of observables, we will carry out a more detailed and thorough study of parameter covariances, as well as applying our methods to other empirical interactions in other model spaces. This will entail evaluating the posterior without Laplace's approximation, and instead using Markov-chain Monte Carlo sampling. We are investigating the use of eigenvector continuation [44, 35, 75] to explore parameter space efficiently. For the time being, however, it seems that this approximate Hessian is a good approximation. This is not surprising, but it is useful. Nonetheless, moving to larger spaces, which grow exponentially in dimensions and compute time, will be challenging. New technologies still in development, such as quantum computing may make possible better and more rigorous uncertainty quantification.

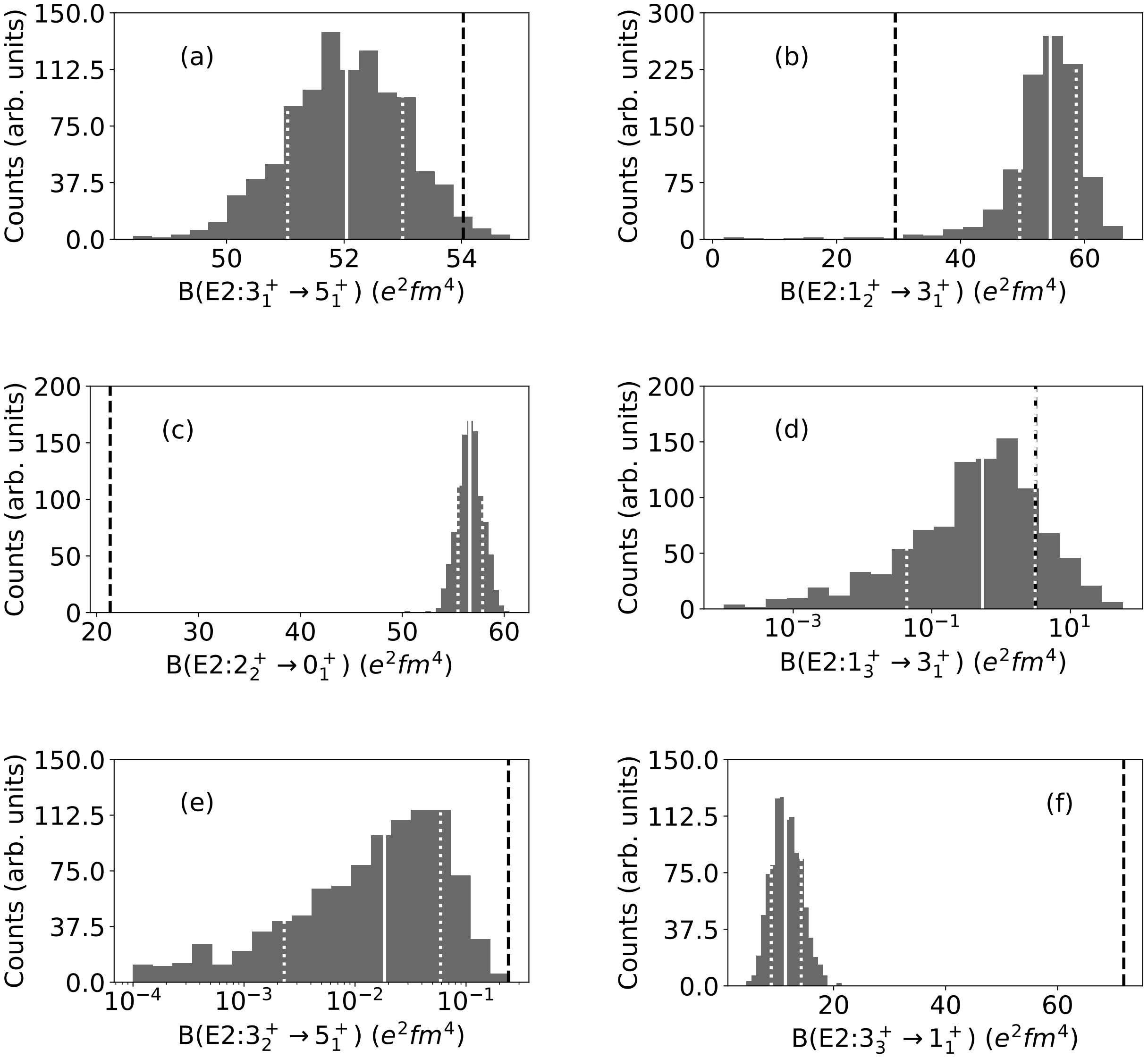


Figure 4.6. Distributions of the electric quadrupole (E2) transition strengths for ^{26}Al. Black dashed line shows experimental value [129]. The median values and uncertainty intervals are highlighted in white : (a) $3_1^+ \rightarrow 5_1^+ : 52.04^{+0.99}_{-1.0}$, (b) $1_2^+ \rightarrow 3_1^+ : 54.47^{+4.19}_{-4.92}$, (c) $2_2^+ \rightarrow 0_1^+ : 56.63^{+1.26}_{-1.16}$, (d) $1_3^+ \rightarrow 3_1^+ : 0.53^{+2.53}_{-0.49}$, (e) $3_2^+ \rightarrow 5_1^+ : 0.017^{+0.041}_{-0.015}$, and (f) $3_3^+ \rightarrow 1_1^+ : 11.38^{+2.82}_{-2.53}$, all in units of e^{2}fm^4 .

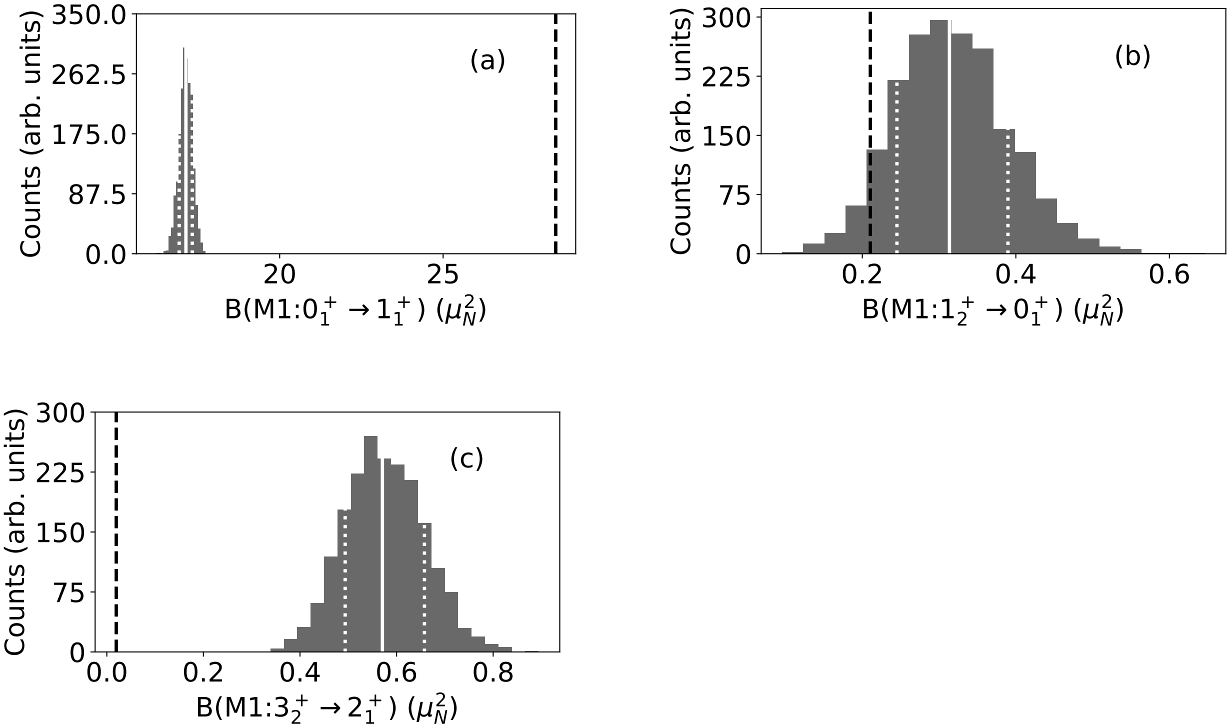


Figure 4.7. Distributions of the magnetic dipole transition strengths for ^{18}F. Black dashed line shows experimental value [129]. The uncertainty interval is highlighted in white: (a) $0_1^+ \rightarrow 1_1^+$: $17.13^{+0.19}_{-0.21}$, (b) $1_2^+ \rightarrow 0_1^+$: $0.31^{+0.076}_{-0.068}$, and (c) $3_2^+ \rightarrow 2_1^+$: $0.57^{+0.087}_{-0.077}$, all in units μ_N^2.

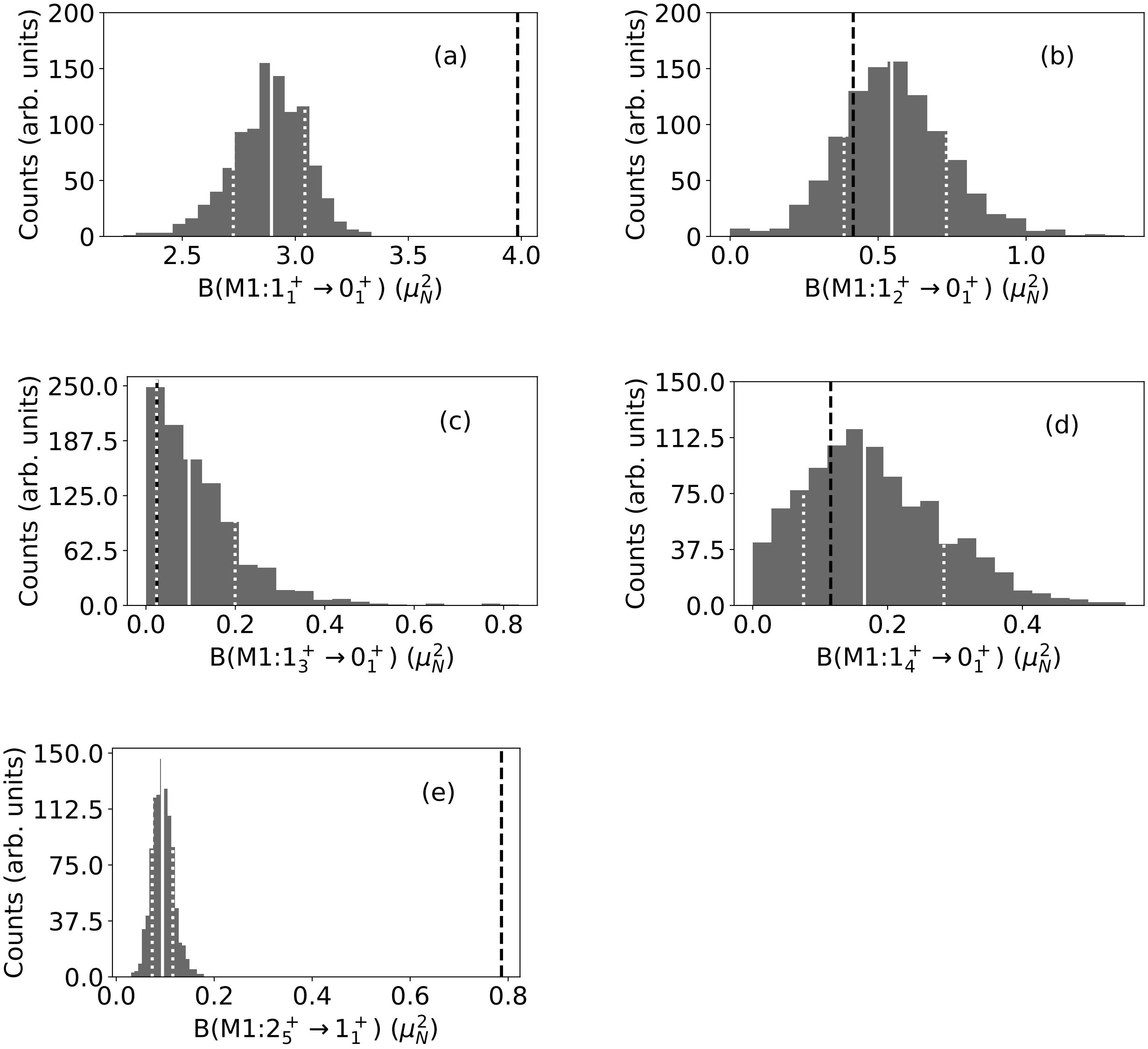


Figure 4.8. Distributions of the magnetic dipole transition strengths for ^{26}Al. Black dashed line shows experimental value [129]. The uncertainty interval is highlighted in white: (a) $1_1^+ \rightarrow 0_1^+ : 2.89^{+0.15}_{-0.17}$, (b) $1_2^+ \rightarrow 0_1^+ : 0.55^{+0.18}_{-0.16}$, (c) $1_3^+ \rightarrow 0_1^+ : 0.096^{+0.10}_{-0.07}$, (d) $1_4^+ \rightarrow 0_1^+ : 0.17^{+0.12}_{-0.09}$, and (e) $2_5^+ \rightarrow 1_1^+ : 0.095^{+0.022}_{-0.021}$, all in units μ_N^2.

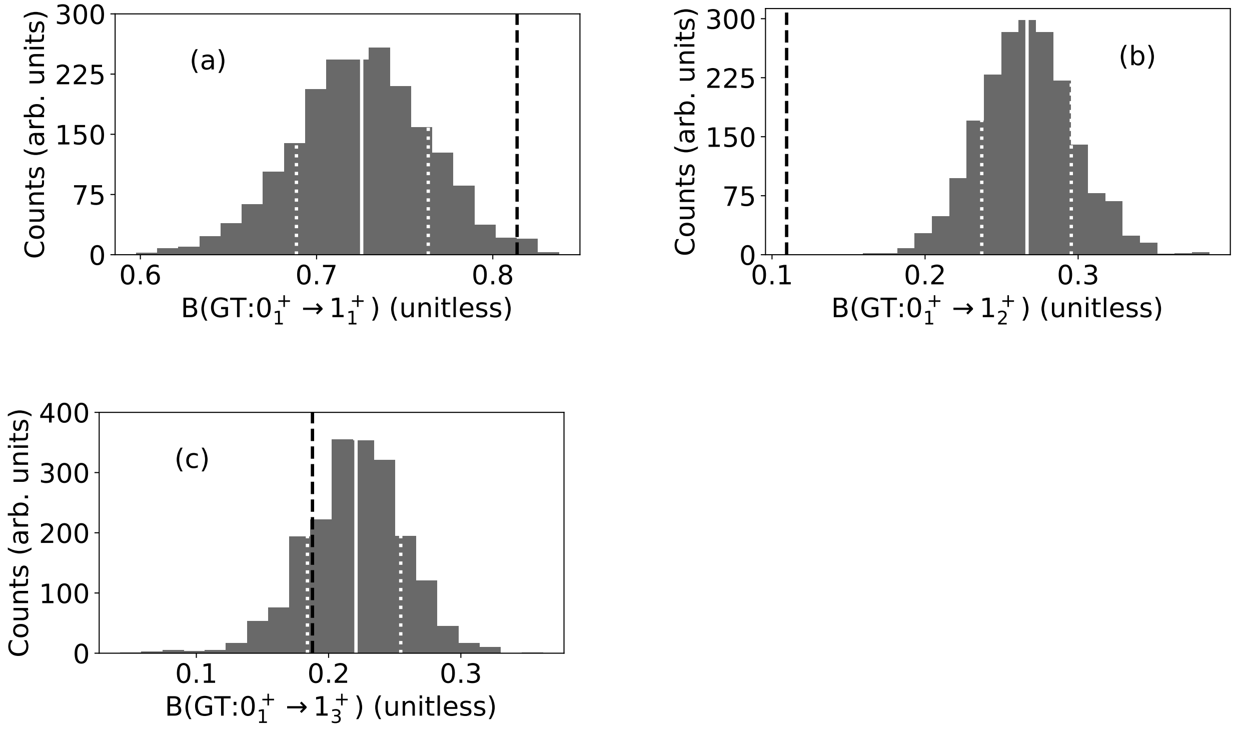


Figure 4.9. Distributions of the Gamow-Teller (GT) transition strengths for β^--decay of ^{26}Ne to ^{26}Na. Black dashed line shows experimental value [130]. The uncertainty interval is highlighted in white: (a) $0_1^+ \rightarrow 1_1^+ : 0.726^{+0.038}_{-0.037}$, (b) $0_1^+ \rightarrow 1_2^+ : 0.267^{+0.029}_{-0.030}$,and (c) $0_1^+ \rightarrow 1_3^+ : 0.22^{+0.034}_{-0.037}$.

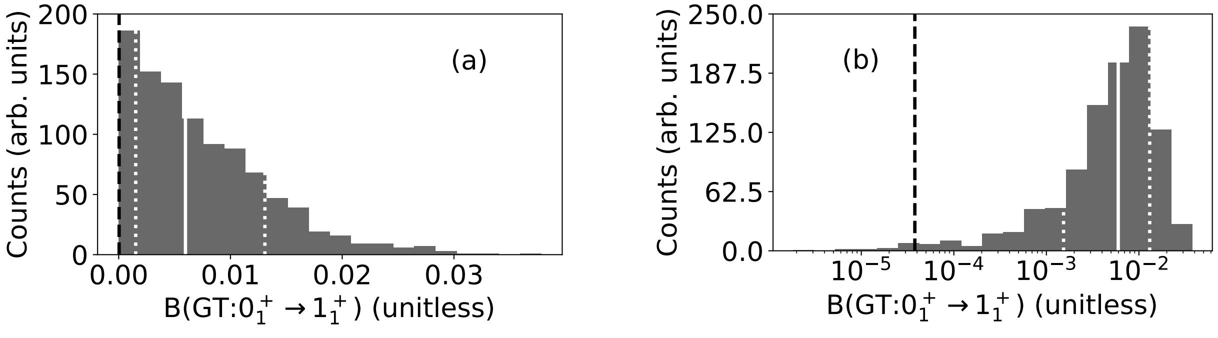


Figure 4.10. Distribution of the Gamow-Teller (GT) transition strength for β^--decay of ^{32}Si to ^{32}P ($0_1^+ \rightarrow 1_1^+$). The left plot is a linear scale in B(GT) and the right is log-scale. Black dashed line shows experimental value of 0.000038 [130]. The uncertainty interval is highlighted in white: $0.00597^{+0.0071}_{-0.0045}$.

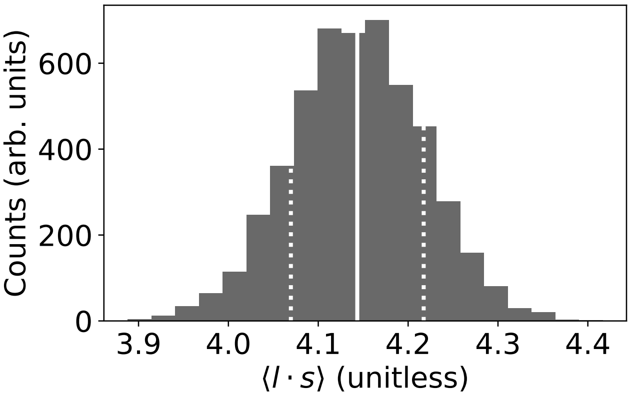


Figure 4.11. Distribution of $\langle l \cdot s \rangle$ in the ground-state of ^{36}Ar. The 1σ interval is highlighted in white: 4.143 ± 0.074 .

CHAPTER 5

PROJECT 1B: UNCERTAINTY QUANTIFICATION OF TRANSITION OPERATORS IN THE SHELL MODEL

Submitted to Phys. Rev. C [43] and available on arXiv.

5.1 Introduction

Over the last two decades, interest in uncertainty quantification (UQ) has grown rapidly in many sciences, and theoretical nuclear physics is no exception [28, 93, 135, 80, 102, 134, 30, 101, 64, 139]. Providing theoretical predictions with error bars that reflect the true limits in our knowledge of a physical system allows for meaningful comparisons between different theoretical models. Furthermore, UQ can be considered a driver of the experiment-theory feedback loop. Proper UQ on the theoretical side can be used to identify the experimental measurements that will have the largest impact in reducing such uncertainty. New measurements, with their own error bars, can then be used to improve current theoretical models. This relationship between theory and experiment becomes particularly relevant in the context of upcoming flagship experimental programs like those at the Facility for Rare Isotope Beams (FRIB) [8], the Deep Underground Neutrino Experiment (DUNE) [4], as well as dark matter searches [125] and neutrinoless double-beta decay experiments [36, 29].

The UQ endeavor includes not only computing error bars for theoretical predictions, but also studying correlations between our variables, whether those be solutions (wavefunctions), coupling constants, observables, etc.. Bayesian statistical analyses [119], which are a natural fit for probabilistic modeling of theory, have become popular in the nuclear science community. Significant advances in computational techniques and tools like Markov Chain Monte Carlo [16], Hamiltonian MC [31], NUTS [62], and the `emcee` library [41] can help achieve efficient and accurate evaluation of probability distributions. Furthermore, the development of emulator models (eigenvector continuation [45, 76], Gaussian processes [107], and even neural networks [55, 99]) allows researchers to study perturbations to complicated models, trading some accuracy for far less computation.

In this work we apply UQ techniques to nuclear shell model calculations to analyze variability of coupling constants with respect to a particular nuclear force model, which in turn requires a statistical description of individual transition matrix elements. In particular we use MCMC and the implementation of the `emcee` library for sampling probability distributions. Future research on shell-model UQ should also investigate other methods, especially emulators for calculations which require significant computational resources. After a brief description of the empirical nuclear shell model in Section 5.2, we present statistical descriptions of model parameters in Section 5.3, and how to construct the distributions of observables and operator parameters in Section 5.4. In Section 5.5 we show results of parameter estimation for transition operator parameters.

5.2 The empirical configuration-interaction shell model

For details, see Section 1.3 on the nuclear shell model.

The Hamiltonian has the usual form,

$$\hat{H} = \sum_i \lambda_i \hat{o}_i, \tag{5.1}$$

where $\hat{o}_i$ is the appropriate operator. For empirical calculations the values of the $\{\lambda_i\}$ are fitted to reproduce some observables, almost always energies.

Here we note an important source of model error. Although the interaction parameters often originally arise from some known translationally-invariant (i.e., relative coordinates) representation, and the matrix elements calculated as integrals using some choice of single-particle wave functions, the matrix elements are directly adjusted in the laboratory frame to fit data [22, 21], thus severing any explicit connection to an underlying potential or Lagrangian. (This differs from *ab initio* calculations, where one may carry out specific transformations of the interaction [14, 60] but never loses the thread back to the "original" interaction.) Thus, arguably, one can no longer even fix with certainty the single-particle basis, specifically the radial part of the wave functions, from which the many-body states are built. This has consequences when computing observables, such as radii or electromagnetic transitions, for which the single-particle wave functions are a key input. A frequent choice in the literature is to assume harmonic oscillator basis states, but this is mostly out of simplicity and convenience: there is a single parameter to choose.

To solve the many-body problem, we use a configuration-interaction code, BIGSTICK [67, 69], to generate from the single-particle energies and two-body matrix elements the many-body Hamiltonian and solve for extremal eigenvectors and eigenvalues using the Lanczos algorithm. BIGSTICK uses an M-scheme basis, which means all basis vectors (and thus, all solutions) share the same value for the z-component of total angular momentum, M or J_z. Because the matrix elements are read in as an external file, BIGSTICK does not depend upon a particular choice of single-particle basis. Any other nuclear configuration-interaction shell model code will yield the same results from the same input files.

Our interaction of interest is a widely-used, 'gold standard' empirical shell-model interaction, Brown and Richter's universal *sd*-shell interaction, version B, or USDB [18], a set of 66 parameters fit to energy data for a number of nuclei between ^{16}O and ^{40}Ca. In previous work [42] we performed a sensitivity analysis of USDB, which gives a probability distribution $P(\boldsymbol{\lambda})$ from which the interaction is sampled. An important nuance in using the USDB parameters is that while the single-particle energies are fixed, the two-body matrix elements are scaled by $(A_0/A)^{0.3}$, where A is the mass number of the nucleus, and A_0 is a reference value, here = $A_{\mathrm{core}} + 2 = 18$. We account for this by modifying the Hamiltonian expression $\hat{H} = \sum_i \lambda_i (A_0/A)^{0.3} \hat{\mathcal{O}}_i$ (but only for the two-body matrix elements), so that we implicitly varied the parameters fixed at $A = 18$.

Eigenstates of $\hat{H}$ give us many-body wavefunctions: $|\psi\rangle$; from these we measure transition matrix elements which depend on different initial and final states. The result is a number related to the probability of state $|\psi_i\rangle$ being transformed to state $|\psi_f\rangle$ by an operator $\hat{O}$ representing coupling to an external field:

$$M_{if}(\hat{O}) = \langle \psi_f | \hat{O} | \psi_i \rangle \tag{5.2}$$

The wavefunctions, being solutions to the eigenvalue problem, depend on $\boldsymbol{\lambda}$, and we assign the label θ to any parameters in the operator.

$$M_{if}(\hat{O}; \boldsymbol{\lambda}, \theta) = \langle \psi_f(\boldsymbol{\lambda}) | \hat{O}(\theta) | \psi_i(\boldsymbol{\lambda}) \rangle \tag{5.3}$$

The experimental transition rate is proportional to $|M_{if}|^2$. We leave out the technical elements of computing transition matrix elements, detailed in the literature [22, 21, 24, 123]. The most important point is that we consider standard one-body transition operators easily represented through second quantization:

$$\hat{O}(\theta) = \sum_{rs} \omega_{rs}(\theta) \hat{a}_r^\dagger \hat{a}_s, \tag{5.4}$$

where, again, the dependence of the coefficients ω_{rs} upon the effective parameters θ is well-known.

Hence, we have the computational pipeline of nuclear observables: $\boldsymbol{\lambda}$ determines the Hamiltonian → eigenstates of the Hamiltonian determine wavefunctions → matrix elements of transition operators between those eigenstates determine transition probabilities. Thus, we pursue two questions. First, given $P(\boldsymbol{\lambda})$, how does $M_{if}(\hat{O}; \boldsymbol{\lambda}, \theta)$ behave? And second, given $P(\boldsymbol{\lambda})$ and some experimental observations O, what is our posterior distribution $P(\theta | O)$? In many cases, it is useful to produce a covariance matrix for θ, which describes sensitivity of θ with respect to $O(\boldsymbol{\lambda}, \theta)$ about an optimal value.

5.3 Sensitivity analysis of the interaction

Our prior sensitivity analysis in [42] assumed that the Hamiltonian parameters (matrix elements) $\boldsymbol{\lambda}$ follow a normal distribution; thus their variability is fully described by a covariance matrix C_λ. The covariance is the inverse of the Hessian H_λ (not to be confused with the Hamiltonian $\hat{H}$), that is, the second derivatives of energy errors with respect to parameters $\boldsymbol{\lambda}$ evaluated with the parameters that minimize the error, $\boldsymbol{\lambda}^*$. (The notation $\boldsymbol{\lambda}^*$ for parameters at the minimum, while not common in nuclear physics, is used in uncertainty quantification [53].) We arrived at two important conclusions: (1) the Hessian matrix can be well approximated by a simpler calculation depending on first derivatives alone, and (2) we confirmed previous findings (e.g., [18]) that the parameter space is dominated by a small number of principal components, specifically that only 10 of the total 66 linear combinations of parameters account for 99.9% of the Hamiltonian. (The first most important linear combination carries 95%.)

The Hessian matrix is the second derivative of the χ^2 function about the optimal model parameters $\boldsymbol{\lambda}^*$:

$$[H_\lambda]_{ij} = \frac{1}{2}\frac{\partial^2}{\partial\lambda_i \partial\lambda_j}\chi^2(\boldsymbol{\lambda})\bigg|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}^*} \tag{5.5}$$

where

$$\chi^2(\boldsymbol{\lambda}) = \sum_{\alpha}^{N_d} \frac{(E_\alpha^{\mathrm{exp}} - E_\alpha^{\mathrm{th}}(\boldsymbol{\lambda}))^2}{\sigma(E_\alpha^{\mathrm{exp}})^2 + \sigma(E^{\mathrm{th}})^2} = \boldsymbol{e}_E^T(\boldsymbol{\lambda}) C_E^{-1} \boldsymbol{e}_E(\boldsymbol{\lambda}). \tag{5.6}$$

Here $\boldsymbol{e}_E(\boldsymbol{\lambda})$ is the energy error vector, C_E is a matrix with energy variances along the diagonal, and N_d is the number of data points. The two contributions to variances are the experimental variance $\sigma(E_\alpha^{\mathrm{exp}})^2$ and the *a priori* theoretical variance $\sigma(E^{\mathrm{th}})^2$ which is the same for all observations. The latter is tuned such that the χ^2 per degree of freedom,

$$\chi_\nu^2(\boldsymbol{\lambda}) = \frac{\chi^2(\boldsymbol{\lambda})}{\nu} = \frac{\chi^2(\boldsymbol{\lambda})}{N_d - N_p} \tag{5.7}$$

(a.k.a. the reduced χ^2) is 1. In doing so, we ensure the Hessian matrix be scaled appropriately, so the inverse may be interpreted as covariance. This procedure for determining the "static" theoretical uncertainty estimate is the same for the later analyses of transition strengths.

We found the Hessian for interaction parameters is well approximated by $H_\lambda \approx A_\lambda \equiv J^T_{E(\lambda)} C_E^{-1} J_{E(\lambda)}$ where $\left[J_{E(\lambda)}\right]_{i\alpha} = \partial E^{\mathrm{th}}_\alpha(\boldsymbol{\lambda})/\partial\lambda_i$ is the Jacobian matrix, which is computed easily using the Feynman-Hellman theorem [59, 38]: $\left[J_{E(\lambda)}\right]_{i\alpha} = \langle\alpha|\hat{\mathcal{O}}_i|\alpha\rangle$, an expectation value of eigenstates $\hat{H}(\boldsymbol{\lambda})|\alpha\rangle = E^{\mathrm{th}}_\alpha(\boldsymbol{\lambda})|\alpha\rangle$. This approximation replaces an $O(N_p^2)$ calculation with an $O(N_p)$ calculation while only introducing small errors.

We thus describe the distribution for interaction parameters $\boldsymbol{\lambda}$ as normal with mean at the fitted USDB values and with the covariance given by our approximation:

$$\boldsymbol{\lambda} \sim \mathcal{N}\left(\mathrm{mean} = \boldsymbol{\lambda}_{\mathrm{USDB}}, \mathrm{cov} = [J^T C_E^{-1} J]^{-1}\right) \equiv P(\boldsymbol{\lambda}) \tag{5.8}$$

We use this description to propagate uncertainty of $\boldsymbol{\lambda}$ to calculations of observables.

5.4 Bayesian parameter estimation

Bayesian data analysis is so named because it makes use of *Bayes' rule* to create statistical models:

$$P(\theta|y) = \frac{P(y|\theta)P(\theta)}{P(y)} \tag{5.9}$$

The *posterior* $P(\theta|y)$ is the ultimate goal of our analysis: a distribution describing the quantity of interest θ given observations y. The *likelihood* $P(y|\theta)$ describes the converse of the posterior: how observations y behave given θ. The *prior* $P(\theta)$ is the probability distribution of θ according to our prior knowledge. Finally, the *evidence* $P(y)$ describes any bias to observations y. Oftentimes, including in this work, $P(y)$ is constant.

Following typical Bayesian data analysis, θ stands for any parameter or vector of parameters we are modeling; here those parameters are the coupling constants (and, for electric quadrupole transitions, the basis harmonic oscillator length parameter) appearing in transition operators. This should not be confused with the vector of Hamiltonian parameters $\boldsymbol{\lambda}$, which is well described by a Gaussian approximation, while θ will be evaluated by Markov Chain Monte Carlo. To describe θ, we ultimately must compute the posterior distribution with respect to experimental observations O, $P(\theta|O)$. We take several mathematical steps to put the posterior $P(\theta|O)$ into a form we can compute. First, we introduce the interaction parameters as a marginal variable (that is, a new variable being integrated over) in $\mathbb{R}^{N_p}$, where N_p is the number of parameters (here, 66):

$$P(\theta|O) = \int_{\mathbb{R}^{N_p}} P(\theta, \boldsymbol{\lambda}|O)\, d\boldsymbol{\lambda} \tag{5.10}$$

By the chain rule of probabilities, we can reinterpret $\boldsymbol{\lambda}$ as a conditional variable if we also insert its prior in the integrand.

$$\int_{\mathbb{R}^{N_p}} P(\theta, \boldsymbol{\lambda}|O)\, d\boldsymbol{\lambda} = \int_{\mathbb{R}^{N_p}} P(\theta|O, \boldsymbol{\lambda}) P(\boldsymbol{\lambda})\, d\boldsymbol{\lambda} \tag{5.11}$$

Since we can easily sample the distribution $P(\boldsymbol{\lambda})$ via Eq. (5.8), we approximate the integral over $P(\theta|O, \boldsymbol{\lambda})P(\boldsymbol{\lambda})$ as an average of $P(\theta|O, \boldsymbol{\lambda}_k)$ for N_s interaction samples $\boldsymbol{\lambda}_k$.

$$\int_{\mathbb{R}^{N_p}} P(\theta|O, \boldsymbol{\lambda}) P(\boldsymbol{\lambda})\, d\boldsymbol{\lambda} \approx \frac{1}{N_s} \sum_{\boldsymbol{\lambda}_k \sim P(\boldsymbol{\lambda})} P(\theta|O, \boldsymbol{\lambda}_k) \tag{5.12}$$

This approximation works due to the idea of importance sampling in Monte Carlo integration, and the integral is better approximated as $N_s \to \infty$. By the central limit theorem, the sum converges to the integral with errors on the order of $1/\sqrt{N_s}$; to keep this error less than 1%, we use 10,000 samples for each calculation.

Next, we apply Bayes' rule to the summand. As mentioned above, the probability distribution of experimental observations, $P(O)$, often referred to as the "evidence", is the same for all possible O, and thus is constant. We ignore it and the factor of $1/N_s$ since the posterior need not be normalized in order to model θ.

$$\frac{1}{N_s} \sum_{\boldsymbol{\lambda}_k \sim P(\boldsymbol{\lambda})} P(\theta|O, \boldsymbol{\lambda}_k) = \frac{1}{N_s} \sum_{\boldsymbol{\lambda}_k \sim P(\boldsymbol{\lambda})} \frac{P(O|\theta, \boldsymbol{\lambda}_k) P(\theta)}{P(O)} \propto P(\theta) \sum_{\boldsymbol{\lambda}_k \sim P(\boldsymbol{\lambda})} P(O|\theta, \boldsymbol{\lambda}_k) \tag{5.13}$$

The right-hand side of this equation we can compute, and the likelihood is:

$$P(O|\theta, \boldsymbol{\lambda}_k) = \exp\left[-\frac{1}{2}\chi^2(\theta, \boldsymbol{\lambda}_k)\right] \tag{5.14}$$

The function $\chi^2(\theta, \boldsymbol{\lambda}_k)$ is similar to Eq. (5.6), but is defined for the observables O instead of energies.

$$\chi^2(\theta, \boldsymbol{\lambda}_k) = \boldsymbol{e}_O^T(\theta, \boldsymbol{\lambda}_k) C_O^{-1} \boldsymbol{e}_O(\theta, \boldsymbol{\lambda}_k), \tag{5.15}$$

where $[\boldsymbol{e}_O(\theta, \boldsymbol{\lambda}_k)]_\alpha = (O_\alpha^{\mathrm{exp}} - O_\alpha^{\mathrm{th}}(\theta, \boldsymbol{\lambda}_k))$ is the error vector and $[C_O]_{\alpha\alpha} = \sigma(O_\alpha^{\mathrm{exp}})^2 + \sigma(O^{\mathrm{th}})^2$ is the matrix with total variances along the diagonal. Here, the *a priori* theoretical uncertainty $\sigma(O^{\mathrm{th}})$ is determined in the same way as for Eq. (5.7), by setting the reduced χ^2 involving θ,

$$\chi_\nu^2(\theta, \boldsymbol{\lambda}_k) = \frac{\chi^2(\theta, \boldsymbol{\lambda}_k)}{\nu} = \frac{\chi^2(\theta, \boldsymbol{\lambda}_k)}{N_d - N_p}, \tag{5.16}$$

to unity and evaluating at $\boldsymbol{\lambda}_{\mathrm{USDB}}$, which also ensures the Hessian with respect to O has proper scaling, $H_O = C_O^{-1}$. Here, N_d is the number of observations O and N_p is the number of total parameters in θ and $\boldsymbol{\lambda}$.

For simplicity of the calculation, $\sigma(O^{\mathrm{th}})$ is computed once using a reasonable *a priori* choice of parameters, and does not change with samples of θ and $\boldsymbol{\lambda}$. Finally, we have an expression for the posterior distribution θ in terms of things we can compute:

$$P(\theta|O) \propto P(\theta) \sum_{\boldsymbol{\lambda}_k \sim P(\boldsymbol{\lambda})} \exp\left[-\frac{1}{2}\boldsymbol{e}_O^T(\theta, \boldsymbol{\lambda}_k) C_O^{-1} \boldsymbol{e}_O(\theta, \boldsymbol{\lambda}_k)\right] \tag{5.17}$$

Each transition operator has unique parameters θ and observations O, but the general process of describing $P(\theta|O)$ is the same for any observable. We decide on the prior $P(\theta)$, construct the expression in Eq. (5.17), and measure it. Since contributions to the likelihood in Eq. (5.14) are highly non-Gaussian, we evaluate using Markov Chain Monte Carlo, the affine-invariant ensemble sampler from `emcee` [40]. However, by the central limit theorem the sum over $P(O|\theta, \boldsymbol{\lambda}_k)$ approaches a Gaussian as the number of observations becomes large, meaning θ can ultimately be well described by mean and covariance alone.

Due to our frequentist approximation of the total likelihood function, the above procedure might be called "pseudo-Bayesian". A fully Bayesian analysis, rather than summing over many likelihoods, would probably define likelihood as a function of $(\theta, \boldsymbol{\lambda})$ together and construct the posterior accordingly: $P(\theta)P(\boldsymbol{\lambda})P(O|\theta, \boldsymbol{\lambda})$. Indeed, doing so would elucidate correlations between the Hamiltonian matrix elements and operator parameters. We choose to construct the approximate likelihood from many samples of $\boldsymbol{\lambda}$ mainly due to practical challenges. In order to evaluate the likelihood function, which contains a sum over observables in many nuclides, perturbations to the Hamiltonian must not result in the relevant eigenstates vanishing or otherwise being too difficult to track. A single evaluation of the likelihood requires at least dozens of individual shell model calculations, perhaps even hundreds in extreme cases. Sampling according to our prior PCA makes it more likely that these calculations produce a sensible answer for every observable. Our codes compute wavefunction overlaps upon each iteration to help track eigenstates; this grants some robustness but it is not perfect. A potential solution to this problem is using eigenvector continuation, which recently has been applied to shell model calculations [138], but that is beyond the scope of this paper.

5.5 Results

Here we present the results of uncertainty quantification for Gamow-Teller (GT), electric quadrupole (E2), and magnetic dipole (M1) transitions. The analyses are presented in order of increasing difficulty: GT has only one parameter and thus is the simplest; E2 has three but only two are independent; while M1 has four parameters, we must couple two of them in order to get a useful result. All our experimental observations O are reduced transition strengths, $B(\hat{O}) = |M(\hat{O})|^2/(2J_i + 1)$, where $M(O)$ is the reduced [33] matrix element for transition operator $\hat{O}$, and J_i is the initial angular momentum. Transitions outside the range of model validity were excluded, including isotopes which have no protons or neutrons (or proton/neutron holes) in the valence space. For electric quadrupole (E2) and magnetic dipole (M1) transitions, we converted to Weisskopf units to get a sense of their strength with respect to single particle estimates, then truncated datasets to exclude extreme cases. E2 transitions with strength $< 0.1\times$ and $> 150\times$ the Weisskopf single particle estimate were excluded. This shrunk the total E2 data set from 236 to 153 transitions. For M1, which tend to be smaller on average, we only left out those with $< 0.01\times$ the Weisskopf estimate. We also dropped some M1 transitions involving very excited states, in particular if either state is above 6 excitations for a particular total angular momentum J. This altogether shrunk our M1 data set from 167 to 143 transitions.

For each transition operator, we present an *a priori* theoretical uncertainty: it is a fixed estimate of theoretical error based on setting the reduced χ^2 to one using a *prior* parameter estimate. These provide a useful prediction of theory error, but they can be sensitive to choices of the UQ analysis, prior parameterizations, data sets, etc. The study of observables using USD Hamiltonians by Richter, Mkhize, and Brown (2008) [109] serves as our primary source for prior information on parameter uncertainties.

Exact probability distributions mentioned are either uniform on a closed interval $[a, b]$, written $U_{[a,b]}$, or normal with mean μ and standard deviation σ, written $\mathcal{N}(\mu, \sigma)$.

5.5.1 Gamow-Teller transitions

Both the vector and axial-vector weak couplings, g_V, g_A respectively, have been measured from the β-decay of a free neutron, with $|g_A/g_V| \approx 1.28$ [109]. Empirical shell-model calculations of allowed transitions, specifically Gamow-Teller reduced transition matrix elements

$$M_{if}(GT) = \left\langle J_f \left\| g_A^{\text{eff}} \vec{\sigma} \tau_{\pm} \right\| J_i \right\rangle, \tag{5.18}$$

when compared to experiment, consistently lead to a quenched coupling, $g_A^{\text{eff}} = Q g_A$ [21, 24]; Q is called the *quenching factor*. Recent work in *ab initio* calculations have shown quenching can largely be accounted for by including physics beyond capabilities of the effective shell model, including two-body currents, sometimes interpreted as meson exchange, and long-range energy correlations [57].

Using 185 low-lying β^+, β^-, and electron-capture transitions, we assign an *a priori* theoretical uncertainty in the (dimensionless) $B(GT)$ to be 0.30, based on setting the reduced χ^2 to unity. Examples of the resulting $B(GT)$ distributions are shown in Fig. 5.1. The only *a priori* assumption in our UQ is that $0.5 < g_A^{\text{eff}}/g_A < 1.0$ so our prior is set to a uniform distribution within those bounds: $P(Q) = U_{[0.5,1.0]}$. Fig. 5.2 show our derived posterior, which is Gaussian with $Q = 0.762 \pm 0.025$. This is more tightly constrained than the estimate from Richter et al. 2008, $Q = 0.764 \pm 0.114$ [109], but our result is consistent with their conclusion that quenching of g_A in empirical calculations is robust and independent of the Hamiltonian.

5.5.2 Electric quadrupole transitions

The electric quadrupole (E2) reduced transition matrix element is

$$M_{if}(E2) = \langle J_f || \left[e_p E2_p + e_n E2_n \right] || J_i \rangle \tag{5.19}$$

where $E2 := r^2 Y_2^m$ and e_p, e_n are the effective charges for protons and neutrons, respectively. Because we assume harmonic oscillator single-particle wave functions, the matrix elements of r^2 are proportional to b^2, where b is the oscillator length parameter. Thus, the parameters of interest are $\theta = (b^2, e_p, e_n)$. However, b^2 and effective charges both contribute to the overall scaling of the matrix element, so we chose to separate this UQ into two steps: first we fix b^2 and model (e_p, e_n) , then fix (e_p, e_n) and model b^2.

First, we assign prior distributions for the parameters. Effective charges for E2 calculations take values of $e_p \in (1, 2)$ and $e_n \in (0, 1)$. Previous work by Richter *et al.* using the USDB interaction found optimal values $e_p = 1.36(5)$ and $e_n = 0.45(5)$ [109]. The oscillator length is parameterized by an existing global fit $b^2 = 0.9 \times A^{1/3} + 0.7$, called the Blomqvist-Molinari formula [13]. In total, we first model $\theta_1 = (e_p, e_n)$ with b^2 given by the formula, then separately reparameterize the oscillator length by

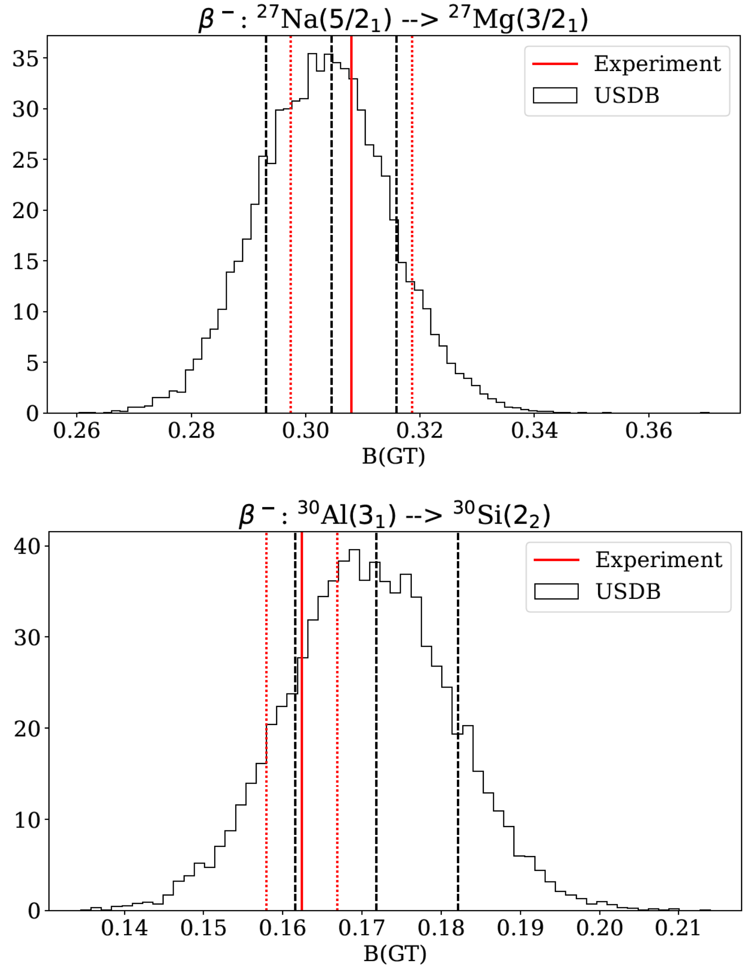


Figure 5.1. Two examples of B(GT) histograms computed using Monte Carlo. Due to the near-linearity of the model, B(GT) is almost always very nearly Gaussian.

$b^2 = \mu A^{1/3} + \beta$ and model $\theta_2 = (\mu, \beta)$ with effective charges fixed to prior optima (from [109]). These estimates are used to compute our *a priori* theoretical uncertainty estimate of 5.4 Weisskopf units, based on setting the reduced χ^2 to unity.

5.5.2.1 Effective charges: e_p, e_n

Whether we use a uniform prior, $P(e_p) = U_{[1,2]}$ and $P(e_n) = U_{[0,1]}$, or a Gaussian prior centered on previous optimal results, $P(e_p) = \mathcal{N}(\mu = 1.36, \sigma = 1)$ and $P(e_p) = \mathcal{N}(\mu = 0.45, \sigma = 1)$, the resulting posteriors for effective charges are almost identical (including reducing the prior standard deviations down to 0.5). The assigned

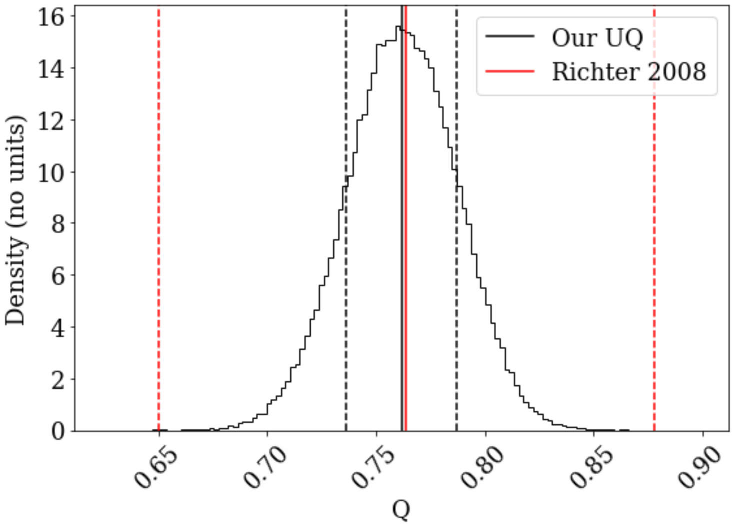
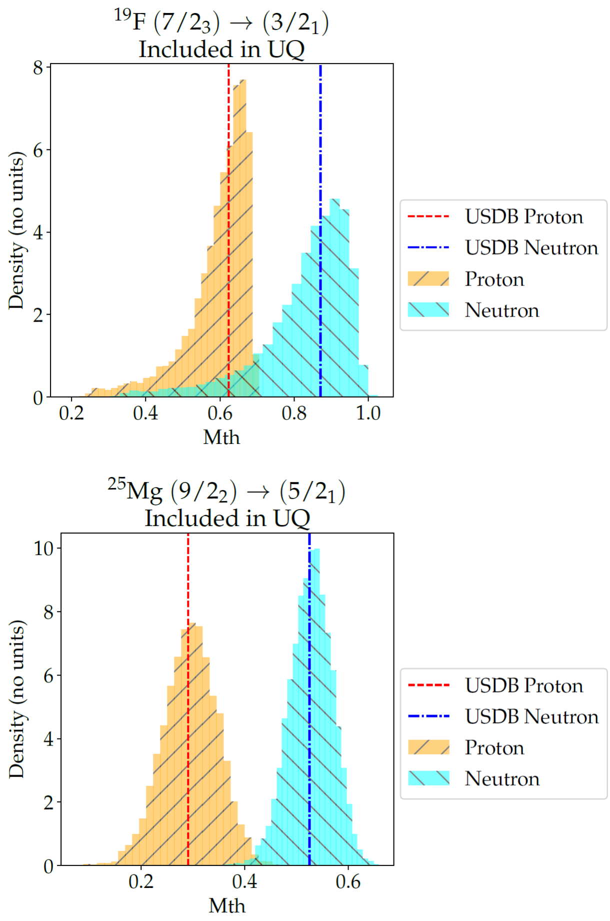


Figure 5.2. Histogram of the Gamow-Teller quenching factor Q via Monte Carlo; the posterior is Gaussian, $Q = 0.77 \pm 0.013$. This is more tightly constrained than the estimate from [109], $Q = 0.764 \pm 0.114$.

uncertainties in [109] are $\sigma = 0.05$ for both proton and neutron, too small to construct a sensible prior with, and thus the resulting posterior is constrained mainly by the likelihood.

Our results for e_p, e_n, 1.44 ± 0.17 and 0.40 ± 0.16, respectively, have central values within uncertainty of the estimates of [109], although we arrive at nearly three times the standard deviation. Fig. 5.4 shows the effective charges are strongly correlated, as expected. Fig. 5.5 shows the same data but in the isospin basis: the isoscalar component, $e_p + e_n = 1.84 \pm 0.05$ is tightly constrained, and we see much larger uncertainty on the isovector component $e_p - e_n = 1.04 \pm 0.32$. This agrees with a recent *ab initio* study [120].

Using our results for optimal effective charges, we compute again the static theoretical uncertainty in $B(E2)$ to be 4.1 in Weisskopf units. We suggest that static theoretical uncertainties be taken with a somewhat wide confidence interval, since this value is dependent on the experimental data and the parameters θ. (Additionally, since we have shown that the interaction is dominated by only 10 parameters, one could argue this uncertainty is actually smaller. If we compute the theoretical uncertainty



^{19}F $(7/2_3) \rightarrow (3/2_1)$
Included in UQ

^{25}Mg $(9/2_2) \rightarrow (5/2_1)$
Included in UQ

Figure 5.3. Two examples of M(GT) histograms computed using Monte Carlo.

again, setting the reduced chi-squared in Eq. 5.16 for $B(E2)$ to unity, and using $N_p = 10$ rather than 66, we get a 1σ value of 3.0 Weisskopf units.)

5.5.2.2 Oscillator length: b

Following the Blomqvist-Molinari formula, $b^2 = 0.9A^{1/3} + 0.7$, we parameterize by $b^2 = \mu A^{1/3} + \beta$, and assign normal priors centered at $\mu = 0.9$, $\beta = 0.7$. In an effort to use an informative prior that will not bias our result too strongly toward the Blomqvist-Molinari result (which was fit to nuclei beyond the *sd*-shell), we leave the prior standard deviation relatively large, $\sigma = 1$. Furthermore, we expect both μ and β to be positive, so the priors are only nonzero for positive values. Since any

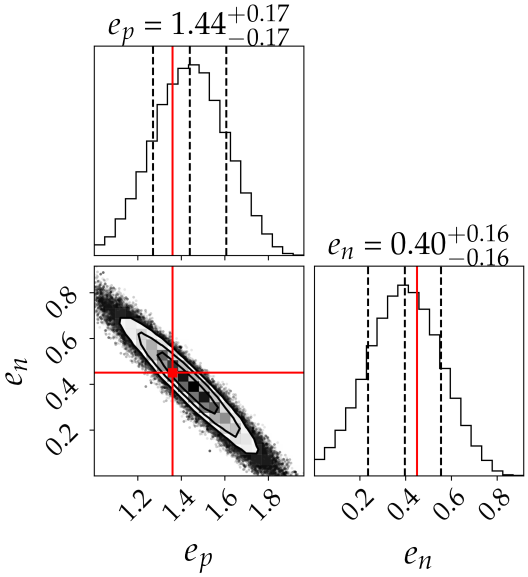


Figure 5.4. Joint histogram of the effective charges for proton and neutron in E2 transitions, with no data truncation based on relative errors. Red lines show values from [109], which were used to determine Gaussian priors. Flat priors give almost identical results.

normalization constant will not have an effect on the result of this calculation, the priors in Eq. 5.20 are defined as proportionalities. (We are not computing any *probabilities* in this work, thus the posterior need not be normalized. Regardless, the posterior could in principle be normalized without keeping track of individual normalizations of likelihood and prior distributions.)

$$P(\mu) \propto \begin{cases} \mathcal{N}(0.9, 1) & \text{if } \mu > 0 \\ 0 & \text{if } \mu < 0 \end{cases} \tag{5.20a}$$

$$P(\beta) \propto \begin{cases} \mathcal{N}(0.7, 1) & \text{if } \beta > 0 \\ 0 & \text{if } \beta < 0 \end{cases} \tag{5.20b}$$

For this calculation, the effective charges were fixed at the optimal values presented in [109]. The MCMC evaluation of μ and β are shown in Fig. 5.6. The two are strongly correlated and reveal a tendency of μ to decrease and β to increase,

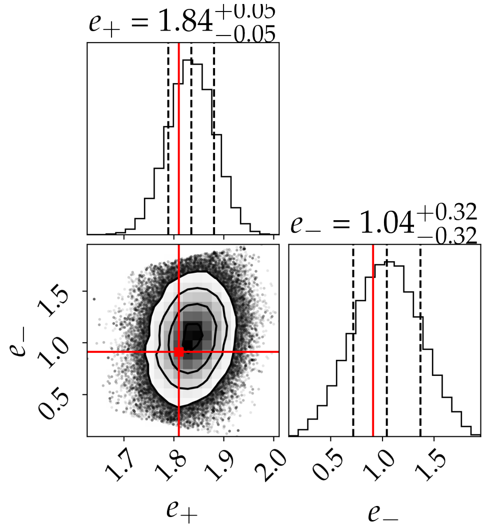


Figure 5.5. Effective charges determined by E2 transitions, in isoscalar/isovector terms: $e_\pm = e_p \pm e_n$. We see by comparing the standard deviations that the isoscalar component e_+ is much more tightly constrained than the isovector component e_-. Note that correlation between isospin components is far less than between proton/neutron components.

corresponding to a flattening out of b^2 as a function of $A^{1/3}$. This indicates that b^2 is less dependent on A in the sd-shell than the global fit indicates. Finally, we can use the posterior distributions of μ, β to compute b^2 as a function of mass number A with an uncertainty band, as shown in Fig. 5.7.

5.5.3 Magnetic dipole transitions

The magnetic dipole reduced transition matrix element is

$$M_{if}(M1) = \langle J_f || \left[g_{sp}\vec{s}_p + g_{sn}\vec{s}_n + g_{\ell p}\vec{\ell}_p + g_{\ell n}\vec{\ell}_n \right] || J_i \rangle \tag{5.21}$$

where $\vec{s}_t$ is the spin operator and $\vec{\ell}_t$ is the orbital angular momentum operator, and $t = p, n$ denotes action only on the proton/neutron part of the wavefunction. The parameters of interest are the coupling constants: $\theta = (g_{sp}, g_{sn}, g_{\ell p}, g_{\ell n})$. For the free nucleon, these coupling constants are $(5.5857, -3.8263, 1, 0)$, but previous work on M1 transitions with USDB [109] found optimal values of $(5.15, -3.55, 1.159, 0.09)$. The

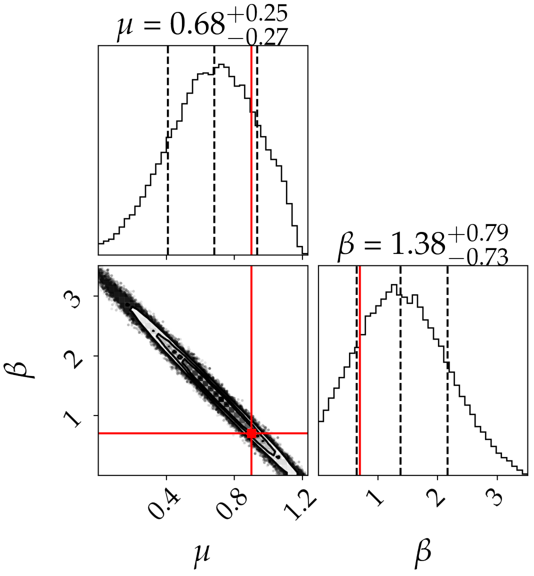


Figure 5.6. Joint histogram of parameters μ and β for oscillator length parameter b, with $b^2 = \mu A^{1/3} + \beta$. The red point indicates the global fit: $\mu = 0.9$ and $\beta = 0.7$. The tendency of μ (slope) to decrease and β (intercept) to increase, which we see in the Monte Carlo results, corresponds to b^2 flattening out as a function of $A^{1/3}$.

slight difference in values using USDB is due to physics left out of the empirical Hamiltonian , as well as optimizing for a finite number of transitions in the *sd*-shell.

The M1 transition is more challenging than the GT and E2 for a few reasons. Transition strengths tend to be small compared to the Weisskopf single-particle estimate, which can be due to cancellation between the four components, and/or quenching of the coupling constants similar to the quenching of g_A in the Gamow-Teller. Individual matrix elements can also be highly non-Gaussian with respect to variation in the Hamiltonian. Due to the central limit theorem however, distributions of the coupling constants approach normality when summing over many experimental data points.

Furthermore, the parameters are not constrained in a naive construction of the posterior distribution. When sampling all four coupling constants and using a flat prior, the parameters are allowed to move within a large region wherein the posterior (and, thus, the χ^2 function) is constant. This result is not useful for assigning uncertainty to

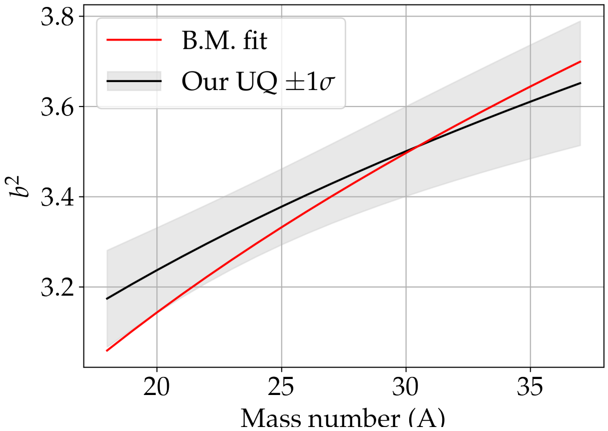


Figure 5.7. Square of the harmonic oscillator length parameter as a function of mass number A, comparing the Blomqvist-Molinari (B.M.) fit against ours. The band indicates 1σ uncertainty.

the coupling constants, so we explored several different paths for how to constrain the posterior in a statistic way. The flat shape of the likelihood means that imposing a restrictive prior alone is insufficient, since the resulting posterior would be fully determined by the prior in that region. To solve this, we transform our parameters to an isospin basis and fix the most tightly constrained dimension as noted by previous work [65]. Since we expect 1) the isovector part to fluctuate more than the isoscalar part, and 2) the spin part to fluctuate more than the orbital part, we choose to fix the orbital isoscalar coupling $g_{\ell p} + g_{\ell n}$ to be fixed to its optimal value found by Richter et al. 2008 [109]. The results for the $B(M1)$ couplings using a flat prior are shown in Fig. 5.8; the fixed orbital isoscalar component appears as a straight line on one correlation plot. Compared to the results in [109] our spin components are smaller in magnitude and orbital components are larger. We also construct informative priors from the results of [109], Gaussian distributions centered at optimal values for USDB and with standard deviations at 3× that of the tabulated results (The factor of 3 is chosen simply to inflate

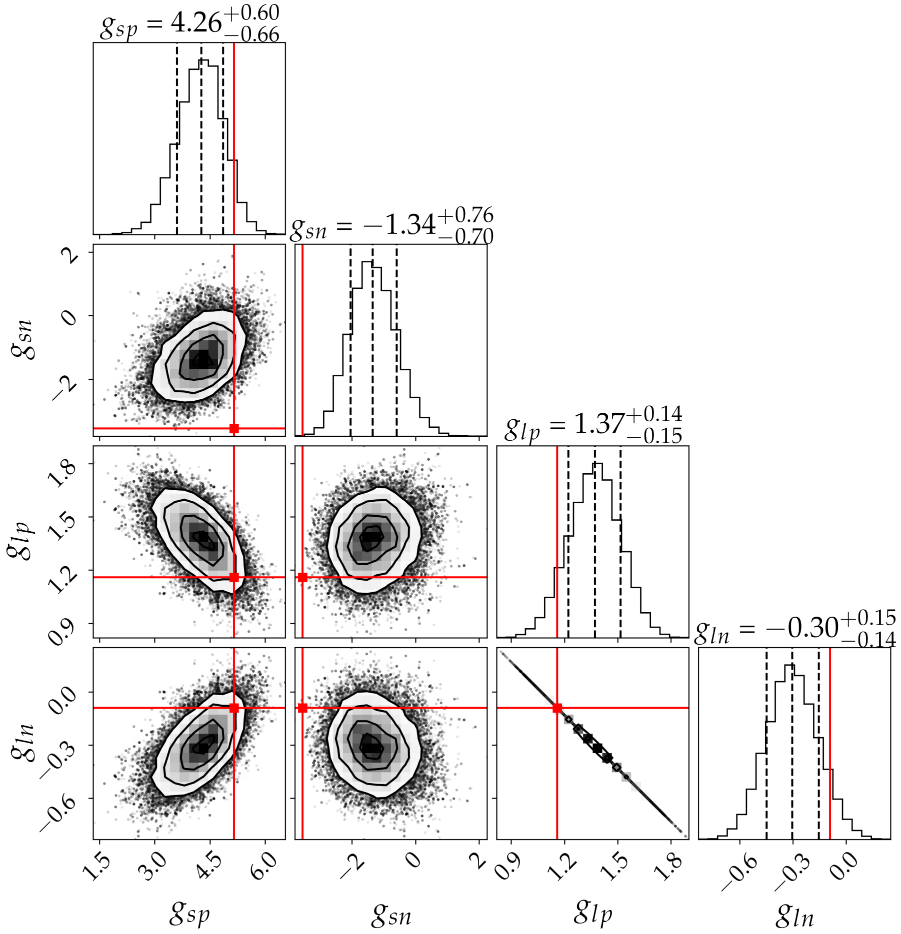


Figure 5.8. Posterior distribution of the M1 coupling constants using flat priors with the orbital isoscalar component fixed to a constant value (determined by optimal values listed in [109]). Hence the joint histogram for g_{ln} versus g_{lp} indicates a perfect anti-correlation. The red points show optimal values listed in [109]. Absolute magnitude of spin couplings is smaller than prior estimates, while absolute magnitude of orbital couplings is larger.

the Gaussian confidence interval from 68% to 99.7%) as shown here:

$$\begin{aligned}
P(g_{sp}) &= \mathcal{N}(5.15, \quad 0.27) \\
P(g_{sn}) &= \mathcal{N}(-3.55, \quad 0.3) \\
P(g_{\ell p}) &= \mathcal{N}(1.16, \quad 0.069) \\
P(g_{\ell n}) &= \mathcal{N}(-0.09, \quad 0.78)
\end{aligned} \tag{5.22}$$

As expected, using informative priors gives results nearer to those previously cited, but the spin components are still notably smaller in absolute magnitude. A number of M1 transitions in fluorine isotopes have large errors, so we also present results for the M1 couplings having dropped 18 particularly troublesome transitions which exhibit large relative errors. A likely explanation for difficulties in these matrix elements is that the structure of fluorine isotopes have significant *p*-*sd* particle-hole intruder configurations not included in our model. This combined with the fact that the M1 matrix element has four components and often involves some cancellations means we can easily over- or under-estimate the total transition strength. The UQ results without these 18 troublesome fluorine transitions is shown in Fig. 5.10.

5.5.4 Parameter sensitivity from energies, transitions, and sum rule operators

In our previous study [42] we evaluated the covariance of Hamiltonian parameters $\boldsymbol{\lambda}$ with respect to energies, i.e., Eq. (5.5), (5.6). If we evaluate with respect to other observables, such as transition strengths or transition sum rules, we get very different estimates. Here we show some results of computing approximate Hessian matrices of $\boldsymbol{\lambda}$ with respect to energy, which was used for our sensitivity analysis in this paper, and again using B(GT) values instead of energies. The (non-energy-weighted) sum-rule operator for transition $\hat{O}$ is $\hat{O}^\dagger \hat{O}$, and is so called because the expectation value implicitly counts up transition contributions over final states: $\langle \hat{O}^\dagger \hat{O} \rangle_i \propto \sum_f \langle i | \hat{O}^\dagger | f \rangle \langle f | \hat{O} | i \rangle$.

We compute all Hessians by the approximation $H_\lambda \approx J^T_{\Omega(\lambda)} C^{-1}_\Omega J_{\Omega(\lambda)}$ where Ω is the observable (energy, Gamow-Teller strengths, sum rules) and $\left[J_{\Omega(\lambda)} \right]_{i\alpha} = \partial \Omega^{\text{th}}_\alpha(\boldsymbol{\lambda}) / \partial \lambda_i$, and C_Ω is the diagonal matrix of errors.

The Hessian for energies, Eq. (5.5), (5.6), is plotted in Figure 5.12. Of all Hessian matrices, this appears to have the most structure, showing some complex correlations. The most sensitive parameters have 1σ uncertainties around 800-900 eV, and correspond to the Hamiltonian matrix elements:
$\langle 0d_{5/2} 0d_{5/2}; J, T = 4, 1 | \hat{H} | 0d_{5/2} 0d_{5/2}; J, T = 4, 1 \rangle$,

$\langle 0d_{5/2}1s_{1/2}; J, T = 3, 1|\hat{H}|0d_{5/2}1s_{1/2}; J, T = 3, 1\rangle$, and $\langle 0d_{3/2}0d_{5/2}; J, T = 4, 1|\hat{H}|0d_{3/2}0d_{5/2}; J, T = 4, 1\rangle$ which are the 51st, 60th, and 24th parameter in our arbitrary ordering, respectively. These are matrix elements between normalized two-body states $|ab; JT\rangle$ with nucleons in orbitals a and b coupled up to total angular momentum J and total isospin T; the commonplace notation for the shell model is $V_{JT}(ab, cd)$.

The Hessian for Gamow-Teller transitions, B(GT), is in Figure 5.13. We can see two Hamiltonian parameters which are especially important for B(GT), are $V_{41}((0d_{5/2})^2, (0d_{5/2})^2)$ and $V_{31}(0d_{5/2}1s_{1/2}, 0d_{5/2}1s_{1/2})$, which were also important for energies, which have 1σ uncertainties of 32 and 43 keV respectively. That is, GT transition strengths are particularly sensitive to the two-body matrix elements (TBMEs) of these two-body operators. These also reveal strong negative correlations between their TBMEs. We find that the Hamiltonian matrix element $V_{41}((0d_{5/2})^2, (0d_{5/2})^2)$ is very important in all transition types, not only the Gamow-Teller. The Hamiltonian matrix elements $V_{31}(0d_{5/2}1s_{1/2}, 0d_{5/2}1s_{1/2})$ and $V_{41}(0d_{5/2}0d_{3/2}, 0d_{5/2}0d_{3/2})$ are also especially sensitive for other transition strengths. On average, the three operators listed above are the most important for the transition strengths we computed.

5.6 Conclusions

We have presented a method of uncertainty quantification for parameters of transition matrix elements resulting from empirical shell model calculations. Our parameters fall into two categories: those in the nuclear Hamiltonian (λ), and those in the transition operator (θ). The present analysis is primarily concerned with the latter. While a fully Bayesian UQ analysis would ideally model λ and θ together, we assign a distribution to λ and construct a likelihood for θ based on a fixed sampling of λ. The result trades a high-resolution picture of λ in exchange for relatively quick calculation of the θ posterior.

The work presented here fits into the larger picture of theoretical UQ in nuclear theory. While significant efforts have been underway for many years to quantify uncertainties in nuclear theory, from uncertainties in nuclear interactions [92, 49, 23, 133, 87, 100] and *ab initio* calculations of light an medium nuclei [12, 50, 132, 34] to mean field calculations of heavy nuclei [111, 117, 37, 98] and simulations of astronomical nucleosynthesis processes [124, 89, 85, 91, 90, 126], UQ in empirical shell model has been less abundant [139, 64]. In particular this work is, to our knowledge, the first approach to quantifying uncertainties of transition operators.

Further research should use our results to inform more accurate UQ analyses. For instance, recent work has shown eigenvector continuation (EC) to be an extremely powerful approach to emulating eigenvalue problems. One could do away with the simplistic $P(\lambda)$ and instead define the likelihood with a EC model. That way, one may evaluate a joint posterior $P(\lambda, \theta|D)$ by MCMC and potentially get a more accurate correlation analysis.

5.7 Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under Award Number DE-FG02-03ER41272.

Computing support for this work came in part from the Lawrence Livermore National Laboratory institutional Computing Grand Challenge program. Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344.

Corner plots made using the `corner` library [39].

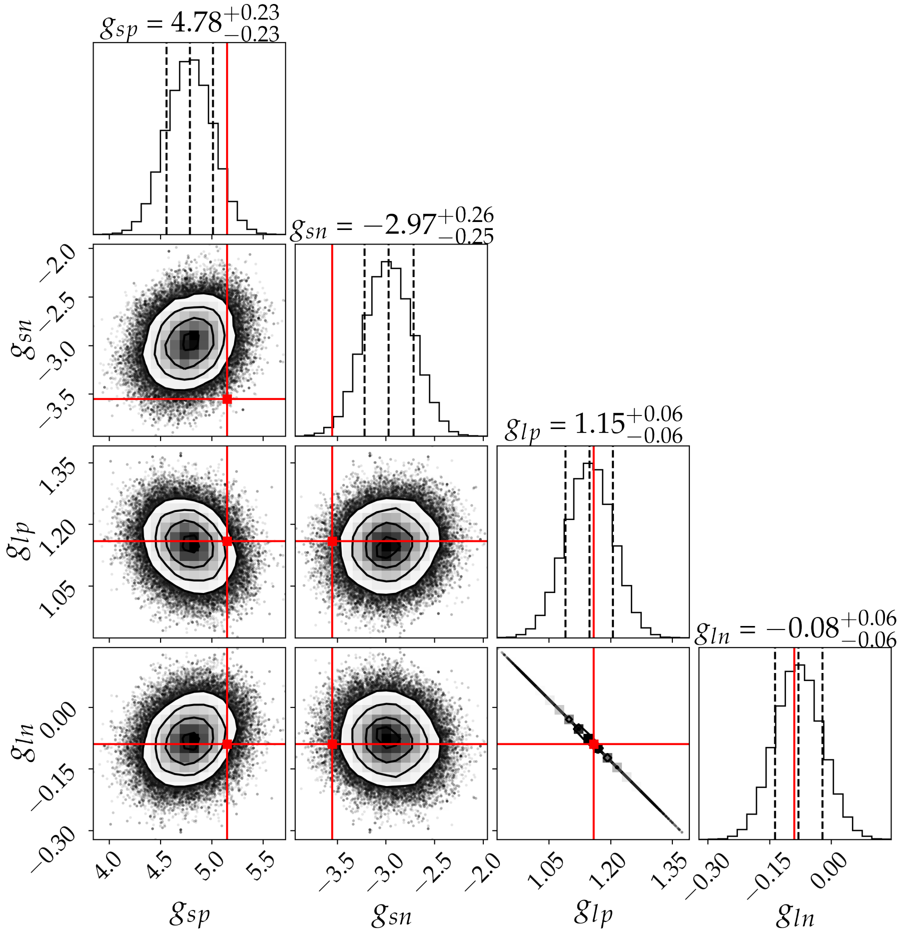


Figure 5.9. Similar to Fig. 5.8: posterior distribution of the M1 coupling constants, but using Gaussian rather than flat priors. The orbital couplings are perfectly anti-correlated because their sum is fixed to a constant. We apply informative priors shown in Eq. (5.22), which are based on previous measurements [109], which are shown here in red. The result confirms that the orbital components are more constrained than the spin components. Again, we find that absolute magnitude of spin couplings is smaller than prior estimates, while absolute magnitude of orbital couplings is larger.

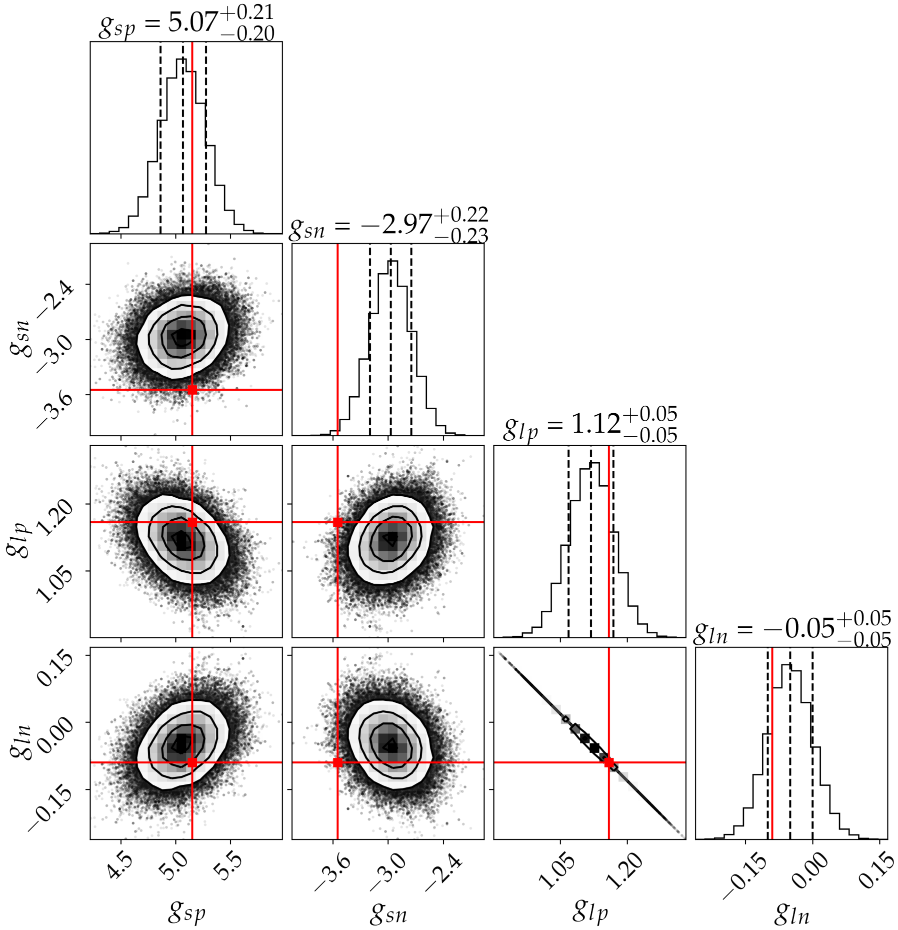


Figure 5.10. Posterior distribution of the M1 coupling constants excluding some transitions in fluorine with large errors. Removing the troublesome data points has a substantial change only on g_{sp}, driving it slightly closer to the previously cited value in [109].

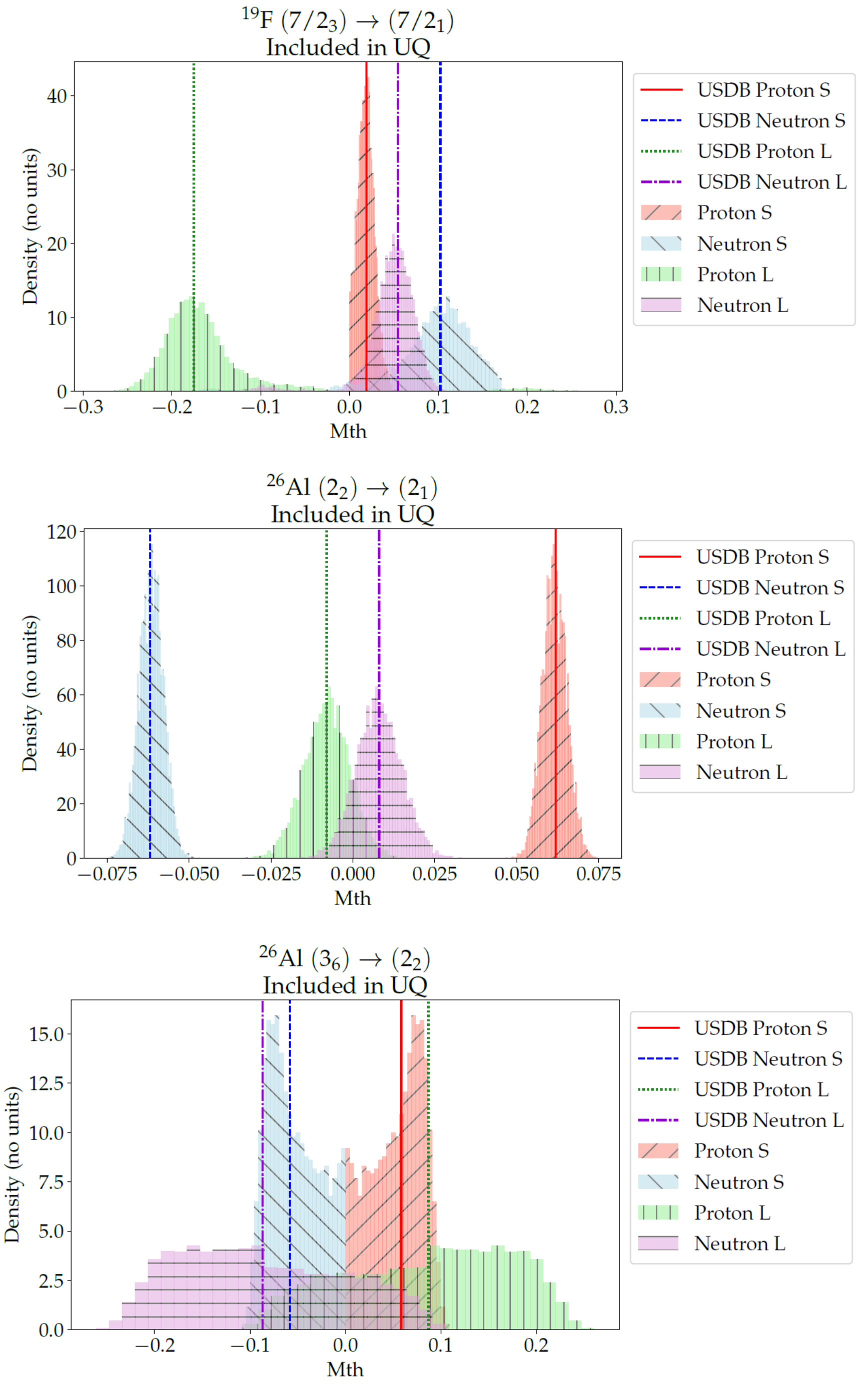


Figure 5.11. Three examples of M(M1) histograms computed using Monte Carlo. Most matrix elements are Gaussian but nonlinear dependence on the wavefunction can lead to asymmetric distributions.

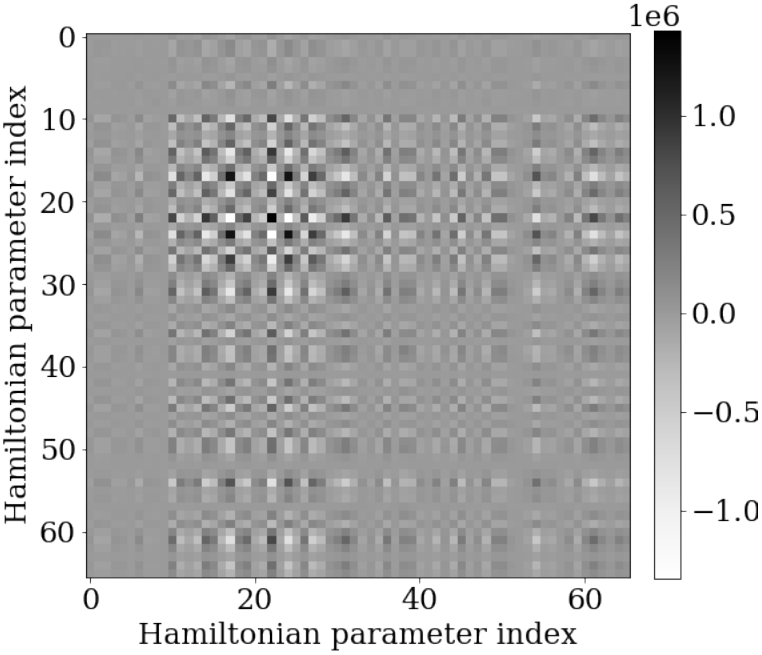


Figure 5.12. $A_{\lambda(E)}$**, approximate Hessian matrix for Hamiltonian parameters** $\boldsymbol{\lambda}$**, computed from energies. Note how the structure of the matrix is very different when using** $B(GT)$ **transition strengths in place of energies, as shown in Fig. 5.13.**

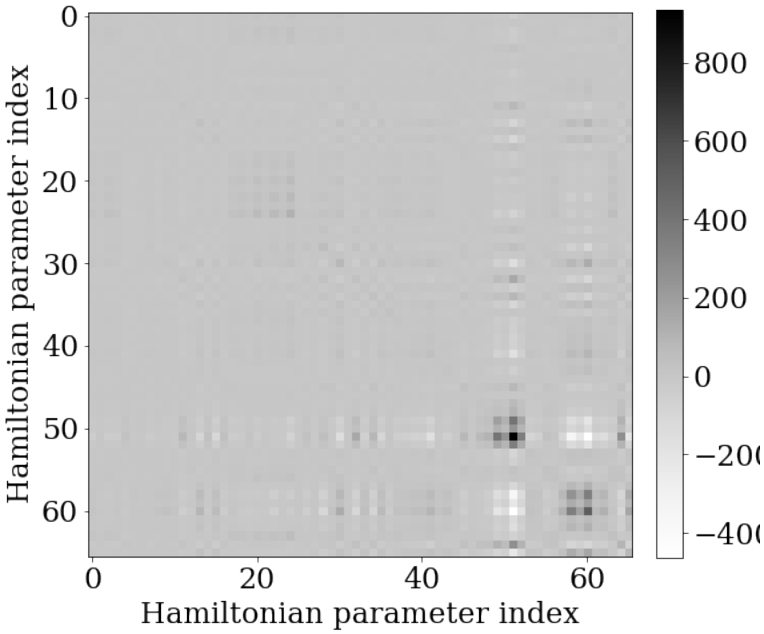


Figure 5.13. $A_{\lambda(B)}$**, approximate Hessian matrix for Hamiltonian parameters** $\boldsymbol{\lambda}$**, computed from Gamow-Teller transition strengths. The B-value is dominated by two TBMEs (#51, and 60 here): isovector pairs** $(0d_{5/2})(0d_{5/2})$ **with** $J = 4$**, and isovector pairs** $(0d_{5/2})(1s_{1/2})$ **with** $J = 3$**.**

CHAPTER 6

PROJECT 2: ILLUMINATING TRENDS IN CROSS SECTION EVALUATIONS USING GENERATIVE MACHINE LEARNING

To be submitted to Physical Review.

6.1 Introduction

6.1.1 Nuclear cross section evaluations

Since many applications of nuclear theory require continuous descriptions of nuclear reaction cross sections over scattering energy, experimental data are combined with theoretical models to produce so-called cross section *evaluations*. These evaluations are intended to give a comprehensive picture of scattering processes and are organized into large libraries; these are an essential resource for astrophysical simulation, radioisotope studies for medicine, reactor engineering, and national security applications. Evaluated libraries include complicated systematic trends due to the fundamental physics, but these can be difficult to study due the tremendous size of these libraries and their density of information. The motivation for this work is to develop a machine learning system to facilitate illuminating and analyzing trends in cross sections and using existing information to predict cross sections beyond experimental barriers.

As a first step, we focus on the TENDL library [77] and the inelastic neutron scattering channel (n, n') in particular, with resonances excluded; this choice was made because of relative simplicity of correlations in this channel. Because of odd-even staggering, the well-known behavior of systematic trends in nuclei to oscillate with parity of proton/neutron numbers Z, N, we only work with even-even nuclei in the present study. While this significantly reduces the complexity of trends the model must learn, it nonetheless requires a model with sufficiently large complexity to make reasonable predictions. This simplification makes it much easier to interpret results and better develop the model; this work acts as a foundation for future development of more sophisticated models.

Many different types of correlations exist in this data set (i.e. at different scales over the chart of nuclides), making it a ripe target for data science and machine learning. For instance, a predictive model must learn correlations in the cross sections between values of scattering energy, correlations between nuclei with similar numbers of constituent particles, and (in future development) correlations between cross sections in different reaction channels.

6.1.2 Model overview

Our deep learning model has two major components working in tandem: a variational auto-encoder (VAE) responsible for encoding the cross section evaluation in a dense representation, and the generative adversarial network (GAN) responsible for learning to transform a cross section at one nuclide to that of a nearby nuclide. The VAE learns to encode the cross section data in a smaller dimensional space, called the *latent space*. Cross sections are continuous functions of energy, and they have strong short-range correlations and weak long-range correlations; in other words, the covariance matrix is diagonally dominant. We leverage this property by using convolutional layers in the VAE neural networks which are formulated to learn multi-scale correlations; each consecutive hidden layer is responsible for a slightly larger or smaller correlation length. This is similar to models used for image recognition: long-range correlations are not ignored, but the relative importance of short-range correlations is built directly into the network architecture. The VAE has an additional benefit of mapping inputs to normal distributions in the latent space, rather than points like a simple autoencoder would. This results in smooth and very efficient encoding of data, and thus we think is more likely to provide better interpolation properties. The generative adversarial network is designed to work on the encoded cross section and map between nuclides. The GAN consists of *generator* and *discriminator* networks: the former transforms the encoded cross section and the latter classifies the outputs as realistic/unrealistic according to what features are present in the training data. These two networks are adversaries; while training they compete with each other until an equilibrium is reached. At that point the transformed data *appears*, according to the discriminator, to have all the necessary features of the real data. Lastly, each linear axis across the chart of nuclides ($\updownarrow$, $\swarrow\!\!\!\nearrow$, $\leftrightarrow$, $\nwarrow\!\!\!\searrow$) gets its own model, and as such the full predictive system is an *ensemble* (which has advantages and disadvantages). With this construction we can make predictions for cross sections of nuclides outside the training set, chaining together predictions using many paths across the chart. Our hypothesis is that this model, when trained correctly, can help illuminate systematic trends in nuclear data libraries.

6.1.3 A convolutional variational autoencoder (VAE) for nuclear data

The first step in our procedure, as is common for machine learning tasks in general, is to find an ideal representation of the data. This almost always includes some kind of normalization, and also may include an *encoding* to simplify correlations in the data and reduce dimension. One powerful method of data encoding uses a neural network model called an *auto-encoder* (AE). In practice, the AE can be thought of as a nonlinear principal component analysis (PCA). When doing linear PCA via singular value decomposition (SVD), one finds orthogonal principal components of data and the whole data set can be represented using those components as a basis. A common procedure for data encoding with SVD is to drop the basis components with small singular values, and represent all data with the remaining components, thus one achieves an efficient representation of the data in a smaller dimension. Inevitable errors are introduced by reducing the number of variables, but PCA ensures components with small singular values are the least important and so ignoring them introduces the least error possible. However, linear PCA includes some crucial assumptions about the data: the data is a linear combination of explanatory variables, all explanatory variables have been observed roughly the same amount, correlations follow a Gaussian distribution, the data is primarily unimodal, etc. Nuclear cross section evaluations break all of these rules, and so we must turn to a more robust encoding method. An example of linear PCA encoding and reconstruction on our cross section data is shown in Fig. 6.1; reproducing the small-scale features would require keeping many principal components, thus defeating the purpose of using this approach. (Furthermore, cross sections are strictly positive, and using linear PCA to reduce dimension inevitably produces negative cross sections.)

A simple AE model is a neural network mapping from data X to itself, and some layer inside the network is the latent space representation of the input. The model is split into an *encoder* part E before the latent layer and a *decoder* part D after the latent layer, and the full model is a function convolution $D(E(x)) = x'$. With weights θ and ϕ for D and E respectively, the simplest loss function penalizes errors over a batch of N_b data:

$$\mathcal{L}_{AE}(\theta, \phi; \boldsymbol{x}) = \frac{1}{N_b} \sum_{i=1}^{N_b} |D_\theta(E_\phi(x_i)) - x_i|^2. \tag{6.1}$$

Thus $E^{-1} \approx D$; the encoder and decoder transform to and from the latent space. When the AE networks are converged, we can encode all input data into the latent space representation, $E(X) = Z$, and the rest of the model will work entirely on the encoded data Z.

One problem that arises with the simple AE is that we do not have any control over the distribution of latent space encodings, and this is undesirable because we plan

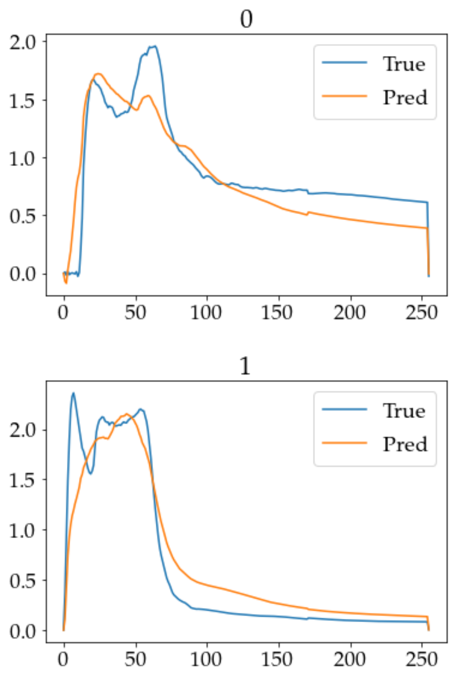


Figure 6.1. Reconstruction of two cross section evaluations using a linear PCA with 32 variables, out of an original 256 dimensional cross section vector. The linear PCA does not do a good job representing our data because there are too many variations.

to use these encodings as input for another model. Having a smooth yet efficient encoding is a high priority, and so we turn to a more appropriate formulation of the encoding problem: variational Bayes (VB) and in particular the variational autoencoder (VAE) [73]. VB refers to statistical methods which approximate a probability distribution with a conditional (or joint) one, often using Kullback-Leibler divergence (KLD) as a measure of closeness. In the present case, we want to approximate the distributions $P_{E(\phi)}(z|x)$, the encoding, and $P_{D(\theta)}(x'|z)$, the decoding. The VAE is essentially a mapping of the VB problem onto the autoencoder neural network structure: the problem has a robust statistical foundation and we introduce deep neural networks to learn the probability distributions. The VAE maps the latent variables to parameters of Gaussian distributions rather than points: this is called the *reparameterization trick*, mapping each input to a mean and variance. This means the latent representation of x used in the VAE is not $E(x) = z$ but rather $E(x) = (\mu_z, \sigma_z^2)$.

When training the VAE, the latent representation is a random sample from $\mathcal{N}(\mu_z, \sigma_z^2)$. Each latent space representation z is mapped to output x' according to the probability distribution $P(E(x)) = \mathcal{N}(\mu_z, \sigma_z^2)$. When using the VAE for encoding after training, the random noise is removed, so the encoding is deterministic $E(x) = \mu_z$, and so we typically denote the encoded data $\mu_z = z$ as before.

The VAE loss function $\mathcal{L}_{VAE}$ is a combination of the AE loss and VB loss, reconstruction error and Kullback-Leibler divergence (KLD) respectively,

$$\mathcal{L}_{VAE}(\theta, \phi; \vec{x}) = -\frac{1}{N_b} \sum_{i=1}^{N_b} KLD\left[P_{E(\phi)}(z|x_i) || \mathcal{N}(\mu_z, \sigma_z^2)\right] + \mathcal{L}_{AE}(\theta, \phi; \vec{x}). \tag{6.2}$$

The function $KLD(\cdot||\cdot)$ is lower when the probability distributions on either side of the $||$ are similar and large when they are very different. In our case, where the comparison is made to a Gaussian distribution, this function reduces nicely:

$$KLD\left[P_{E(\phi)}(z|x) || \mathcal{N}(\mu_z, \sigma_z^2)\right] = \frac{\beta}{2}\left[1 - \log \sigma_z^2 - \mu_z^2 - \sigma_z^2\right]. \tag{6.3}$$

The scale of the KLD loss term is parameterized by the number β and in practice this is not a static quantity. With a sufficiently complex neural network it is easy to find (via numerical optimization) a minimization of KLD with large reconstruction error: in other words, the latent space representations are Gaussian but E and D are not good approximations to the distributions they should be learning. To help this, we employ a technique called cyclic annealing [46] which schedules the KLD β to oscillate from 0 to 1 a few times before ultimately fixing to 1. This resulted in good convergence and is especially easy to implement in code.

Lastly, our VAE layers are fully convolutional. A convolutional neural network is likely superior to the usual densely-connected feed-forward neural network for data dominated by local correlations, especially for a use-case like encoding. This is especially true in very complicated problems where larger neural networks are employed; for a fixed number of layers the convolutional network has fewer parameters than the dense, and thus is generally easier to train. The combination of convolutional layers and learning VB results in an efficient mapping to the latent space: it appears that large-scale features in the cross section are provided large σ_z and thus are smoother with respect to the latent space, and smaller-scale features are given a smaller σ_z. So, the latent space appears to reflect naturally the trends present in the data.

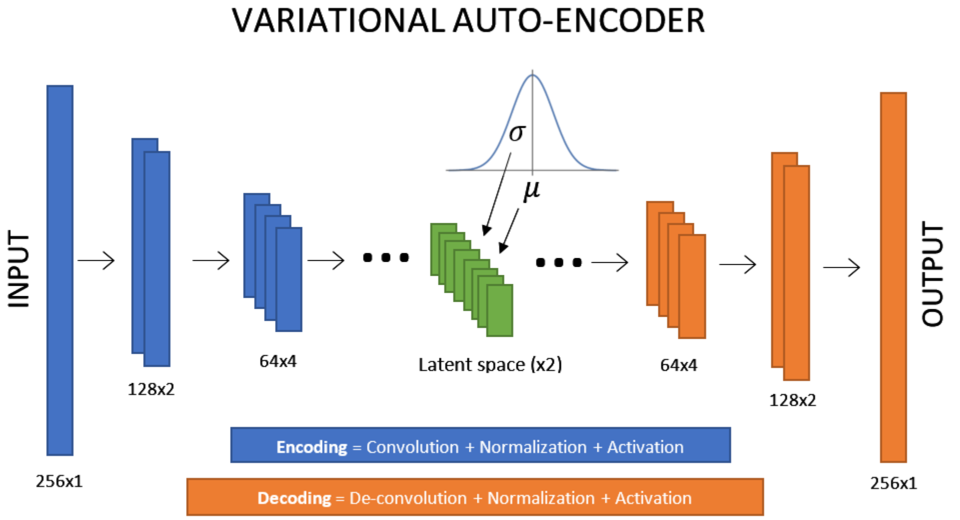


Figure 6.2. A schematic of the deep convolutional variational auto-encoder. The green central layer is the *latent layer* where variables are reparameterized to the mean and variance of a Gaussian distribution.

6.1.4 Generative adversarial networks

We employ a *generative adversarial network* (GAN) to learn trends in the cross sections. Before introducing the GAN, it is worth mentioning that deep generative learning (and the subset of it which is adversarial) as a field of study has expanded and matured significantly in the years since the first GAN work was published [56]. In the present research we have developed our architecture based on experience with the relevant physics, but there are *many* different types of GAN models one may consider

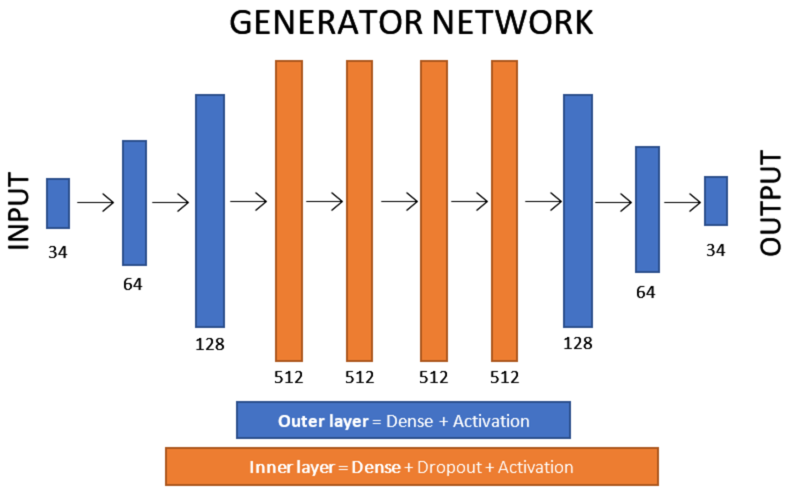


Figure 6.3. Diagram of generator network structure in the cycleGAN. The network shown the largest one of several configurations tested.

for this task [70, 104, 54, 63, 141]. We cannot possibly give a comprehensive description of all the ways a GAN may be applied to nuclear data. Our model is the result of over two years of ongoing study and development, but it is not a perfect solution, and it is our hope that this work will prompt other researchers to develop more creative and effective methods for these sort of problems.

6.1.4.1 Simple GANs

Generative models are a family of machine learning techniques which are designed to learn, and thus sample, a probability distribution. The VAE discussed in the last section is a generative model: the decoder network can be fed a normal random variable and generate random samples of the learned distribution. What sets the GAN apart is a modification of the generating network's loss function: rather than relying on the latent space distribution of data, one introduces an entirely separate network to learn the definitive trends in training data and penalize the generating network when the output does not match those trends. As such, the adversarial nature of the model allows for a generalization of the statistical requirements we had in the VAE. The simplest GANs consist of two networks: one generator G and one discriminator D, both

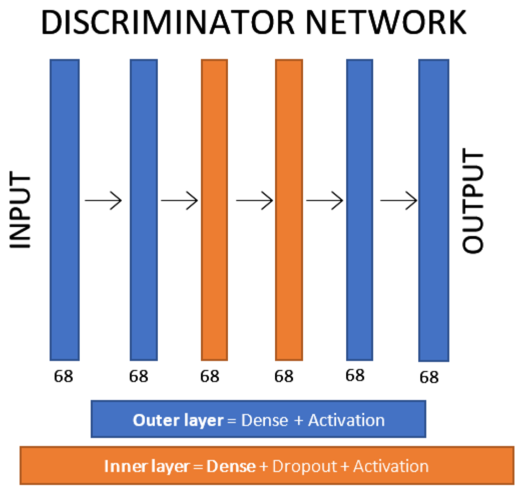


Figure 6.4. Diagram of discriminator network structure in the cycleGAN.

acting on data $x \in X$. The generator learns a mapping from a standard normal to the target distribution, and the discriminator scores the validity of generated data, $D(x') = P(x' \in X | x \in X)$. This type of GAN is useful for fully unsupervised learning; that is, the generator learns the relevant probability distribution without any labelled inputs (aside from the labels for $x \in X$ and $G(x) \notin X$, which are supplied to the discriminator, but these are trivial to generate). During training, the weights of networks G and D are optimized with respect to a loss function, and the networks compete with one another in what is (notoriously) tantamount to finding a Nash equilibrium; methods for reliable GAN training is an open area of research. Training the system results in a generator which can "convincingly" (according to the Discriminator) create new samples x', the correlations of which closely resemble those of the training data: $x' \tilde{\in} X$ (i.e. x' closely resembles an element of X, and/or $x' \sim P(X)$). We denote this simple GAN as the pair $\{G, D\}$. We can write the networks as explicit functions of weights θ as $G = G(\theta_G)$ and $D = D(\theta_D)$ for clarity.

The loss function for the simple GAN is different for generator and discriminator, but both can be expressed in terms of two expectation values ($\mathbb{E}$),

$$v = \mathbb{E}_{x \sim P(X)} \log(D(x)) + \mathbb{E}_{x \sim P(G)} \log(1 - D(x)). \tag{6.4}$$

The distribution $P(G)$ means the distribution which the generator samples. The respective loss functions are $\mathcal{L}_G(\theta_G) = v(\theta_G)$ and $\mathcal{L}_D(\theta_D) = -v(\theta_D)$. The discriminator loss is low when it correctly identifies both true data and generated data, thus both terms are large. The generator loss only depends on the second term, and is low when the discriminator assigns a high probability to the generator outputs; thus we might say the discriminator is being "fooled", classifying the generated data as real.

Although the ideal outcome is to find a Nash equilibrium, this can be very difficult and so training a discriminator to effectively classify true/fake data, and subsequently training a generator to "fool" it can be a useful approach. It is common for GANs to be trained this way, then repeat the process by training the discriminator a bit further. While both networks are training they compete and which has the advantage can oscillate; this process can continue for a long time without clear indication of making progress, but the model may still be useful in this state despite not being fully converged.

In practice, we may express this loss function in Eq. 6.4 in a slightly different way. One often uses a loss function like binary cross-entropy (BCE) to evaluate probabilities, which the discriminator network emits. BCE is the preferred loss function when the classifier network ends with a single sigmoid neuron: it compares a label probability y with a predicted probability p, and outputs a number that decreases as $y \to p$. (In practice, one may design a classifier which emits a so-called *logit* value on $(-\infty, +\infty)$ and a modified version of the BCE to handle logits instead of probabilities. This is equivalent but may allow for better convergence in some cases.) The BCE loss term in terms of p and y is

$$BCE\,[p, y] := -\left[y \log(p) + (1 - y) \log(1 - p)\right]. \tag{6.5}$$

BCE can also be evaluated for a set of inputs, as in batch training, simply by averaging the individual BCE values for each pair. In terms of BCE for a single batch of N_b data points, the losses are

$$\begin{aligned}\mathcal{L}_D(\theta_D) &= \frac{1}{N_b}\sum_{i=1}^{N_b}\left[BCE[D(x_i),1] + BCE[D(G(z_i)),0]\right] \\ \mathcal{L}_G(\theta_G) &= \frac{1}{N_b}\sum_{i=1}^{N_b} BCE[D(G(z_i)),1],\end{aligned} \tag{6.6}$$

which I find a little more intuitive than Eq .6.4. Ultimately, the GAN learns to approximate the probability distribution of the data $P(X)$ and the evaluation of the generator on random noise $G(z)$ approximates sampling from $P(X)$. To better adapt the model to our nuclear data problem, we consider a slightly more advanced form called the *cycle-consistent* GAN.

6.1.4.2 Cycle-consistent GANs

Zhu et al [140] demonstrated the effectiveness of *cycle-consistent* GANs (cycleGANs) for image-to-image translation: two GANs $\{G, D_G\}$ and $\{F, D_F\}$, may be used to map between two distinct probability distributions, for $x \in X$ and $y \in Y$, as $G(x) = y' \tilde{\in} Y$ and $F(y) = x' \tilde{\in} X$. The discriminator networks D_G and D_F score the validity of the two resulting candidates: $D_G(y') = P(y' \in Y | y \in Y)$ and $D_F(x') = P(x' \in X | x \in X)$. The cycleGAN loss functions add two new terms to the loss functions in Eq. 6.6. First, cycle loss ensures that applying the generators in succession returns the input (i.e. the convolution of co-inverse generators is the identity function). For a batch of N_b data points, cycle loss is

$$\mathcal{L}^G_{\text{cycle}} = \frac{1}{N_b}\sum_{i}^{N_b}|X_i - F(G(X))_i| + \frac{1}{N_b}\sum_{i}^{N_b}|Y_i - G(F(Y))_i|. \tag{6.7}$$

One such application is *style-transfer* [70] where G learns to transform X so that correlations match those of Y, and F learns the reverse. Style-transfer employs the constraint that generators be idempotent ($G^2 = G$): $G(y) \tilde{\in} Y$ and $F(x) \tilde{\in} X$, because the generators should act as projections into the respective spaces (that is, $G(x) \tilde{\in} Y$ but $G(G(x)) \tilde{\in} Y$ as well). The term "cycle-consistent" refers to an additional constraint that each generator be the inverse of its partner: $G(F(y)) = G(x') = y'$ and $F(G(x)) = F(y') = x'$. In practice, this helps ensure the generators learn systematic trends rather than simply "memorize" the training data. We can denote such a cycleGAN system as the set $\{G, F, D_G, D_F\}$.

6.1.4.3 The physics-informed cycleGAN

Our method for transforming cross-sections is a modified version of the cycleGAN for style-transfer: we rely on cycle-consistency, but there are some important differences. First, our data is paired, so we do not use the the projection constraint present in style-transfer. Having generators be projectors makes sense in style-transfer because there are two distinct distributions being mapped between, but we want the generators to learn changes based on changing proton/neutron numbers which should always (probably) have an affect on the cross section. Our generators map between data points within a single domain, and consequently we have a single shared discriminator $D_G = D_F = D$. We denote the modified cycleGAN as $\{G, F, D\}$.

Second, we introduce a new loss term: *target loss*, equal to the mean absolute error between the model and target cross section. Without target loss, the total loss function would have many minima not relevant to the physical solution, so the introduction of this term is referred to as *physics-informing*. The total loss function is thus biased toward a solution to the adversarial GAN problem which correctly reproduces the physics. We introduce a tunable parameter λ_{target} for controlling the weight of target loss, which is scheduled to be large at the beginning of training so the optimizer converges quickly, then dropped later in training as to not overfit. Target loss is

$$\mathcal{L}^{G}_{\text{target}} = \frac{1}{N_b}\sum_{i}^{N_b}|X_i - F(Y)_i| + \frac{1}{N_b}\sum_{i}^{N_b}|Y_i - G(X)_i|. \tag{6.8}$$

The overfitting problem is subtle, in part because it is quite different from overfitting in simple regression models (which is very well understood). A typical regression problem abides the Gauss-Markov assumptions: errors between model and target have mean zero and a diagonal covariance matrix $C_{ii} = \sigma^2$ with the same variance value for each observation. This assumption does not hold for the present approach: cross section evaluations are in some cases much more detailed than others, different evaluations may involve different theoretical models and experimental factors, and the library may even include some mistakes. This makes it very difficult to make any assumptions about the form of the covariance matrix of errors. What we can do, however, is control overfitting via model complexity, regularization, and by tuning λ_{target}.

We judge overfitting based on evaluations for validation data, as in a typical regression problem: our assumption is that when errors on validation data are approximately equal to those on training data, *locally*, the probability of having overfit is low. The converse however is not true: a low probability of local overfitting does not necessarily mean that errors on validation data are approximately equal to those on nearby training data. The TENDL library has many individual cross section evaluations which have been finely tuned to experimental data, while surrounding cross sections have not, thus if that cross section is held out for validation we cannot always expect the model to predict those details. In short, the model cannot predict a systematic relation between cross sections that is not present in the training data.

Regularization of the neural networks is implemented in two ways: dropout, and label noise. Dropout is used in generators and discriminator. During training, upon each iteration, some percentage of neurons are randomly chosen to turn off, outputting zero. This makes network predictions more robust, since no single neuron can be relied on all of the time, and it makes it much less likely that the network will “memorize” data. We use a dropout probability of 50% on the inner hidden layers of the generators and 25% for the discriminator; hidden layers near the input and output do not use dropout (see Figs. 6.3 and 6.4). Label noise is a regularization tool which works on the discriminator; rather than using labels 0 or 1 for all training data, a small random ϵ is chosen and added/subtracted to the label accordingly. That is, upon every iteration we choose $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ and then replace $0 \to 0 + |\epsilon|$ and $1 \to 1 - |\epsilon|$. The standard deviation σ_ϵ is 0.1 in the present application (although, this value was not arrived at via any optimization, and in general such an exploration may be warranted). This has the effect of smoothing the discriminator predictions around the limits, keeping the discriminator from assigning probabilities “overconfidently”, so to speak. It is generally common to have label noise applied to only one label (i.e. either 0 or 1, not both), however in the present work we found applying the noise to both gave better results.

6.1.4.4 Challenges with the convolutional GAN

Our initial attempts at this project involved not the separate convolutional VAE and dense GAN, but rather a single convolutional GAN which was intended to learn both latent space encodings and transforms of cross sections. This construction was difficult to work with, and although the exact reasons for difficulty are unclear, we can certainly learn some things by comparing that model to our final version. In the old model, the generator of the cycleGAN followed a U-net design [112] illustrated in Fig. 6.5. The U-net design has shown success in GAN models on 2D image data, so our assumption was that 1D cross section data is similar enough that we could use a 1D version of the same model structure. The loss function was the same as that of our dense GAN; the discriminator network followed the design of a simple multi-layer convolutional classifier. However, when training it we found the U-net model to have poor convergence properties and especially was difficult to control overfitting while maintaining good predictions, and in particular to optimize the coefficient of target loss. If the coefficient λ_{target} is too large then the model would overfit and predictions would be unreliable, and if too small the model would not converge (i.e. predicted cross sections were very noisy and would not match the training set).

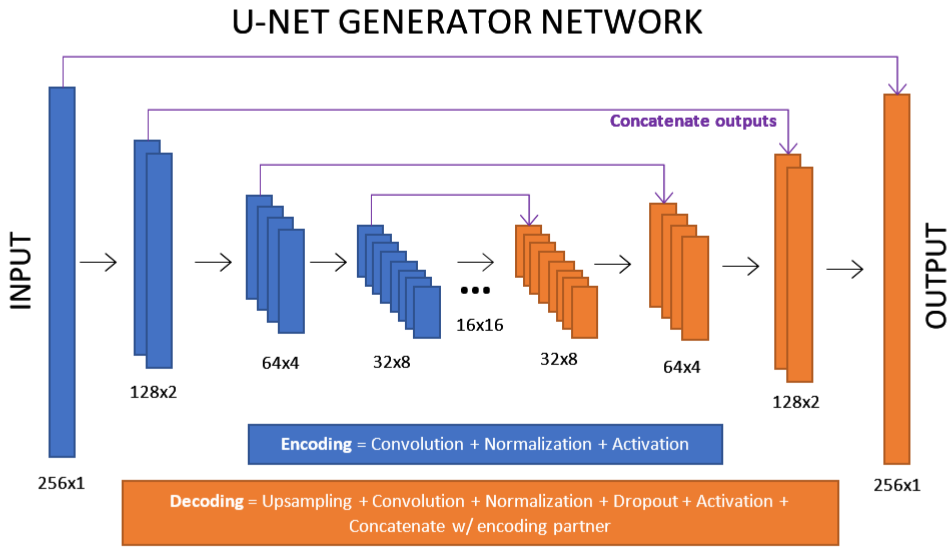


Figure 6.5. Diagram of U-net style generator network used in our initial attempts but ultimately was too difficult to optimize for this particular problem.

An interesting physics connection can be seen in comparing the results of the old model with the new. Since the old model would not converge fully, it could not learn systematic trends across the chart, and in particular Fig. 6.6 shows that error increases around $N = 50$ and 82, which are magic numbers. This indicates that the model had not learned to incorporate changes in the cross sections that are due to shell structure. These errors are not present in final versions of our model.

This is not to say that a 1D U-net style cycleGAN is in general a poor design, only that in our particular implementation for this problem, it did not give good

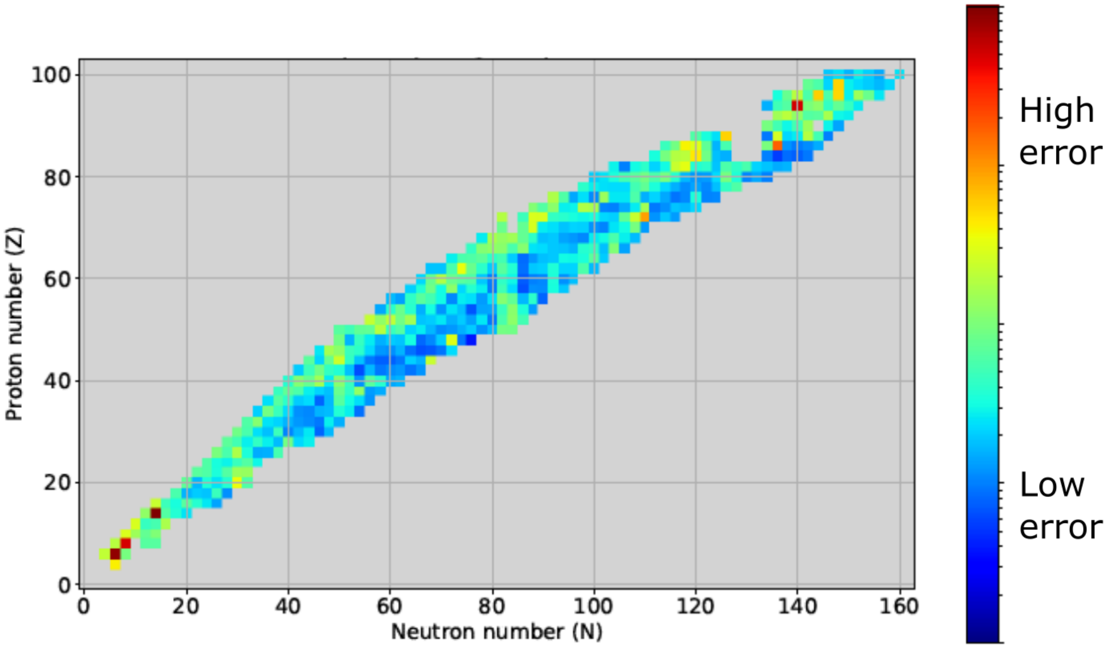


Figure 6.6. Average error of local predictions using the U-net GAN model, which was prone to overfitting, unstable, and ultimately abandoned in favor of the VAE+GAN system. Note that higher errors are clearly visible around magic numbers $N = 50, 82$, indicating the model has not learned how cross sections are dependent on shell structure.

results. Whether that model could be successful in some applications may be a worthy question to pursue, but that is beyond the scope of this research.

6.2 The physics-informed cycleGAN model for nuclear cross sections

We must invent some notation to clarify this section. First, for individual directions on the chart (cardinals + diagonals) we use the index $d \in \{\uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow, \leftarrow, \nwarrow\}$. Second, we use the index a for "axis", meaning each co-inverse pair of directions, so $a \in \{\updownarrow, ⤢, \leftrightarrow, ⤡\}$.

Our full model consists of four cycleGANs, one for transforming along each axis across the chart of nuclides: the $\leftrightarrow$ axis corresponds to adding/subtracting neutron pairs, the $\updownarrow$ axis to adding/subtracting proton pairs , the ⤢ axis to adding/subtracting both proton and neutron pairs (a.k.a. isoscalar), and the ⤡ axis to swapping a proton pair for neutron pair and vice versa (a.k.a. isovector). Denoting each cycleGAN set as M, we have the following four separate models. The subscript of generators G, F indicate the transform they have learned.

$$\begin{aligned}
M_{\leftrightarrow} &= \{G_{(N,Z)\to(N+2,Z)}, F_{(N,Z)\to(N-2,Z)}, D_{\leftrightarrow}\} \\
M_{\updownarrow} &= \{G_{(N,Z)\to(N,Z+2)}, F_{(N,Z)\to(N,Z-2)}, D_{\updownarrow}\} \\
M_{\text{⤢}} &= \{G_{(N,Z)\to(N+2,Z+2)}, F_{(N,Z)\to(N-2,Z-2)}, D_{\text{⤢}}\} \\
M_{\text{⤡}} &= \{G_{(N,Z)\to(N+2,Z-2)}, F_{(N,Z)\to(N-2,Z+2)}, D_{\text{⤡}}\}
\end{aligned} \tag{6.9}$$

However, for simplicity we rename the generator networks to T_d with a subscript arrow indicating the direction of transformation on the chart of nuclides.

$$\begin{aligned}
M_{\leftrightarrow} &= \{T_{\rightarrow}, T_{\leftarrow}, D_{\leftrightarrow}\} \\
M_{\updownarrow} &= \{T_{\uparrow}, T_{\downarrow}, D_{\updownarrow}\} \\
M_{\text{⤢}} &= \{T_{\nearrow}, T_{\swarrow}, D_{\text{⤢}}\} \\
M_{\text{⤡}} &= \{T_{\searrow}, T_{\nwarrow}, D_{\text{⤡}}\}
\end{aligned} \tag{6.10}$$

We standardize cross sections to lie on a fixed energy domain of $0 - 30$ MeV in 256 bins, and normalize them individually to a maximum of 1. This way, we factor out the task of predicting amplitudes and just focus on the shape of the curve. In principle the scales could be included in the predictive model in future development, but that would likely require some non-trivial modification of the VAE encoding process. Let Σ with no subscript mean the set of all cross section evaluations, $\Sigma = \{\sigma_{(N,Z)}\}$. For each cycleGAN, a set of cross sections Σ_a is prepared by *pairing* cross sections along the relevant axis. Sets of pairs of cross sections are denoted as follows.

$$
\begin{aligned}
\Sigma_{\leftrightarrow} &= \{(\sigma_{(N,Z)}, \sigma_{(N+2,Z)})\} \\
\Sigma_{\updownarrow} &= \{(\sigma_{(N,Z)}, \sigma_{(N,Z+2)})\} \\
\Sigma_{\neswarrow} &= \{(\sigma_{(N,Z)}, \sigma_{(N+2,Z+2)})\} \\
\Sigma_{\nwsearrow} &= \{(\sigma_{(N,Z)}, \sigma_{(N+2,Z-2)})\}
\end{aligned}
\tag{6.11}
$$

Preparation of data for the GAN takes two steps. First, we use the VAE to encode each cross section, $E(\sigma_{(N,Z)}) = z_{(N,Z)}$. Let Z with no subscript mean the set of all encoded cross sections, $Z = \{z_{(N,Z)}\}$. The sets of all pairs of encoded cross sections are denoted similarly with Z_a as follows.

$$
\begin{aligned}
Z_{\leftrightarrow} &= \{(z_{(N,Z)}, z_{(N+2,Z)})\} \\
Z_{\updownarrow} &= \{(z_{(N,Z)}, z_{(N,Z+2)})\} \\
Z_{\nearrow} &= \{(z_{(N,Z)}, z_{(N+2,Z+2)})\} \\
Z_{\nwsearrow} &= \{(z_{(N,Z)}, z_{(N+2,Z-2)})\}
\end{aligned}
\tag{6.12}
$$

Second, each vector $z_{(N,Z)}$ is appended with normalized values of proton and neutron number N, Z. We can denote the resulting form of our data as $\tilde{Z}_a$ which is the pairs of encoded cross sections with normalized proton/neutron numbers appended. As such, the dimension of the generator input/output is equal to the VAE latent dimension plus 2.

The discriminator networks have a different job than in a cycleGAN used for style-transfer. Rather than accept a single data vector $\tilde{z}'$ and determine $P(\tilde{z}' \in \tilde{Z})$, the discriminator D_a accepts an ordered pair $(\tilde{z}'_1, \tilde{z}'_2)$ and learns $P((\tilde{z}'_1, \tilde{z}'_2) \in \tilde{Z}_a)$. This has several benefits. The order of the pair matters, so discriminator D_a should assign 1 to (that is, accept) only pairs of vectors which together match the distribution of variables along axis a in the data.

6.3 Optimization

The loss function for each cycleGAN has four main components: cycle-consistency losses, prediction losses, and the two adversarial losses. Without loss of generality, we write the loss function for a single cycleGAN $\{G, F, D\}$ as follows.

$$\mathcal{L} = \lambda_{\text{cycle}}\mathcal{L}^{G}_{\text{cycle}} + \lambda_{\text{target}}\mathcal{L}^{G}_{\text{target}} + \lambda_{\text{fake}}\mathcal{L}^{D}_{\text{fake}} + \lambda_{\text{real}}\mathcal{L}^{D}_{\text{real}} \tag{6.13}$$

The coefficients are determined simply by trial and error, we found that $\lambda_{\text{cycle}} = 1, \lambda_{\text{fake}} = \lambda_{\text{true}} = 1$ worked well for this case. The target loss coefficient λ_{target} is set larger (e.g. 5-10) at the beginning of training, then scheduled to decrease to 1. This allows for us to bring the system to the relevant region of parameter space early, but ultimately relax the constraint that predictions match very closely with TENDL data.

We use a translated and scaled sigmoid function to do schedule changes in values while training. The sigmoidal schedule function,

$$S_{m,s}(t) = \begin{cases} S_i & t < t_i \\ S_i + \frac{S_f - S_i}{1+\exp[-(t-m)/s]} & t_i < t < t_f \\ S_f & t > t_f \end{cases} \tag{6.14}$$

begins at S_i then in the window between times t_i and t_f gradually approaches the value S_f. The parameters m and s control the translation and scale (in time) respectively. This function is used to smoothly schedule the decrease in learning rate and λ_{target}. The initial and final epochs t_i, t_f depend on convergence, but are on the order of 10,000 epochs.

We found that our model, as was the case in the original cycleGAN paper, converged more quickly using a bacth size of $N_b = 1$. This is quite unlike many other machine learning applications where batch size may be set to several dozen. This is mostly a practical decision, but is well known in the GAN community; one generally gets much more accurate predictions with a batch size of 1 [105]. However, this decision can make training relatively slow, since epoch time scales like $O($ # data points / batch size)90.

Development and training was carried out on heterogeneous architecture at Lawrence Livermore National Laboratory with IBM POWER9 chips: per training session, the CPU creates the relevant data environment including the cycleGAN model with appropriate hyperparameters and sends it to the on-board GPU for weight optimization. Optimization was performed using ADAM [72] with a moving average wrapper [137]. ADAM already includes a momentum contribution, but the moving average means the weights are even less sensitive to small changes in the loss function, which is very helpful because gradients of adversarial losses can be very noisy.

We have prepared two data sets for experimentation: dataset $\mathcal{A}$ has a subset of 20 cross sections distributed along the length of the chart held out for validation, and dataset $\mathcal{B}$ has a 3x3 region with ^{170}Yb at the center, 9 cross sections total, held out as a validation set. Examples of loss histories for dataset $\mathcal{A}$ and $\mathcal{B}$ are shown in Figs. 6.7 and 6.8 respectively. In these experiments, we have assisted the adversarial nature of the model slightly by introducing a *freezing* value for each loss term. When the discriminator loss drops below a freezing point, it's weights are fixed to allow the generator to catch up. The generators similarly have their own freezing value. Evidence of this can be seen in the loss histories: where sudden spikes appear in generator loss terms, corresponding with a sudden drop in discriminator loss, this means the discriminator has reached it's freezing point and we allow the generator to catch back up.

The loss terms $\mathcal{L}_{\text{target}}$ and $\mathcal{L}_{\text{cycle}}$ tend to follow one another very closely, and this makes sense since both are controlled by the generator completely. The discriminator scores are shown in the dot-dash lines, and one can clearly see those loss terms in opposition with the generator's adversarial losses, shown in dotted lines. That is, the generators and discriminator are competing, as they should be. Also apparent is that

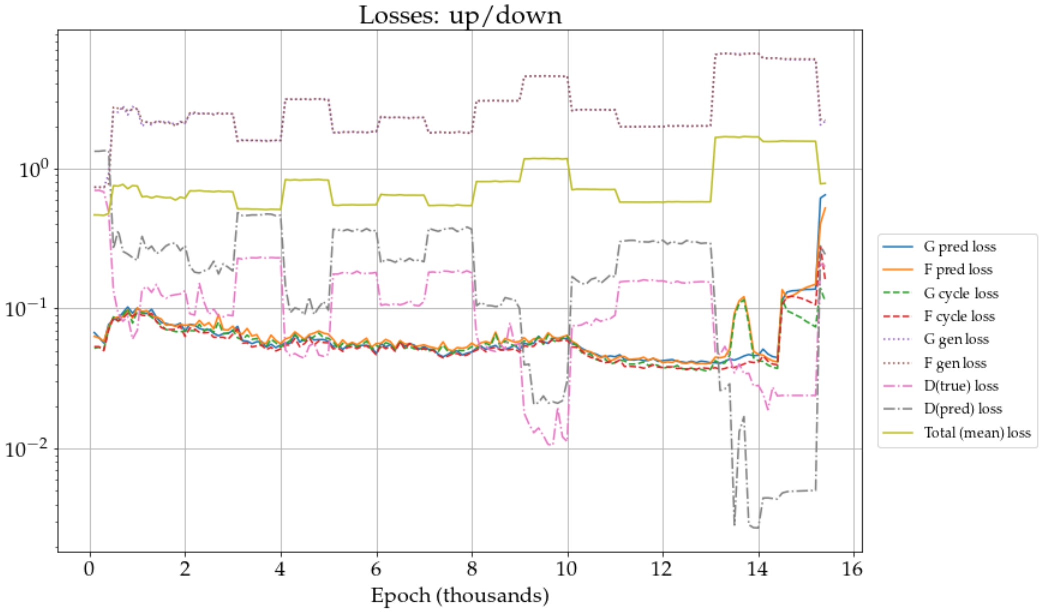


Figure 6.7. Loss history for dataset $\mathcal{A}$ shown from 0 to 16,000 epochs.

some "victories" seen in the discriminator have little effect on the generators, while others have a very large effect.

6.4 Predicting cross sections

There are several ways to use the model to make predictions. The simplest use of the model is to predict cross sections of direct neighbors; we call this a local prediction. For local predictions, beginning from a cross section σ_0, the prediction of neighbor in the d direction is $\sigma' = D(T_d(E(\sigma_0)))$. Each tile shows the TENDL cross section in black and the predictions from direct neighbors in colored dashed lines. We can judge how well the model has fit the data by comparing predictions of validation data with the training data. Our assumption is that if errors on training data are on the same scale as errors on validation data, then the model is unlikely to be overfit.

Figures 6.11 , 6.12, 6.13, and 6.14 show regions of local predictions using dataset $\mathcal{A}$. Figures 6.15, 6.16, 6.17, 6.18, 6.19, 6.20, 6.21, show regions of local predictions using dataset $\mathcal{B}$. The heatmaps for datasets $\mathcal{A}$ and $\mathcal{B}$, shown in Figs. 6.10 and 6.9 respectively,

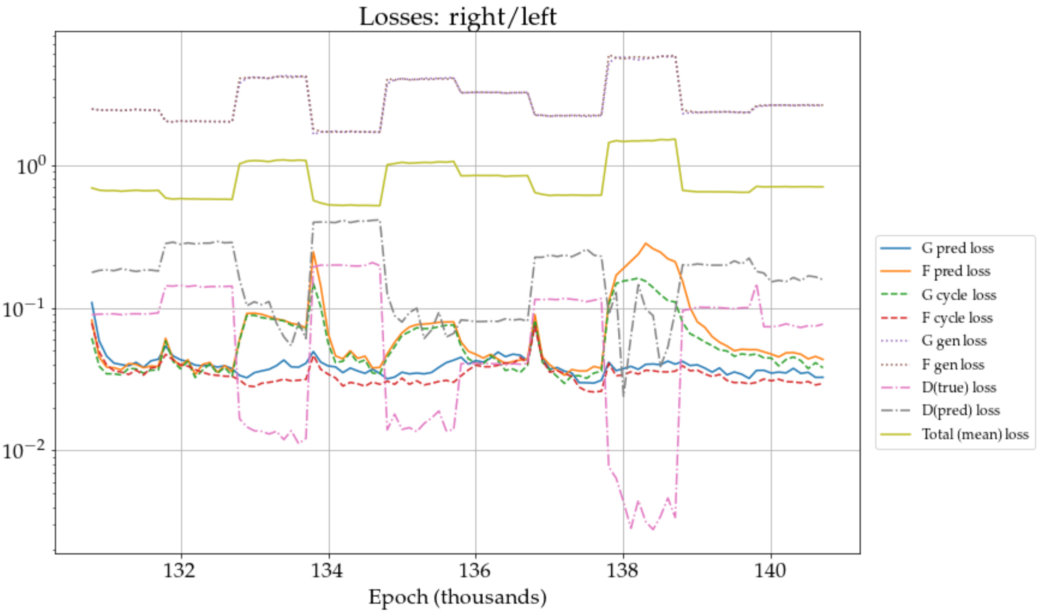


Figure 6.8. A portion of loss history for dataset $\mathcal{B}$, shown from 130 to 140 thousand epochs.

show a global picture of local predictions, with average MAE values represented as colors, and each tile is a single cross section. Validation data are outlined in light green.

Since each generator learns changes in cross sections according to a particular physical process, we can make predictions in stepwise paths across the chart, plugging each consecutive prediction back into the system as input. This process is not restricted to the region of training data, and so we can use what trends the model has learned inside the training region to extrapolate beyond. To do so, we compute $\sigma' = D(T_\Delta(E(\sigma_0)))$ where T_Δ is a convolution of a sequence of T_d; the sequence Δ then constitutes a path across the chart. Consider for example $\sigma'_1 = D(T_{\downarrow\searrow\rightarrow}(E(\sigma_0)))$ and $\sigma'_2 = D(T_{\rightarrow\searrow\downarrow}(E(\sigma_0)))$. While our starting cross section is the same for both, σ_0, and both paths terminate on the same nuclide, the results will likely be different. The function T_d does not commute with $T_{d'}$ for different directions $d \neq d'$.

As another example, consider $\sigma'_1 = D(T_{\rightarrow\rightarrow}(E(\sigma_{(N,Z)})))$ and $\sigma'_2 = D(T_{\rightarrow}(E(\sigma_{(N+2,Z)})))$. These predictions begin at two different nuclides, (N, Z) and $(N+2, Z)$, but terminate on the same nuclide $(N+4, Z)$. If the networks have

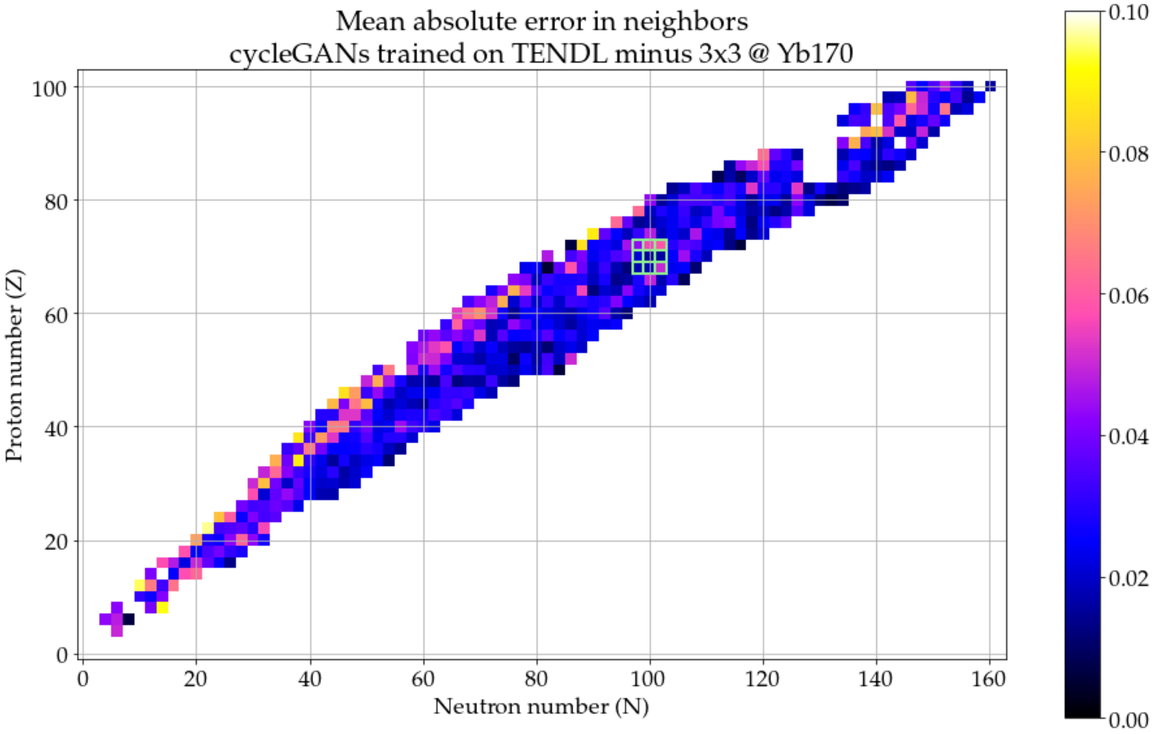


Figure 6.9. Average error from immediate neighbors, trained on data without 3x3 region around ^{170}Yb (outlined in green).

successfully learned the local correlations present in the training data, then it is likely that σ'_1 and σ'_2 will be similar, but there will inevitably be some error. When considering similar predictions to this with longer chains of predictions, errors can compound leading to instability.

The accuracy and stability of extrapolations appear to be dependent on the complexity of local trends around where the predictions are made. Each chain of predictions is stable for a number of steps, around 5-6 on average, but can be longer or shorter depending on trends locally present and the introduction of drastically anomalous data. Some examples of interpolation are given in Figs. 6.22, 6.23, 6.24, and 6.25. In these examples, I have shown a list of TENDL cross sections in one plot and a list of extrapolated GAN predictions in the other. The first predicted curve σ'_0 is simply the first cross section σ_0 encoded and decoded once, $\sigma'_0 = D(E(\sigma_0))$. The second, σ'_1, is the first cross section encoded, transformed once, and decoded: $\sigma'_1 = D(T_d(E(\sigma_0)))$,

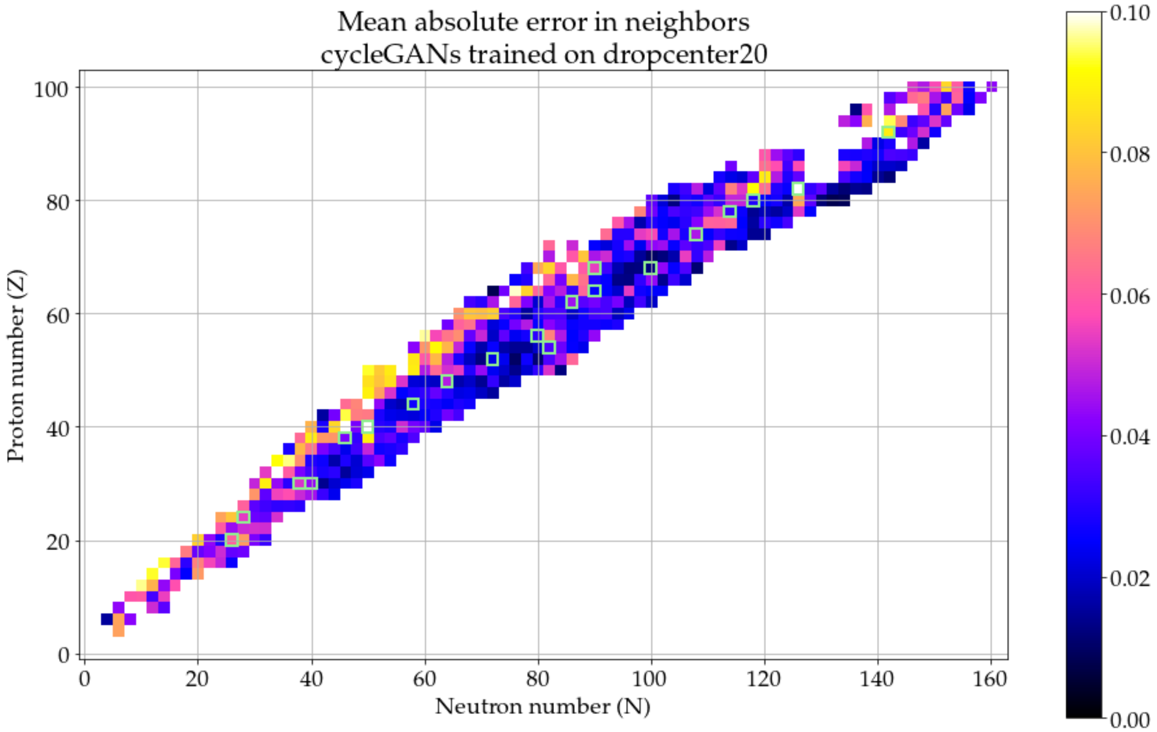


Figure 6.10. Average error from immediate neighbors, trained on data without a scattered subset of 20 nuclides (outlined in green).This model had been trained for far less time than that in the first example, hence the larger and more sporadic error around the edges.

where d is the direction of transformation. The third, σ'_2, is the first cross section encoded, transformed twice, and decoded: $\sigma'_2 = D(T_d^2(E(\sigma_0)))$. This proceeds for n steps: $\sigma'_n = D(T_d^n(E(\sigma_0)))$. Note that the only input to the GAN for the entire chain of predictions is σ_0.

6.4.1 Ensemble predictions

Leveraging the prediction capability, we can perform *ensemble predictions* for cross sections beyond the training set as well. Ensemble methods can be used in many different machine learning problems and are useful because the model(s) produce a set of predictions rather than just one, and thus we have a built-in measure of variability. In our GAN, this consists of computing many paths across the chart and predicting the cross section at a common final nuclide using the full set of transforms. The final set of predictions may be interpreted as representative of a Gaussian distribution and thus we present the ensemble prediction with mean and standard deviation.

Neural network models are commonly understood to have difficulty with extrapolations, at least without the proper modifications. Regularization, which are basically additional constraints, can help us to produce better extrapolations. In particular, *distribution learning* and constraints placed on latent space distributions of data can be helpful. The VAE, for instance, limits the distribution of latent variables using variational Bayes, and this results in a fantastically smooth latent representation (see [73] for more). Our working hypothesis is that distribution learning may be leveraged to achieve good extrapolation behaviour; this certainly helps achieve good results within this work but a proof of this concept does not to my knowledge exist.

Ensemble predictions are formed by a set of individual prediction chains. We first designate a target nuclide and a region of the chart for the starting points. Then, we compute all possible paths which begin within our region of interest and terminate on the target. An added constraint is that the paths do not "back-track"; that is, within each path, the distance to the target nuclide is monotonically decreasing with each step. This property ensures that we are not including undue errors in the ensemble.

Examples of ensemble predictions are shown in Figures 6.26, 6.27, 6.28, 6.29, and 6.30. Each figure shows the TENDL cross section evaluation in pink and the predicted ensemble in thin colored lines. The color corresponds to the nmumber of steps in the path leading to that prediction (see colorbar). So, Figures 6.26 and 6.27 show good predictions and a reasonable spread of errors. As can be seen in Fig. 6.27, short paths are generally more accurate than long paths. This makes sense since predictions from longer paths might accumulate errors which compound with each application.

Fig. 6.28 shows predictions for Cerium-158, which does not have a cross section evaluation in the TENDL library. Interestingly, we see the ensemble prediction is bimodal; that is, each prediction is centered around one of two curves. The first mode increases very fast at 0 MeV, then has a shorter secondary peak around 6 MeV (which is a very common feature in the inelastic neutron scattering channel). It corresponds to predictions from shorter paths (blue and green curves), and so it is likely more accurate. The second mode, which only has one major peak around 5 MeV, is created by longer paths (yellow and red curves), and thus is likely not as accurate as the other. We can thus inspect the ensemble prediction and provide a prediction for the cross section of Cerium-158, which does not have experimental measurements, and also provide a confidence band estimate according to the spread in the ensemble. However, one should be careful here: the ensemble is certainly not normally distributed, and so summarizing it with a mean and 1σ confidence band is inaccurate.

Lastly, I include Figures 6.29 and 6.30 to illustrate some ensemble predictions with large variability. Ensemble predictions for heavy nuclides, like Uranium-234 and Erbium-158, are likely not as accurate as those for lighter nuclides, because there are less data points for heavy nuclides in the training data (which can be seen as the chart gets thinner at the heavy end, nuclides have fewer around them). A future priority for this research may be to decide how to better summarize these ensemble predictions with large variance.

6.5 Conclusion

I have developed a deep generative machine learning model to learn intricate systematic trends in nuclear cross section evaluations in a subset of the TENDL library. The model uses well-established machine learning models, namely the variational auto-encoder and generative adversarial network, but we have made several important modifications to these models to suit the problem of nuclear data, and thus it is a novel development. The model is capable of accurate predictions of cross sections as particle pairs are added and subtracted from the target nucleus, at least within local regions on the chart of nuclides.

This can be used for interpolations and extrapolations, chaining together of predictions, and producing an ensemble of predictions for cross section evaluations. Ensemble predictions are very powerful as they naturally can be used to predict cross sections outside the library, and estimate confidence intervals.

Although the present model is perhaps not well suited for scaling to larger problems, such as data with non-even-even nuclides and including other reaction channels, we hope this research can inspire other researchers to bring creativity and insight to this problem.

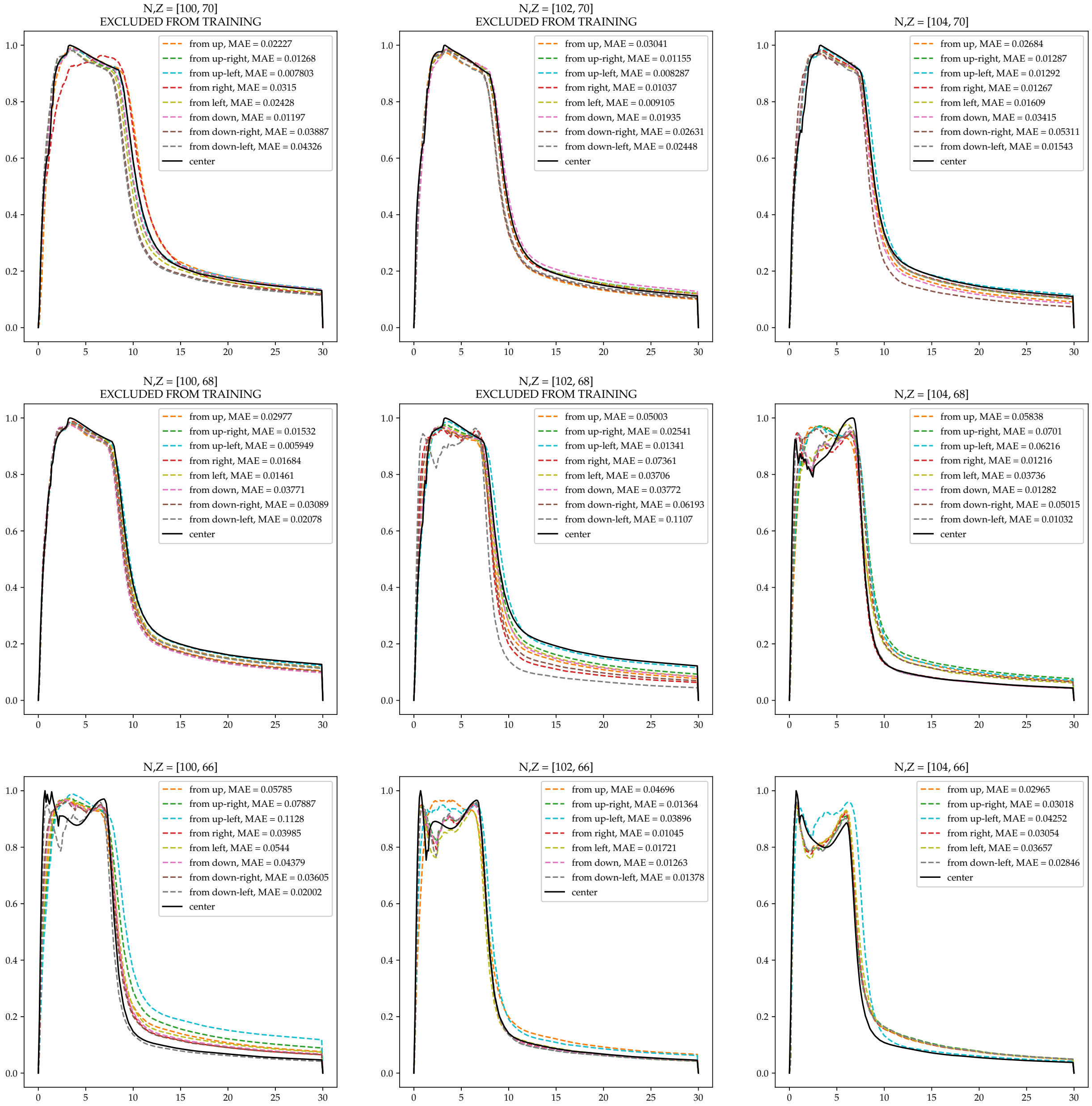


Figure 6.11. TENDL compared to local predictions of the VAE+GAN model. Cross sections which have been excluded from the training set are labelled. In each tile, the solid black line is the TENDL cross section evaluation and each colored dashed line is the prediction from a direct neighbor. The directions color-coding and mean absolute error are given in each legend.

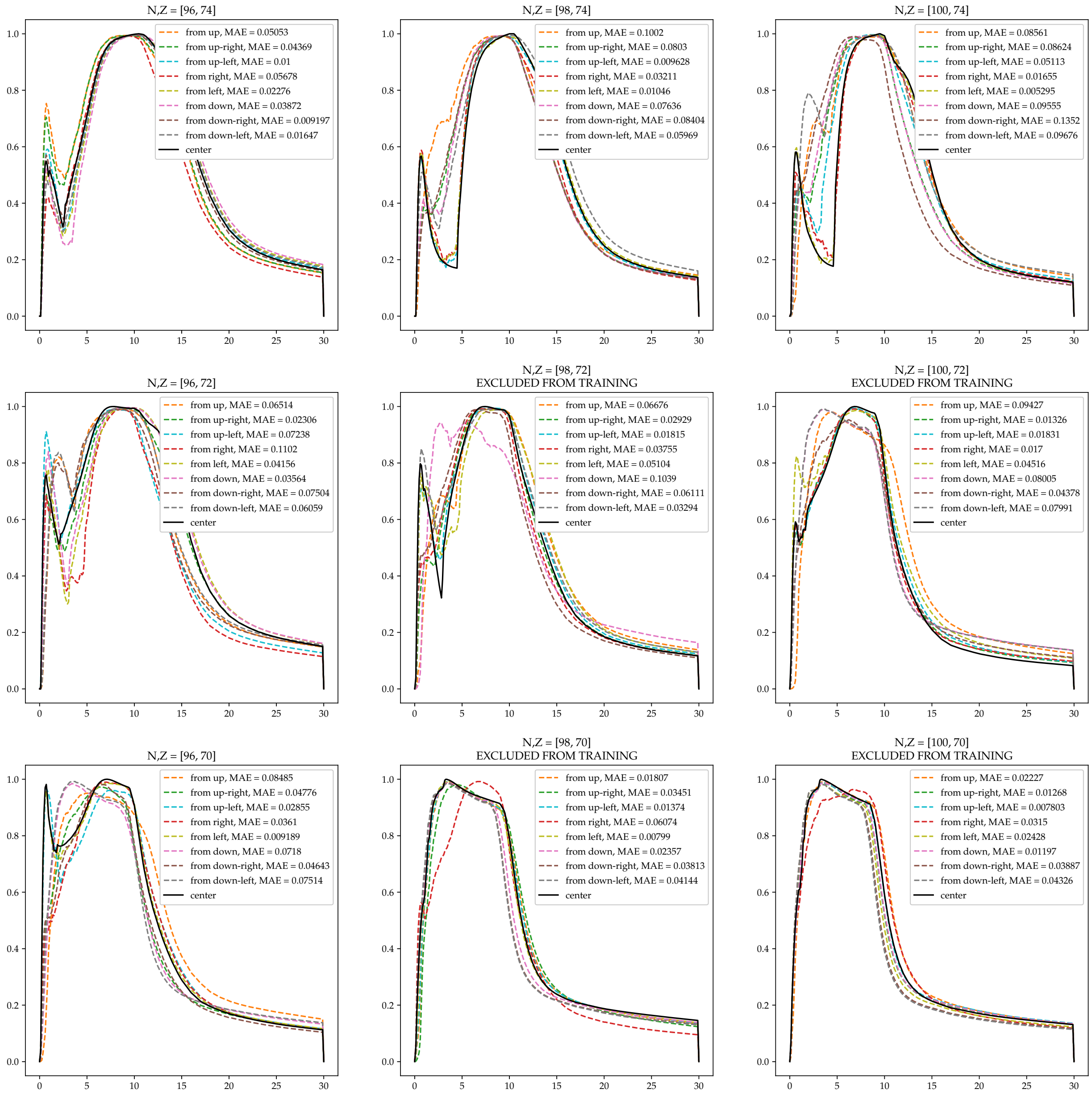


Figure 6.12. TENDL compared to local predictions of the VAE+GAN model. Cross sections which have been excluded from the training set are labelled. In each tile, the solid black line is the TENDL cross section evaluation and each colored dashed line is the prediction from a direct neighbor. The directions color-coding and mean absolute error are given in each legend.

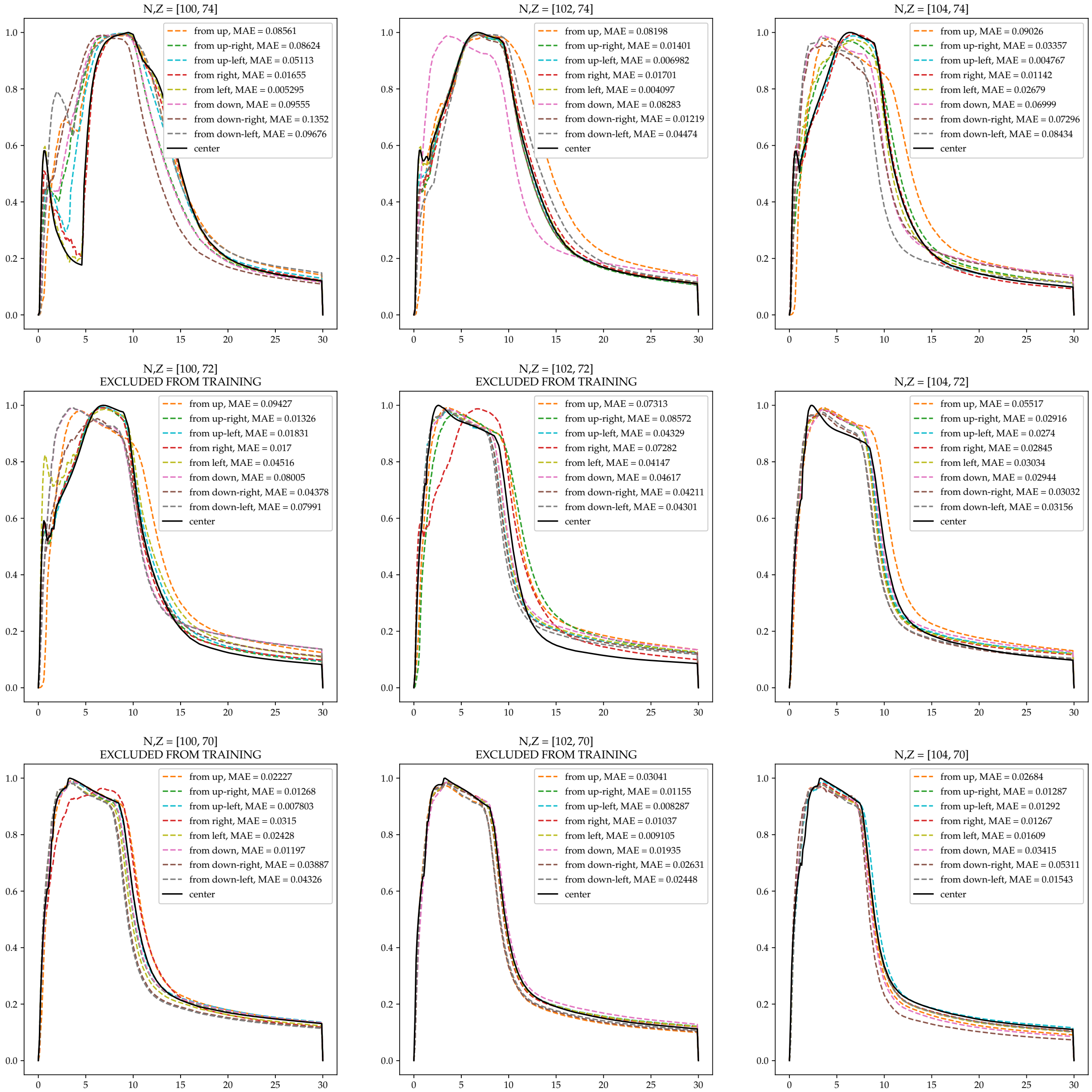


Figure 6.13. TENDL compared to local predictions of the VAE+GAN model. Cross sections which have been excluded from the training set are labelled. In each tile, the solid black line is the TENDL cross section evaluation and each colored dashed line is the prediction from a direct neighbor. The directions color-coding and mean absolute error are given in each legend.

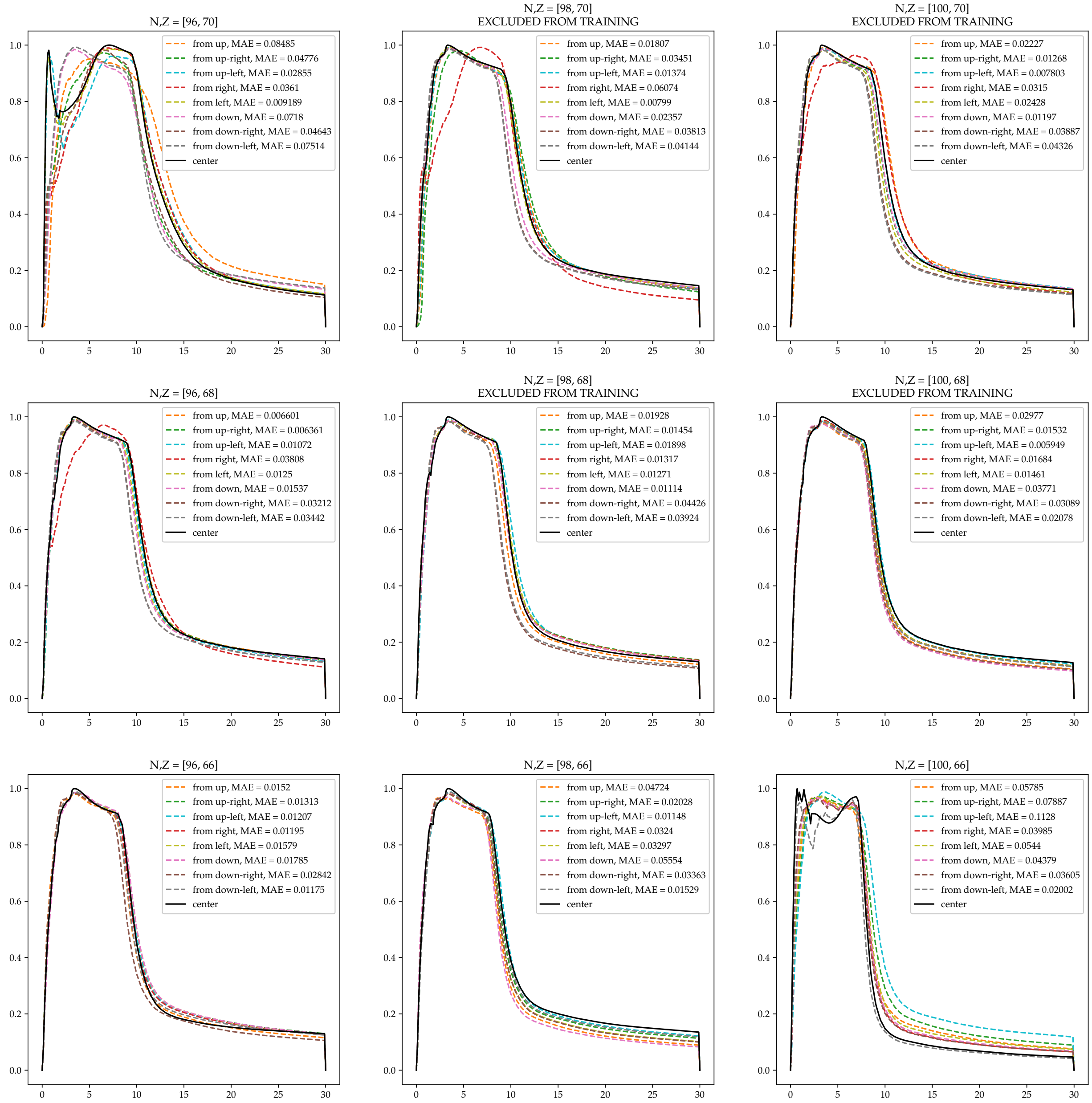


Figure 6.14. TENDL compared to local predictions of the VAE+GAN model. Cross sections which have been excluded from the training set are labelled. In each tile, the solid black line is the TENDL cross section evaluation and each colored dashed line is the prediction from a direct neighbor. The directions color-coding and mean absolute error are given in each legend.

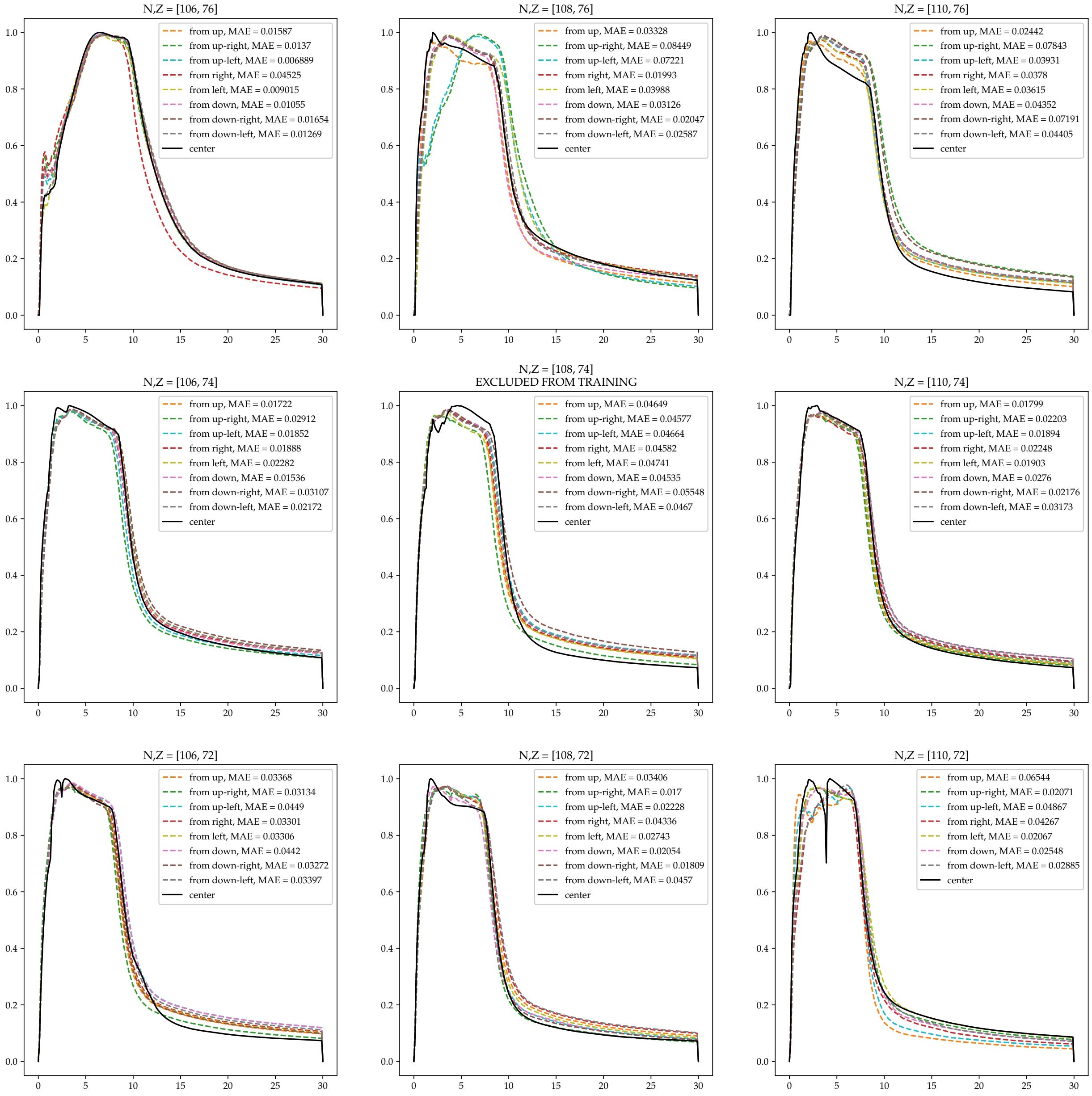


Figure 6.15. TENDL compared to local predictions of the VAE+GAN model. Cross sections which have been excluded from the training set are labelled. In each tile, the solid black line is the TENDL cross section evaluation and each colored dashed line is the prediction from a direct neighbor. The directions color-coding and mean absolute error are given in each legend. Features in this region are pretty simple, so we can reasonably expect the model to make accurate predictions.

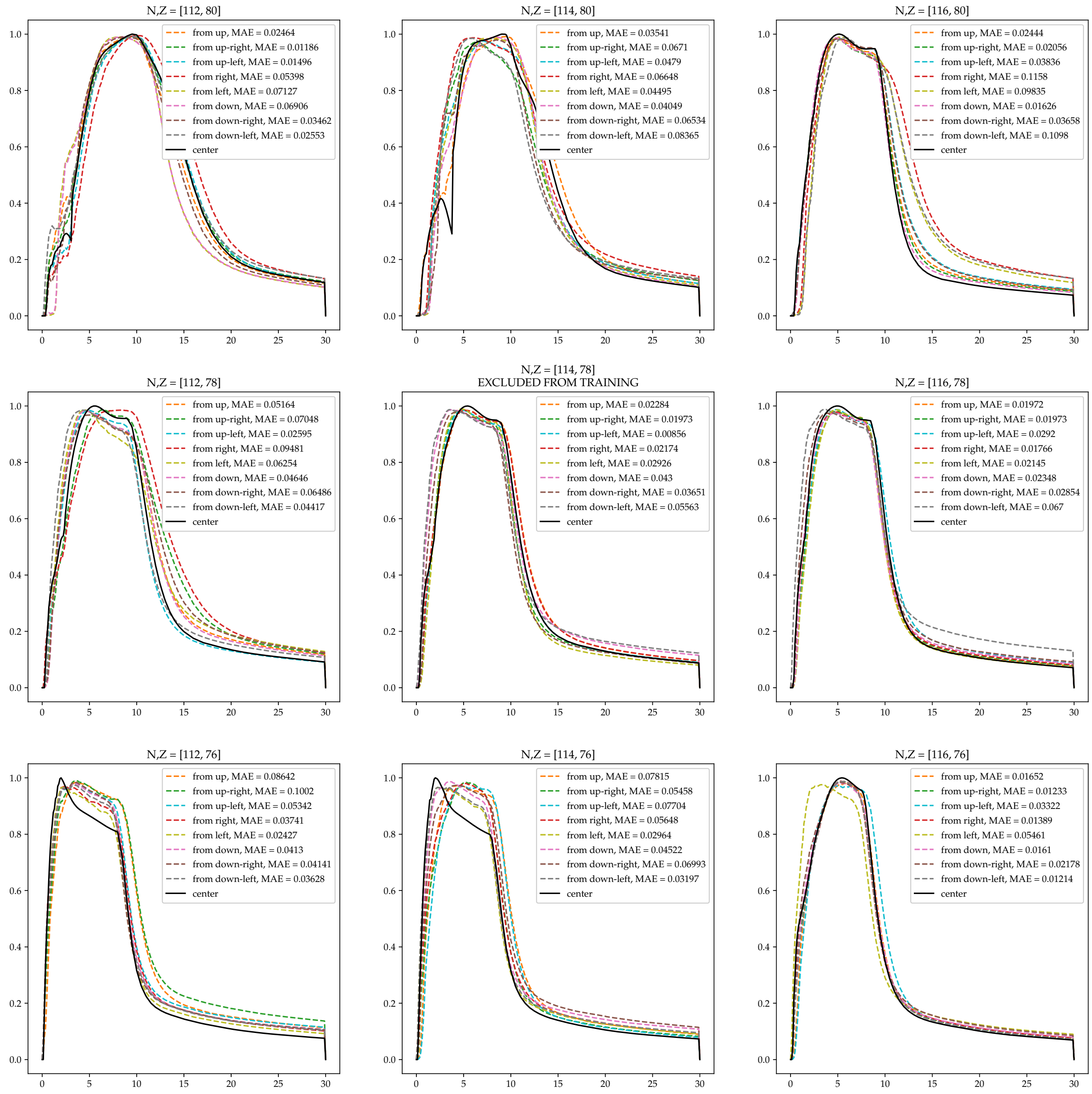


Figure 6.16. TENDL compared to local predictions of the VAE+GAN model. Cross sections which have been excluded from the training set are labelled. In each tile, the solid black line is the TENDL cross section evaluation and each colored dashed line is the prediction from a direct neighbor. The directions color-coding and mean absolute error are given in each legend. Features in this region are pretty simple, so we can reasonably expect the model to make accurate predictions.

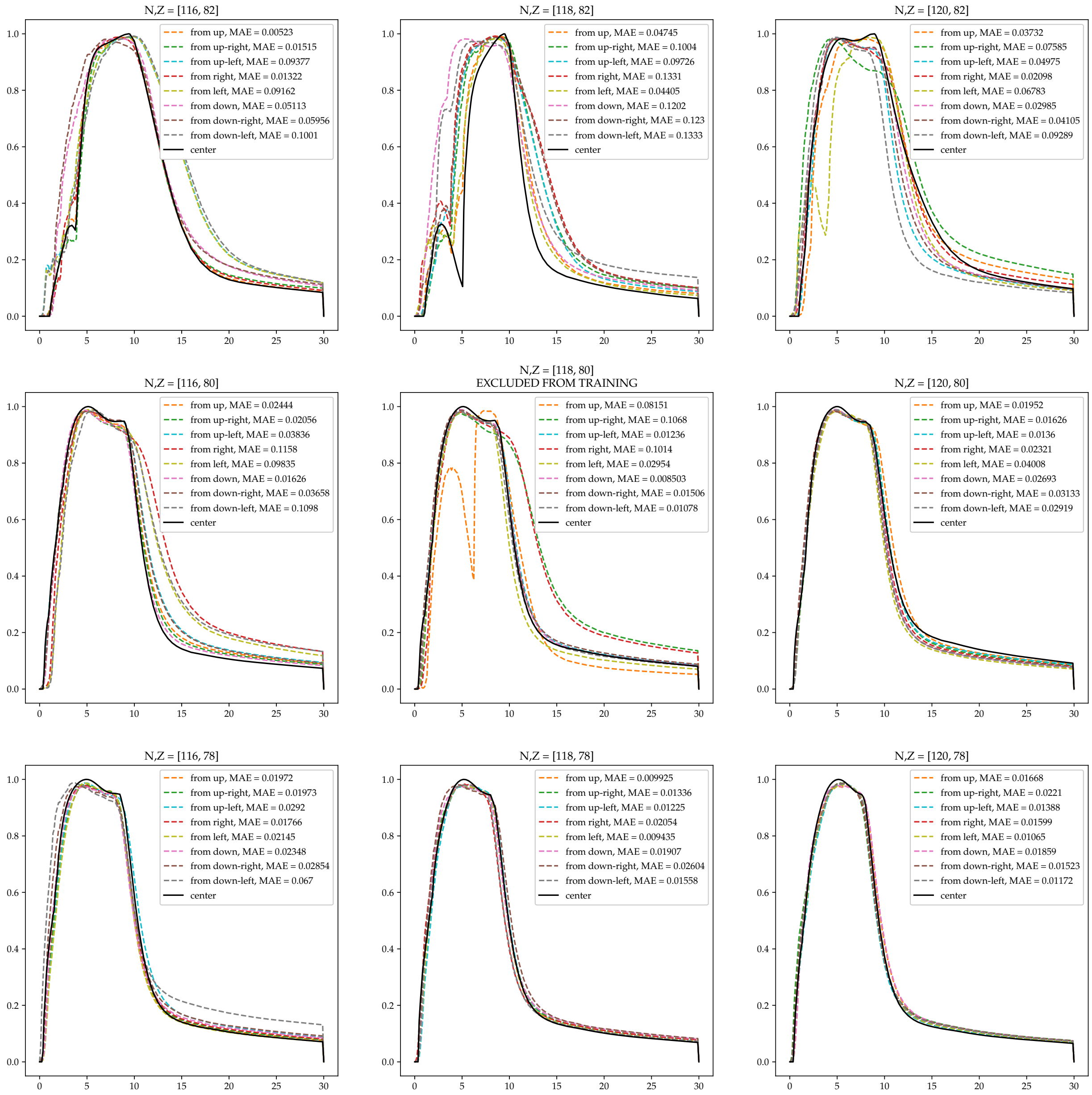


Figure 6.17. TENDL compared to local predictions of the VAE+GAN model. Cross sections which have been excluded from the training set are labelled. In each tile, the solid black line is the TENDL cross section evaluation and each colored dashed line is the prediction from a direct neighbor. The directions color-coding and mean absolute error are given in each legend.

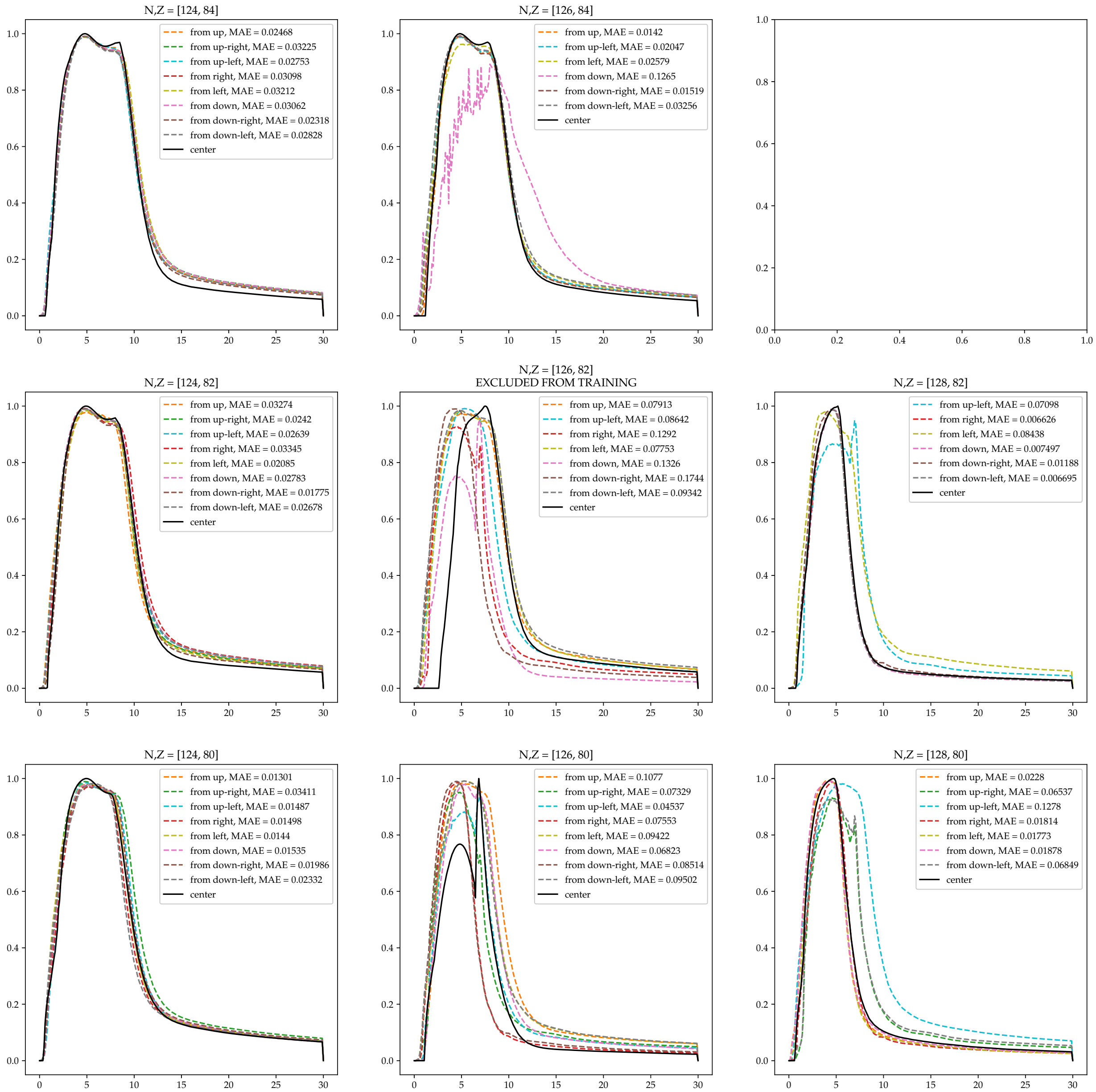


Figure 6.18. TENDL compared to local predictions of the VAE+GAN model. Cross sections which have been excluded from the training set are labelled. In each tile, the solid black line is the TENDL cross section evaluation and each colored dashed line is the prediction from a direct neighbor. The directions color-coding and mean absolute error are given in each legend.

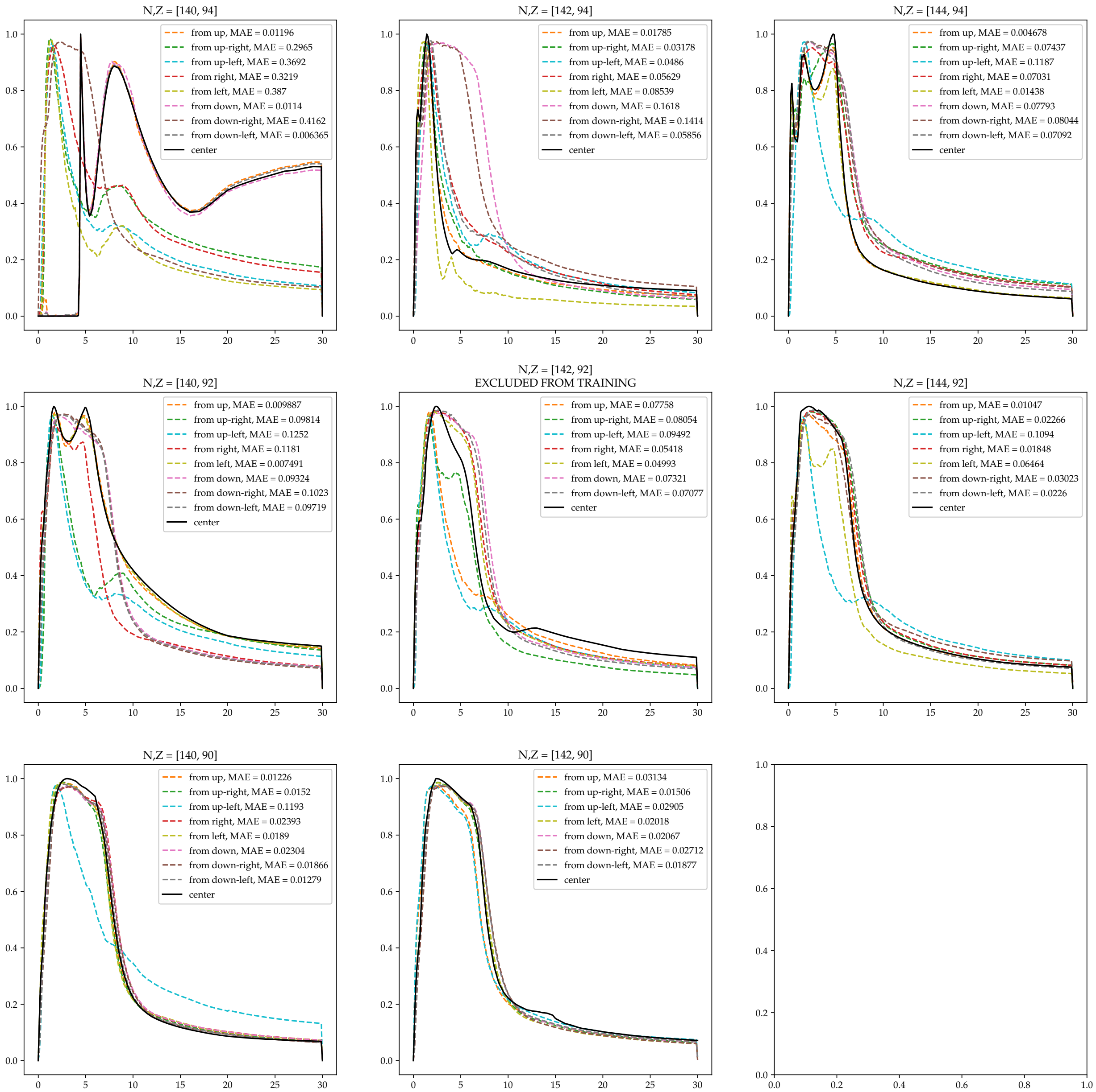


Figure 6.19. TENDL compared to local predictions of the VAE+GAN model. Cross sections which have been excluded from the training set are labelled. In each tile, the solid black line is the TENDL cross section evaluation and each colored dashed line is the prediction from a direct neighbor. The directions color-coding and mean absolute error are given in each legend. In the upper-left we can see Plutonium-234, the shape of which is quite different from those around it.

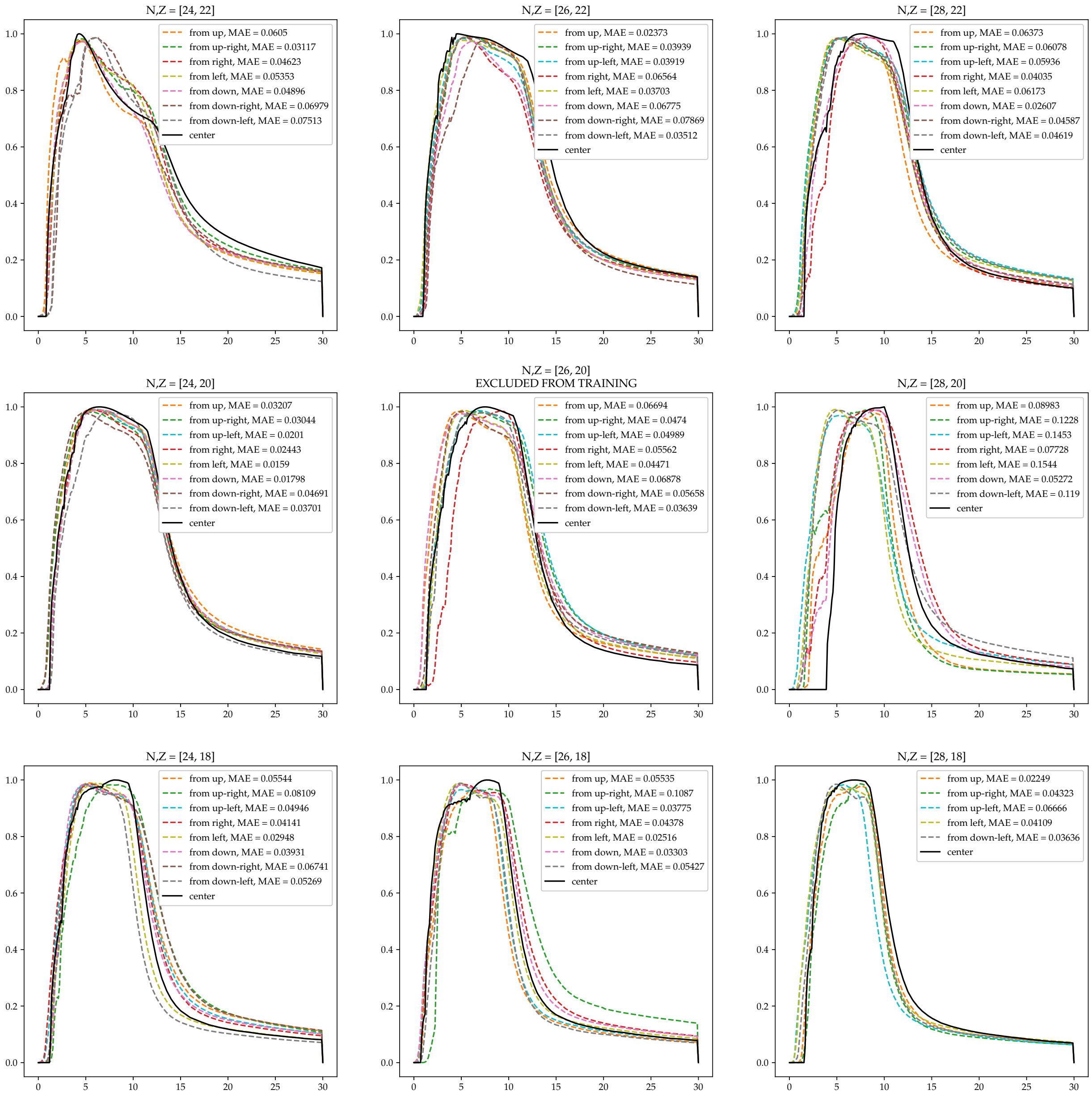


Figure 6.20. TENDL compared to local predictions of the VAE+GAN model. Cross sections which have been excluded from the training set are labelled. In each tile, the solid black line is the TENDL cross section evaluation and each colored dashed line is the prediction from a direct neighbor. The directions color-coding and mean absolute error are given in each legend.

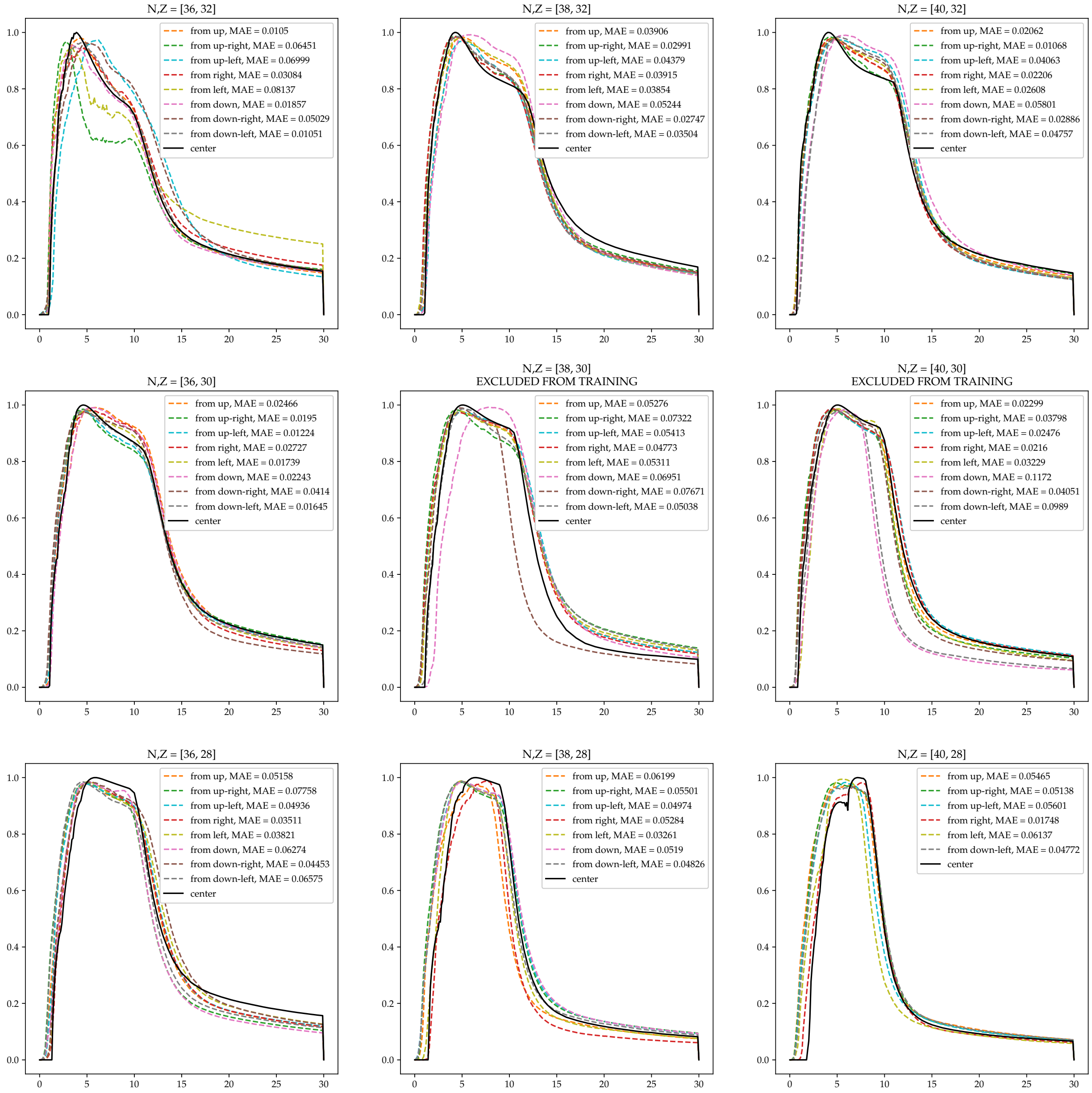
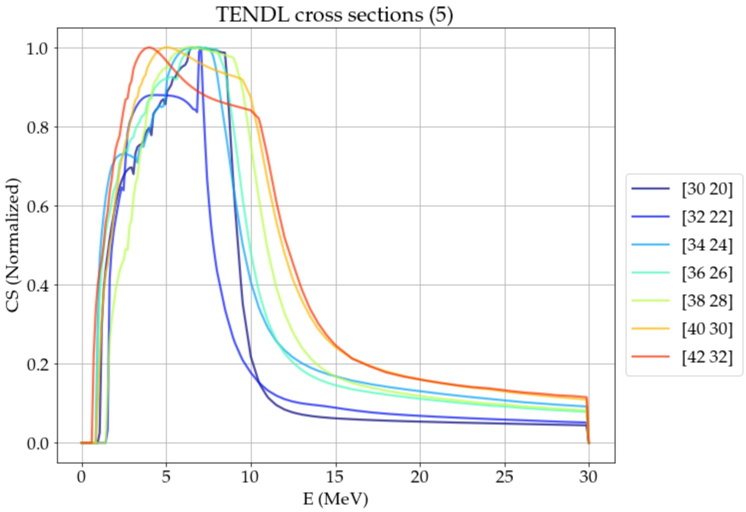
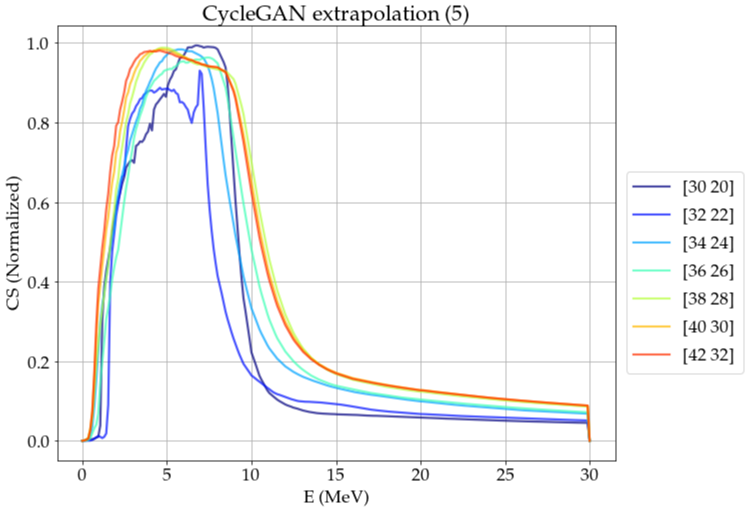


Figure 6.21. TENDL compared to local predictions of the VAE+GAN model. Cross sections which have been excluded from the training set are labelled. In each tile, the solid black line is the TENDL cross section evaluation and each colored dashed line is the prediction from a direct neighbor. The directions color-coding and mean absolute error are given in each legend.

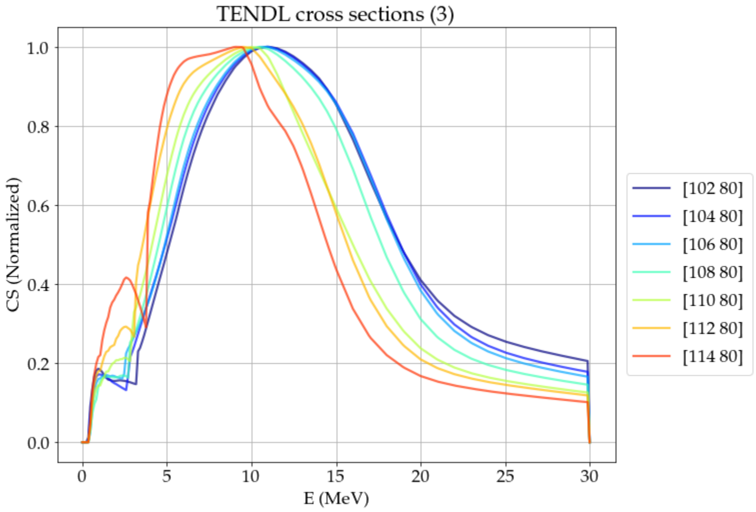


(a) TENDL cross sections, normalized.

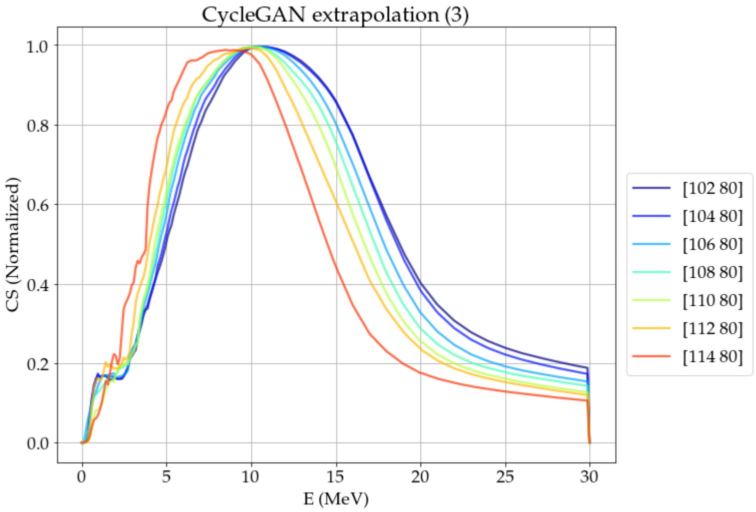


(b) GAN predictions, chained.

Figure 6.22. Cross sections in the $(N, Z) \to (N{+}2, Z{+}2)$ direction ($\nearrow$), beginning at $(N, Z) = (30, 20)$ for 6 steps. The only cross section data supplied to the GAN is the first cross section, shown in the darkest blue. In this case, the GAN has reproduced the larger trends well and only smaller features are lost by the end of the interpolation (red curve).

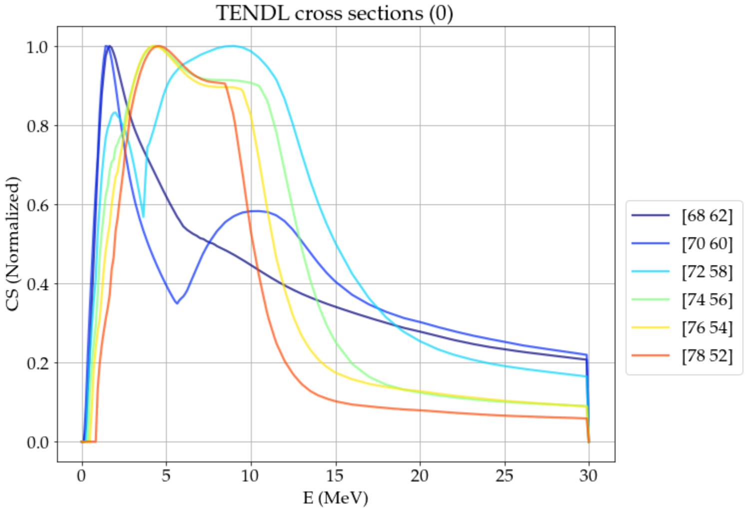


(a) TENDL cross sections, normalized.

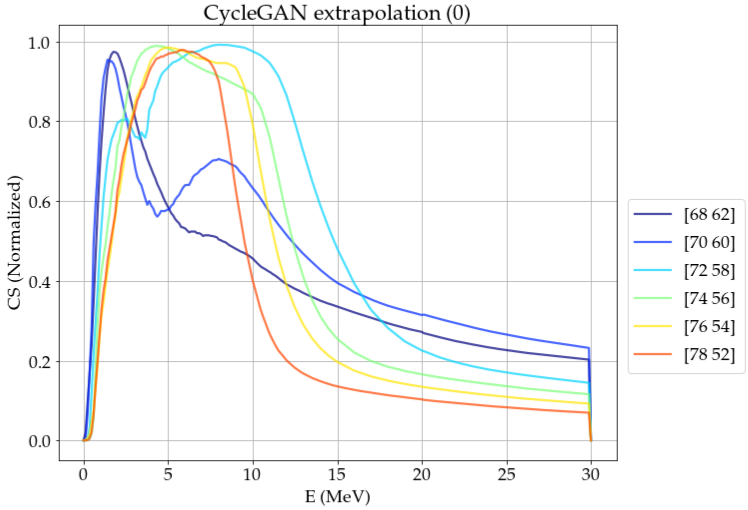


(b) GAN predictions, chained.

Figure 6.23. Cross sections in the $(N, Z) \to (N+2, Z)$ direction (→), beginning at $(N, Z) = (102, 80)$ for 6 steps. The only cross section data supplied to the GAN is the first cross section, shown in the darkest blue. We can see a peak emerge around 2.5 MeV in the TENDL data at $N = 112, 114$, which the GAN does not reproduce. The broader changes, however, are reproduced nicely.

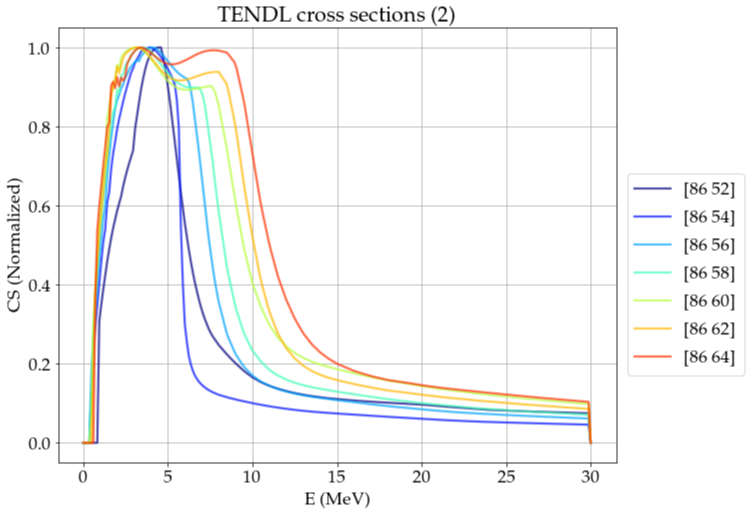


(a) TENDL cross sections, normalized.

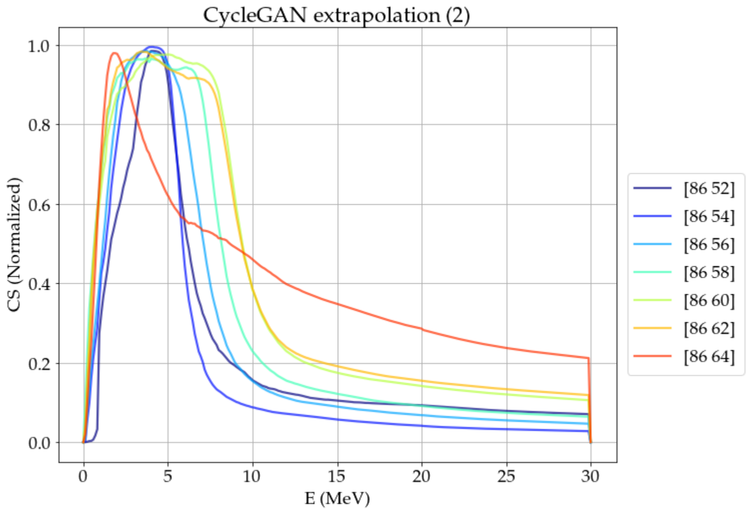


(b) GAN predictions, chained.

Figure 6.24. Cross sections in the $(N, Z) \rightarrow (N{+}2, Z{-}2)$ direction (↘), beginning at $(N, Z) = (68, 62)$ for 5 steps. The only cross section data supplied to the GAN is the first cross section, shown in the darkest blue. This sequence, while including drastic changes, is reproduced nicely by the model. As is typical however, some small details in the predictions are lost 6 steps away (red curve).



(a) **TENDL cross sections, normalized.**



(b) **GAN predictions, chained.**

Figure 6.25. **Cross sections in the $(N, Z) \to (N, Z+2)$ direction (↑), beginning at $(N, Z) = (86, 52)$ for 6 steps. The only cross section data supplied to the GAN is the first cross section, shown in the darkest blue. I include this example because we see the GAN reproduce the trends well until $(86, 64)$ (red curve), which is localized at lower energy than the TENDL data and has a smaller width and a longer tail. Note the prediction just before that in the chain, $(86, 62)$ (yellow), appears quite similar to the TENDL data. This makes sense, since the general shape of $(86, 62)$ is very common among the data and thus the model must rely on smaller changes to decide what comes next.**

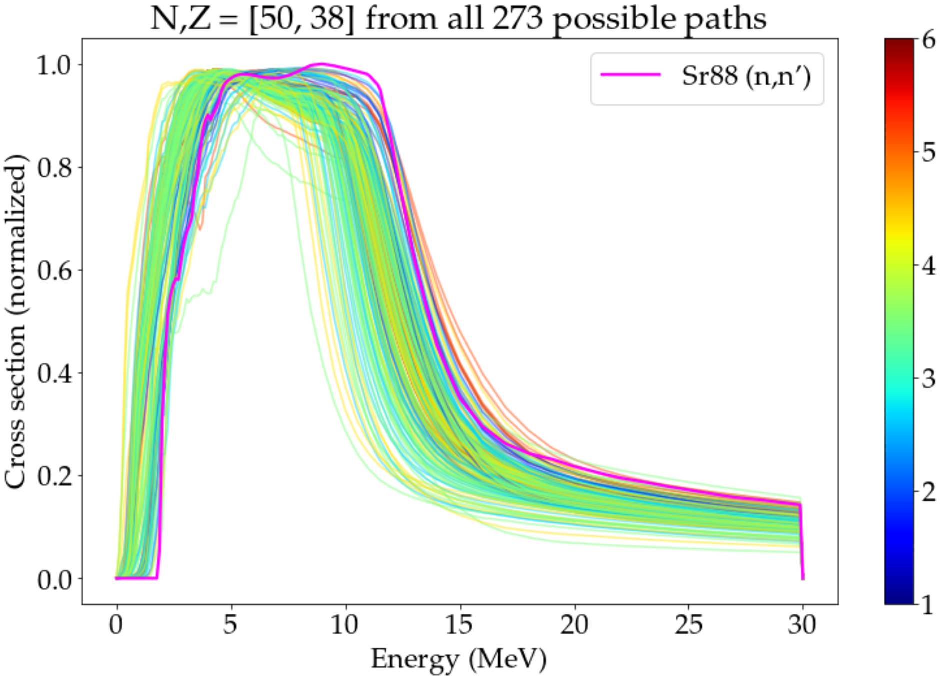


Figure 6.26. Ensemble prediction of the (normalized) cross section of Strontium-88, using 273 paths within a bounded local region of the chart of nuclides. The color of predictions correspond to the number of model evaluations used, as shown in the colorbar. The pink line shows the TENDL cross section evaluation.

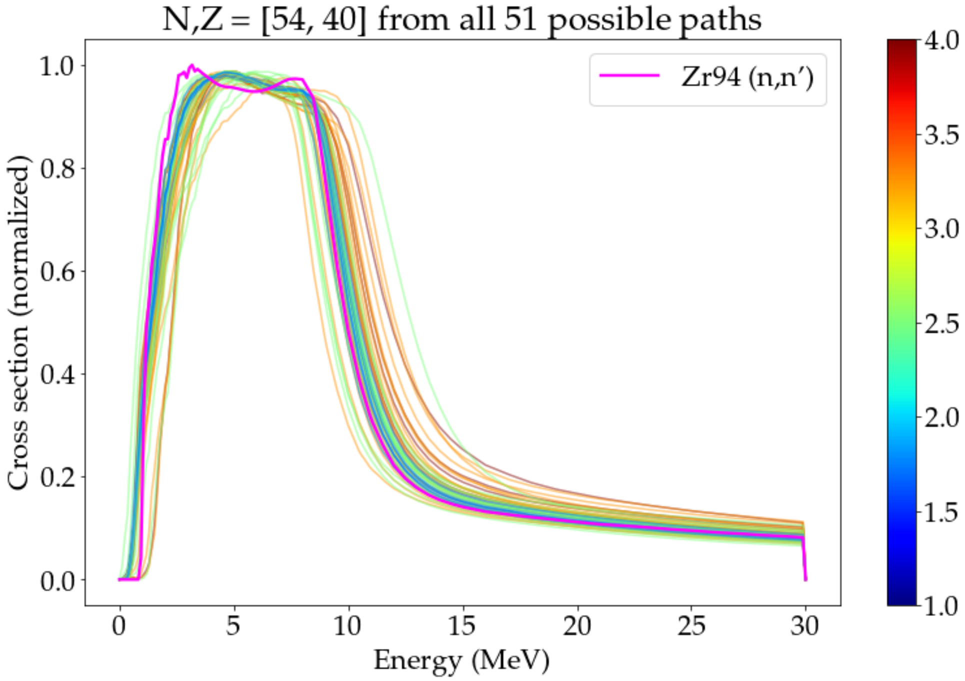


Figure 6.27. Ensemble prediction of the (normalized) cross section of Zirconium-94, using 51 paths within a bounded local region of the chart of nuclides. The color of predictions correspond to the number of model evaluations used, as shown in the colorbar. The pink line shows the TENDL cross section evaluation.

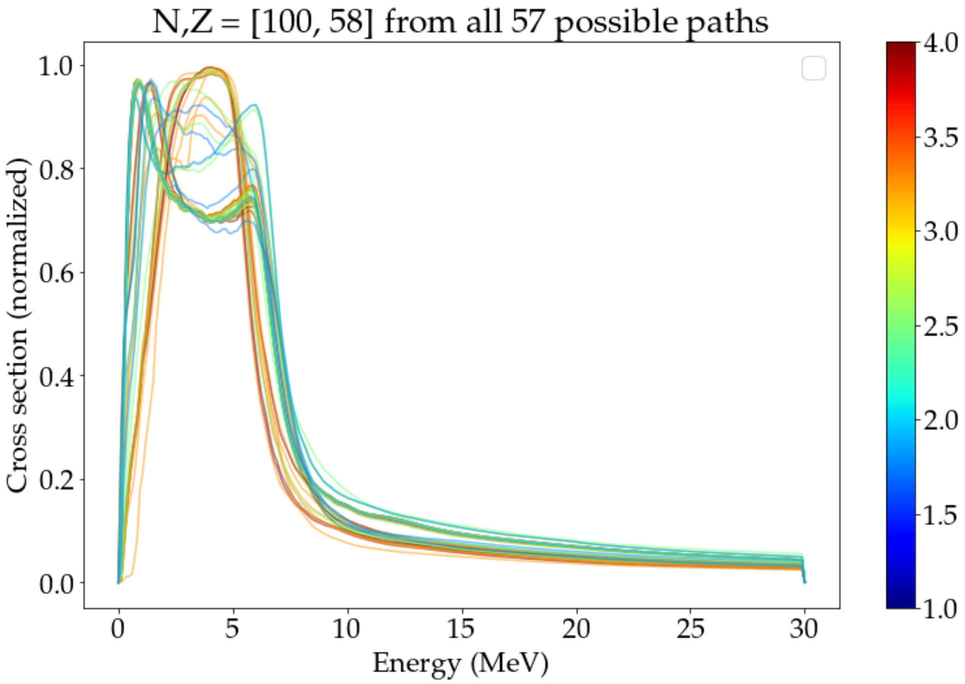


Figure 6.28. Ensemble prediction of the (normalized) cross section of Cerium-158, using 57 paths within a bounded local region of the chart of nuclides. The color of predictions correspond to the number of model evaluations used, as shown in the colorbar.

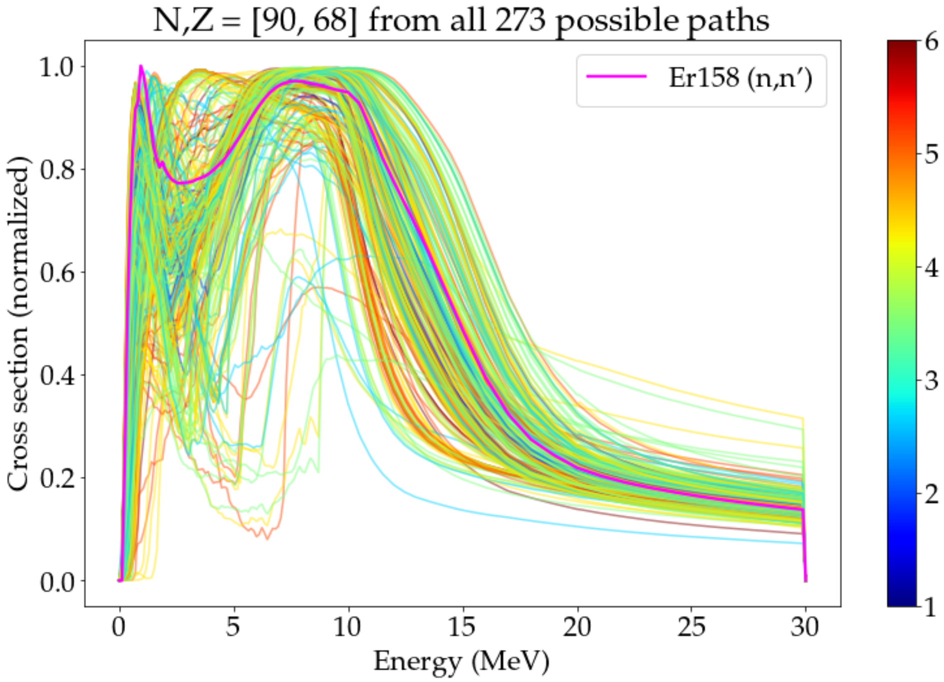


Figure 6.29. **Ensemble prediction of the (normalized) cross section of Erbium-158, using 273 paths within a bounded local region of the chart of nuclides. The color of predictions correspond to the number of model evaluations used, as shown in the colorbar. The pink line shows the TENDL cross section evaluation.**

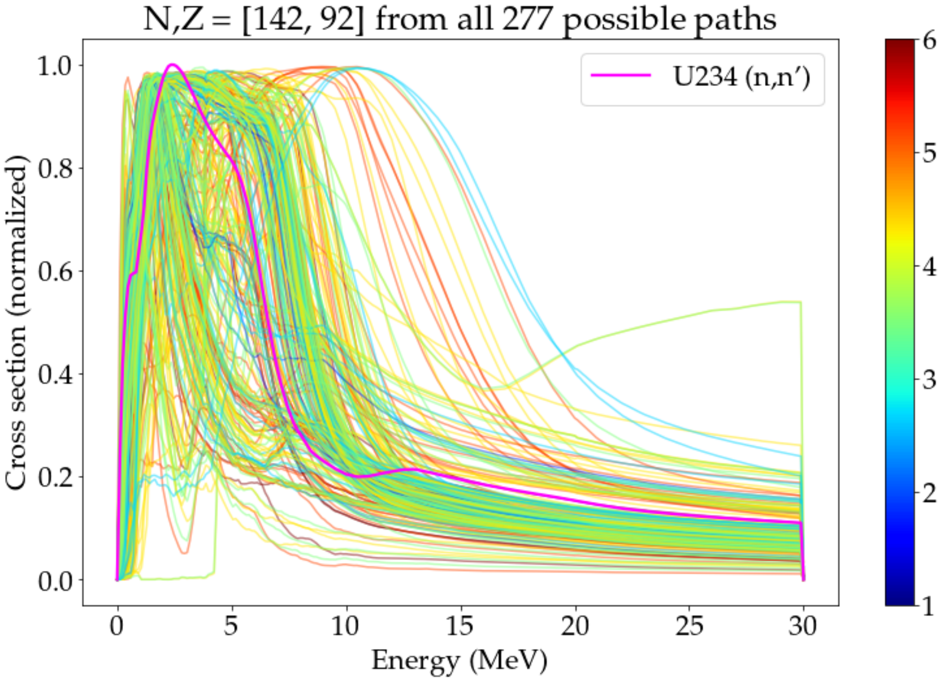


Figure 6.30. Ensemble prediction of the (normalized) cross section of Uranium-234, using 277 paths within a bounded local region of the chart of nuclides. The color of predictions correspond to the number of model evaluations used, as shown in the colorbar. The pink line shows the TENDL cross section evaluation.

CHAPTER 7

CONCLUSIONS

In this dissertation I have described two projects I completed during my doctoral research.

The first project was about uncertainty quantification of configuration interaction shell model calculations. I have shown how to assign a probability distribution to parameters of an empirical interaction model, and then how to propagate that distribution through the calculation of nuclear transition strengths. This provides a posterior distribution for not only the transition strengths, which are compared to experimental measurements, but also for coupling constants which show up in a number of theoretical calculations. This procedure of assigning uncertainties to elements of nuclear theory is important for three main reasons. First, it allows us to assign more accurate confidence intervals to physical quantities, accounting for experimental and theoretical uncertainties. Second, it aids further development of the theory by giving correlations between informative variables, for example showing in detail which parts of the empirical Hamiltonian are most important for beta decay, sum rules, and so on. Third, assigning theoretical uncertainties to physical quantities can be important for efficient experimental design. Experiments are expensive, and the price tag is determined in part by the range of energies necessary to probe the physics of interest. Recommendations from theory as to which energy regimes should be prioritized, versus others which may likely be unfruitful, can lead to saving precious resources of money and work-hours. This is not a new development in nuclear physics of course, experimental design is always guided by theory, but more precise and robust UQ accelerates the physics community toward discovery.

The future of my UQ research holds many possibilities. One of the most powerful recent developments is an emulator method specific to eigenvalue problems, known as *eigenvector continuation* (EC). EC allows one to interpolate eigenvectors of the Hamiltonian by some parameters with extremely good accuracy, as long as the eigenvalue problem can be solved within the range of interest. Rather than computing solutions via the Lanczos algorithm to generate statistics (i.e. 1000-10,000 times), EC allows one to perform the full calculation fewer than 100 times, then use the emulator to generate statistics instead, decreasing the cost of UQ immensely. This was not pursued in my doctoral research because the practical code implementation of EC for a sophisticated CI code like BIGSTICK is far from trivial.

Another goal we can pursue beyond my UQ research is fitting an empirical Hamiltonian for shell model calculations using my results in the *sd*-shell. At the time of writing, the "gold-standard" interaction, as discussed in this dissertation, is USDB, which is fit only using excitation energies. Given the hypothesis that the general accuracy of transition and moment calculations could be improved by including transitions and moments in the fitted dataset, we could further optimize USDB via an iterative method and publish it for others to use, along with estimates of sensitivities and correlations.

The second project in my dissertation was about studying systematic trends in nuclear cross section libraries using deep learning models. This work began during a summer internship at Lawrence Livermore National Lab with Kyle Wendt, and we have continued development for the last two years. In that time, the model has taken many forms and I experienced many challenges along the way. Ultimately, I have designed a system which combines the encoding power (technically *manifold learning*) of a variational auto-encoder (VAE) with the *distribution learning* capability of a generative adversarial network. Training the GAN is nontrivial: adversarial networks are notoriously difficult to train and that is an open area of research at the time of writing.

However, using the VAE to encode data before training allowed us to decouple that task from the generative network, and ultimately the model provides reasonable predictions. This can be used to predict extensions to the evaluated libraries for which experiments may not be available, or to predict cross sections we do have data for in order to compare them with the model's expectations.

It is likely that the future incarnations of models for this project will use fundamentally different deep learning models. The VAE+GAN model has been a worthy endeavor but there are a number of objectives which the current model falls short on. The GAN transforms for adding/subtracting different particles are completely independent, and so correlations between directions cannot be leveraged. Correlations between reaction channels ((n,n),(n,n'),(n,2n),etc.) also cannot be leveraged. Ideally both encoding and transforms would be learned by a single deep network; as discussed, we pursued this in the beginning but ended up factoring the problem into two parts to make development easier. This factorization forces us to use an unnecessarily abstract representation for data, which leads to compounding of errors in predictions. One promising avenue is that of graphical neural networks (GNNs), which as the name suggests work on graph data; since the chart of nuclides and evaluation libraries already have a particular graph structure, using a GNN may allow for a very natural encoding of data and leveraging of correlations. I will be continuing work on this project after this dissertation is published with Kyle Wendt and other collaborators.

BIBLIOGRAPHY

[1] *Kolmogorov–Smirnov Test*, Springer New York, New York, NY, 2008, pp. 283–287.

[2] *1.3.3.24: Quantile-Quantile Plot*, NIST, 2013. `http://www.itl.nist.gov/div898/handbook/`.

[3] *Alphago*, 2017.

[4] R. Acciarri, S. Bansal, A. Friedland, Z. Djurcic, L. Rakotondravohitra, T. Xin, E. Mazzucato, C. Densham, E. Calvo, S. Li, et al., *Long-baseline neutrino facility (lbnf) and deep underground neutrino experiment (dune): Volume 1: The lbnf and dune projects*, technical report, 2016.

[5] P. Agnes, T. Alexander, A. Alton, K. Arisaka, H. Back, B. Baldin, K. Biery, G. Bonfini, M. Bossa, A. Brigatti, et al., *First results from the darkside-50 dark matter experiment at laboratori nazionali del gran sasso*, Physics Letters B, 743 (2015), pp. 456–466.

[6] S. Aldor-Noiman, L. D. Brown, A. Buja, W. Rolke, and R. A. Stine, *The Power to See: A New Graphical Test of Normality*, The American Statistician, 67 (2013), pp. 249–260.

[7] N. Anand, A. L. Fitzpatrick, and W. C. Haxton, *Weakly interacting massive particle-nucleus elastic scattering response*, Phys. Rev. C, 89 (2014), p. 065501.

[8] A. B. Balantekin, J. Carlson, D. J. Dean, G. M. Fuller, R. J. Furnstahl, M. Hjorth-Jensen, R. V. F. Janssens, B.-A. Li, W. Nazarewicz, F. M. Nunes, W. E. Ormand, S. Reddy, and B. M. Sherrill, *Nuclear theory and science of the facility for rare isotope beams*, Mod. Phys. Lett. A, 29 (2014), p. 1430010.

[9] B. R. Barrett, P. Navrátil, and J. P. Vary, *Ab initio no core shell model*, Progress in Particle and Nuclear Physics, 69 (2013), pp. 131–181.

[10] E. BAUM et al., *Chart of the nuclides: Nuclides and isotopes*, 2009.

[11] G. Bertone and D. Hooper, *History of dark matter*, Rev. Mod. Phys., 90 (2018), p. 045002.

[12] S. Binder, J. Langhammer, A. Calci, and R. Roth, Ab Initio *path to heavy nuclei*, Physics Letters B, 736 (2014), pp. 119–123.

[13] J. Blomqvist and A. Molinari, *Collective 0 vibrations in even spherical nuclei with tensor forces*, Nuclear Physics A, 106 (1968), pp. 545–569.

[14] S. K. Bogner, R. J. Furnstahl, and R. J. Perry, *Similarity renormalization group for nucleon-nucleon interactions*, Phys. Rev. C, 75 (2007), p. 061001.

[15] A. Botchkarev, *A new typology design of performance metrics to measure errors in machine learning regression algorithms*, 14 (2019), pp. 045–076.

[16] S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*, Chapman & Hall/CRC, 2011.

[17] B. Brown and B. Wildenthal, *Experimental and theoretical gamow-teller beta-decay observables for the sd-shell nuclei*, Atomic Data and Nuclear Data Tables, 33 (1985), pp. 347 – 404.

[18] B. A. Brown and W. A. Richter, *New "usd" hamiltonians for the sd shell*, Phys. Rev. C, 74 (2006), p. 034315.

[19] B. A. Brown and W. A. Richter, *New "usd" hamiltonians for the sd shell*, Phys. Rev. C, 74 (2006), p. 034315.

[20] B. A. Brown and B. Wildenthal, *Status of the nuclear shell model*, Annual Review of Nuclear and Particle Science, 38 (1988), pp. 29–66.

[21] B. A. Brown and B. H. Wildenthal, *Status of the nuclear shell model*, Annual Review of Nuclear and Particle Science, 38 (1988), pp. 29–66.

[22] P. Brussard and P. Glaudemans, *Shell-model applications in nuclear spectroscopy*, North-Holland Publishing Company, Amsterdam, 1977.

[23] B. Carlsson, A. Ekström, C. Forssén, D. F. Strömberg, G. Jansen, O. Lilja, M. Lindby, B. Mattsson, and K. Wendt, *Uncertainty analysis and order-by-order optimization of chiral nuclear interactions*, Physical Review X, 6 (2016), p. 011019.

[24] E. Caurier, G. Martinez-Pinedo, F. Nowacki, A. Poves, and A. P. Zuker, *The shell model as a unified view of nuclear structure*, Reviews of Modern Physics, 77 (2005), pp. 427–488.

[25] M. B. Chadwick, M. Herman, P. Obložinskỳ, M. E. Dunn, Y. Danon, A. Kahler, D. L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla, et al., *Endf/b-vii. 1 nuclear data for science and technology: cross sections, covariances, fission product yields and decay data*, Nuclear data sheets, 112 (2011), pp. 2887–2996.

[26] G. E. Dahl, D. Yu, L. Deng, and A. Acero, *Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition*, IEEE Transactions on Audio, Speech, and Language Processing, 20 (2012), p. 30–42.

[27] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. a. Ranzato, A. Senior, P. Tucker, K. Yang, Q. Le, and A. Ng, *Large scale distributed deep networks*, in Advances in Neural Information Processing Systems, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, eds., vol. 25, Curran Associates, Inc., 2012.

[28] J. Dobaczewski, W. Nazarewicz, and P. Reinhard, *Error estimates of theoretical models: a guide*, Journal of Physics G: Nuclear and Particle Physics, 41 (2014), p. 074001.

[29] M. J. Dolinski, A. W. Poon, and W. Rodejohann, *Neutrinoless double-beta decay: Status and prospects*, Annual Review of Nuclear and Particle Science, 69 (2019).

[30] C. Drischler, J. Melendez, R. Furnstahl, and D. Phillips, *Quantifying uncertainties and correlations in the nuclear-matter equation of state*, Physical Review C, 102 (2020), p. 054315.

[31] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, *Hybrid monte carlo*, Physics Letters B, 195 (1987), pp. 216–222.

[32] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, *Activation functions in deep learning: A comprehensive survey and benchmark*, Neurocomputing, 503 (2022), pp. 92–108.

[33] A. R. Edmonds, *Angular momentum in quantum mechanics*, Princeton University Press, Princeton, 1996.

[34] A. Ekström, C. Forssén, C. Dimitrakakis, D. Dubhashi, H. T. Johansson, A. S. Muhammad, H. Salomonsson, and A. Schliep, *Bayesian optimization in* ab initio *nuclear physics*, arXiv:1902.00941 [nucl-th, stat], (2019).

[35] A. Ekström and G. Hagen, *Global sensitivity analysis of bulk properties of an atomic nucleus*, Phys. Rev. Lett., 123 (2019), p. 252501.

[36] J. Engel and J. Menéndez, *Status and future of nuclear matrix elements for neutrinoless double-beta decay: a review*, Reports on Progress in Physics, 80 (2017), p. 046301.

[37] J. Erler and P.-G. Reinhard, *Error estimates for the Skyrme–Hartree–Fock model*, Journal of Physics G: Nuclear and Particle Physics, 42 (2015), p. 034026.

[38] R. P. Feynman, *Forces in molecules*, Phys. Rev., 56 (1939), pp. 340–343.

[39] D. Foreman-Mackey, *corner.py: Scatterplot matrices in python*, The Journal of Open Source Software, 1 (2016), p. 24.

[40] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, *emcee: The MCMC Hammer*, 125 (2013), p. 306.

[41] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, *emcee: the mcmc hammer*, Publications of the Astronomical Society of the Pacific, 125 (2013), p. 306.

[42] J. M. R. Fox, C. W. Johnson, and R. N. Perez, *Uncertainty quantification of an empirical shell-model interaction using principal component analysis*, Physical Review C, 101 (2020).

[43] J. M. R. Fox, C. W. Johnson, and R. N. Perez, *Uncertainty quantification of transition operators in the empirical shell model*, 2022.

[44] D. Frame, R. He, I. Ipsen, D. Lee, D. Lee, and E. Rrapaj, *Eigenvector continuation with subspace learning*, Phys. Rev. Lett., 121 (2018), p. 032501.

[45] D. Frame, R. He, I. Ipsen, D. Lee, D. Lee, and E. Rrapaj, *Eigenvector continuation with subspace learning*, Physical review letters, 121 (2018), p. 032501.

[46] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, *Cyclical annealing schedule: A simple approach to mitigating kl vanishing*, 2019.

[47] K. Fukushima, *Cognitron: A self-organizing multilayered neural network*, Biological cybernetics, 20 (1975), pp. 121–136.

[48] K. Fukushima and S. Miyake, *Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition*, in Competition and cooperation in neural nets, Springer, 1982, pp. 267–285.

[49] R. Furnstahl, D. Phillips, and S. Wesolowski, *A recipe for eft uncertainty quantification in nuclear physics*, Journal of Physics G: Nuclear and Particle Physics, 42 (2015), p. 034028.

[50] R. J. Furnstahl, G. Hagen, T. Papenbrock, and K. A. Wendt, *Infrared extrapolations for atomic nuclei*, Journal of Physics G: Nuclear and Particle Physics, 42 (2015), p. 034032.

[51] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, Chapman and Hall/CRC, 2nd ed. ed., 2004.

[52] A. Gelman, D. Lee, and J. Guo, *Stan: A probabilistic programming language*

for bayesian inference and optimization, Journal of Educational and Behavioral Statistics, 40 (2015), pp. 530–543.

[53] R. Ghanem, D. Higdon, and e. a. Owhadi, Houman, *Handbook of uncertainty quantification*, vol. 6, Springer, New York, 2017.

[54] A. Gonzalez-Garcia, J. van de Weijer, and Y. Bengio, *Image-to-image translation for cross-domain disentanglement*, arXiv e-prints, (2018), p. arXiv:1805.09730.

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016. `http://www.deeplearningbook.org`.

[56] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative Adversarial Networks*, arXiv e-prints, (2014), p. arXiv:1406.2661.

[57] P. Gysbers, G. Hagen, J. Holt, G. R. Jansen, T. D. Morris, P. Navrátil, T. Papenbrock, S. Quaglioni, A. Schwenk, S. Stroberg, et al., *Discrepancy between experimental and theoretical β-decay rates resolved from first principles*, Nature Physics, 15 (2019), pp. 428–431.

[58] G. Hagen, T. Papenbrock, D. J. Dean, and M. Hjorth-Jensen, *Ab initio coupled-cluster approach to nuclear structure with modern nucleon-nucleon interactions*, Physical Review C, 82 (2010), p. 034330.

[59] H. Hellman, *Einführung in die quantenchemie*, Franz Deuticke, Leipzig, (1937), p. 285.

[60] H. Hergert, S. Bogner, T. Morris, A. Schwenk, and K. Tsukiyama, *The in-medium similarity renormalization group: A novel ab initio method for nuclei*, Physics Reports, 621 (2016), pp. 165–222.

[61] M. Hjorth-Jensen, T. T. Kuo, and E. Osnes, *Realistic effective interactions for nuclear systems*, Physics Reports, 261 (1995), p. 125–270.

[62] M. D. Hoffman, A. Gelman, et al., *The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo.*, J. Mach. Learn. Res., 15 (2014), pp. 1593–1623.

[63] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, *Multimodal Unsupervised Image-to-Image Translation*, arXiv e-prints, (2018), p. arXiv:1804.04732.

[64] Y. Jaganathen, R. M. I. Betan, N. Michel, W. Nazarewicz, and M. Płoszajczak, *Quantified gamow shell model interaction for psd-shell nuclei*, Phys. Rev. C, 96 (2017), p. 054316.

[65] C. W. Johnson, *Systematics of strength function sum rules*, Physics Letters B, 750 (2015), pp. 72–75.

[66] C. W. Johnson and P. G. Krastev, *Sensitivity analysis of random two-body interactions*, Phys. Rev. C, 81 (2010), p. 054303.

[67] C. W. Johnson, W. E. Ormand, and P. G. Krastev, *Factorization in large-scale many-body calculations*, Computer Physics Communications, 184 (2013), pp. 2761–2774.

[68] C. W. Johnson, W. E. Ormand, K. S. McElvain, and H. Shan, *BIGSTICK: A flexible configuration-interaction shell-model code*, (2018).

[69] C. W. Johnson, W. E. Ormand, K. S. McElvain, and H. Shan, *Bigstick: A flexible configuration-interaction shell-model code*, arXiv preprint arXiv:1801.08432, (2018).

[70] T. Karras, S. Laine, and T. Aila, *A Style-Based Generator Architecture for Generative Adversarial Networks*, arXiv e-prints, (2018), p. arXiv:1812.04948.

[71] M. C. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63 (2001), pp. 425–464.

[72] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv e-prints, (2014), p. arXiv:1412.6980.

[73] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, (2014). arXiv:1312.6114 [cs, stat].

[74] R. D. Klauber, *Student Friendly Quantum Field Theory:: Basic Principles and Quantum Electrodynamics*, Sandtrove Press, Fairfield, Iowa, 2013.

[75] S. König, A. Ekström, K. Hebeler, D. Lee, and A. Schwenk, *Eigenvector continuation as an efficient and accurate emulator for uncertainty quantification*, arXiv preprint arXiv:1909.08446, (2019).

[76] S. König, A. Ekström, K. Hebeler, D. Lee, and A. Schwenk, *Eigenvector continuation as an efficient and accurate emulator for uncertainty quantification*, Physics Letters B, 810 (2020), p. 135814.

[77] A. Koning, D. Rochman, J.-C. Sublet, N. Dzysiuk, M. Fleming, and S. van der Marck, *Tendl: Complete nuclear data library for innovative nuclear science and technology*, Nuclear Data Sheets, 155 (2019), pp. 1–55. Special Issue on Nuclear Reaction Data.

[78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification*

with deep convolutional neural networks, in Advances in Neural Information Processing Systems, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, eds., vol. 25, Curran Associates, Inc., 2012.

[79] A. F. Lisetskiy, B. R. Barrett, M. K. G. Kruse, P. Navratil, I. Stetcu, and J. P. Vary, *Ab-initio shell model with a core*, Physical Review C, 78 (2008), p. 044302.

[80] A. E. Lovell, F. M. Nunes, M. Catacora-Rios, and G. B. King, *Recent advances in the quantification of uncertainties in reaction theory*, Journal of Physics G: Nuclear and Particle Physics, 48 (2020), p. 014001.

[81] A. E. Lovell, F. M. Nunes, J. Sarich, and S. M. Wild, *Uncertainty quantification for optical model parameters*, Phys. Rev. C, 95 (2017), p. 024611.

[82] L. Lyons, *Statistics for nuclear and particle physicists*, cambridge university press, 1989.

[83] A. L. Maas, A. Y. Hannun, A. Y. Ng, et al., *Rectifier nonlinearities improve neural network acoustic models*, in Proc. icml, vol. 30, Citeseer, 2013, p. 3.

[84] R. Machleidt, *The Nuclear force in the third millennium*, Nucl. Phys. A, 689 (2001), pp. 11–22.

[85] D. Martin, A. Arcones, W. Nazarewicz, and E. Olsen, *Impact of Nuclear Mass Uncertainties on the r-process*, Physical Review Letters, 116 (2016), p. 121101.

[86] J. D. McDonnell, N. Schunck, D. Higdon, J. Sarich, S. M. Wild, and W. Nazarewicz, *Uncertainty quantification for nuclear density functional theory and information content of new measurements*, Phys. Rev. Lett., 114 (2015), p. 122501.

[87] J. A. Melendez, R. J. Furnstahl, D. R. Phillips, M. T. Pratola, and S. Wesolowski, *Quantifying correlated truncation errors in effective field theory*, Phys. Rev. C, 100 (2019), p. 044001.

[88] L. Montanet, K. Gieselmann, R. M. Barnett, D. E. Groom, T. G. Trippe, C. G. Wohl, B. Armstrong, G. S. Wagman, H. Murayama, J. Stone, J. J. Hernandez, F. C. Porter, R. J. Morrison, A. Manohar, M. Aguilar-Benitez, C. Caso, P. Lantero, R. L. Crawford, M. Roos, N. A. Törnqvist, K. G. Hayes, G. Höhler, S. Kawabata, D. M. Manley, K. Olive, R. E. Shrock, S. Eidelman, R. H. Schindler, A. Gurtu, K. Hikasa, G. Conforto, R. L. Workman, and C. Grab, *Review of particle properties*, Phys. Rev. D, 50 (1994),

pp. 1173–1814.

[89] M. Mumpower, R. Surman, D. L. Fang, M. Beard, and A. Aprahamian, *The impact of uncertain nuclear masses near closed shells on the r-process abundance pattern*, Journal of Physics G: Nuclear and Particle Physics, 42 (2015), p. 034027.

[90] M. R. Mumpower, G. C. McLaughlin, R. Surman, and A. W. Steiner, *Reverse engineering nuclear properties from rare earth abundances in the r-process*, Journal of Physics G: Nuclear and Particle Physics, 44 (2017), p. 034003.

[91] M. R. Mumpower, R. Surman, G. C. McLaughlin, and A. Aprahamian, *The impact of individual nuclear properties on r-process nucleosynthesis*, Progress in Particle and Nuclear Physics, 86 (2016), pp. 86–126.

[92] R. Navarro Pérez, J. E. Amaro, and E. Ruiz Arriola, *Bootstrapping the statistical uncertainties of NN scattering data*, Physics Letters B, 738 (2014), pp. 155–159.

[93] R. Navarro Pérez, J. E. Amaro, and E. Ruiz Arriola, *Error analysis of nuclear forces and effective interactions*, Journal of Physics G Nuclear Physics, 42 (2015), p. 034013.

[94] R. Navarro Perez, E. Garrido, J. Amaro, and E. Ruiz Arriola, *Triton binding energy with realistic statistical uncertainties*, Phys. Rev. C, 90 (2014), p. 047001.

[95] R. Navarro Pérez, J. Amaro, E. Ruiz Arriola, P. Maris, and J. Vary, *Statistical error propagation in ab initio no-core full configuration calculations of light nuclei*, Phys. Rev. C, 92 (2015), p. 064003.

[96] R. Navarro Pérez and J. Lei, *Is the unusual near-threshold potential behavior in elastic scattering of weakly bound nuclei a precision error?*, Phys. Lett. B, 795 (2019), pp. 200–205.

[97] P. Navrátil, J. Vary, and B. Barrett, *Large-basis ab initio no-core shell model and its application to 12 c*, Physical Review C, 62 (2000), p. 054311.

[98] L. Neufcourt, Y. Cao, S. A. Giuliani, W. Nazarewicz, E. Olsen, and O. B. Tarasov, *Quantified limits of the nuclear landscape*, Physical Review C, 101 (2020), p. 044307.

[99] L. Neufcourt, Y. Cao, W. Nazarewicz, and F. Viens, *Bayesian approach to model-based extrapolation of nuclear observables*, Phys. Rev. C, 98 (2018), p. 034318.

[100] R. N. Pérez, J. Amaro, and E. R. Arriola, *Uncertainty quantification of effective nuclear interactions*, International Journal of Modern Physics E, 25 (2016), p. 1641009.

[101] R. N. Pérez and E. R. Arriola, *Uncertainty quantification and falsification of chiral nuclear potentials*, The European Physical Journal A, 56 (2020), pp. 1–24.

[102] J. Piekarewicz, W.-C. Chen, and F. Fattoyev, *Information and statistics: a new paradigm in theoretical nuclear physics*, Journal of Physics G: nuclear and particle physics, 42 (2015), p. 034018.

[103] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in Fortran 77: the art of scientific computing*, vol. 2, Cambridge university press Cambridge, 1992.

[104] A. Radford, L. Metz, and S. Chintala, *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*, arXiv e-prints, (2015), p. arXiv:1511.06434.

[105] P. S. Rajput, K. Satis, S. Dellarosa, W. Huang, and O. Agba, *cgans for cartoon to real-life images*, 2021.

[106] P. Ramachandran, B. Zoph, and Q. V. Le, *Searching for activation functions*, arXiv preprint arXiv:1710.05941, (2017).

[107] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[108] W. A. Richter and B. A. Brown, ^{26}Mg *observables for the usda and usdb hamiltonians*, Phys. Rev. C, 80 (2009), p. 034301.

[109] W. A. Richter, S. Mkhize, and B. A. Brown, *sd-shell observables for the usda and usdb hamiltonians*, Phys. Rev. C, 78 (2008), p. 064302.

[110] W. A. Richter, S. Mkhize, and B. A. Brown, *sd-shell observables for the usda and usdb hamiltonians*, Phys. Rev. C, 78 (2008), p. 064302.

[111] X. Roca-Maza, N. Paar, and G. Colò, *Covariance analysis for energy density functionals and instabilities*, Journal of Physics G: Nuclear and Particle Physics, 42 (2015), p. 034033.

[112] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, arXiv e-prints, (2015), p. arXiv:1505.04597.

[113] F. Rosenblatt, *The perceptron: a probabilistic model for information storage and organization in the brain.*, Psychological review, 65 (1958), p. 386.

[114] L. Roszkowski, E. M. Sessolo, and S. Trojanowski, *Wimp dark matter candidates and searches—current status and future prospects*, Reports on Progress in Physics, 81 (2018), p. 066201.

[115] J. M. Salter, D. B. Williamson, J. Scinocca, and V. Kharin, *Uncertainty quantification for computer models with spatial output using calibration-optimal bases*, Journal of the American Statistical Association, 114 (2019), pp. 1800–1814. arXiv: 1801.08184.

[116] N. Schunck, J. D. McDonnell, D. Higdon, J. Sarich, and S. Wild, *Quantification of uncertainties in nuclear density functional theory*, Nuclear Data Sheets, 123 (2015), pp. 115–118.

[117] N. Schunck, J. D. McDonnell, J. Sarich, S. M. Wild, and D. Higdon, *Error analysis in nuclear density functional theory*, Journal of Physics G: Nuclear and Particle Physics, 42 (2015), p. 034024.

[118] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, *Mastering the game of go with deep neural networks and tree search*, Nature, 529 (2016), p. 484–489.

[119] D. S. Sivia and J. Skilling, *Data Analysis: A Bayesian Tutorial*, Oxford Science Publications, 2 ed., 2006.

[120] S. R. Stroberg, J. Henderson, G. Hackman, P. Ruotsalainen, G. Hagen, and J. D. Holt, *Systematics of* $e2$ *strength in the sd shell with the valence-space in-medium similarity renormalization group*, Phys. Rev. C, 105 (2022), p. 034333.

[121] S. R. Stroberg, H. Hergert, S. K. Bogner, and J. D. Holt, *Nonempirical interactions for the nuclear shell model: an update*, Annual Review of Nuclear and Particle Science, 69 (2019), pp. 307–362.

[122] S. R. Stroberg, H. Hergert, S. K. Bogner, and J. D. Holt, *Nonempirical interactions for the nuclear shell model: An update*, Annual Review of Nuclear and Particle Science, 69 (2019), pp. 307–362.

[123] J. Suhonen, *From nucleons to nucleus: concepts of microscopic nuclear theory*, Springer Science & Business Media, Berlin, 2007.

[124] R. Surman, M. Mumpower, J. Cass, I. Bentley, A. Aprahamian, and G. C. McLaughlin, *Sensitivity studies for r-process nucleosynthesis in three astrophysical scenarios*, EPJ Web of Conferences, 66 (2014), p. 07024.

[125] T. M. Undagoitia and L. Rauch, *Dark matter direct-detection experiments*, Journal of Physics G: Nuclear and Particle Physics, 43 (2015), p. 013001.

[126] R. Utama and J. Piekarewicz, *Refining mass formulas for astrophysical applications: A Bayesian neural network approach*, Physical Review C, 96 (2017), p. 044308.

[127] U. van Kolck, *Few-nucleon forces from chiral lagrangians*, Phys. Rev. C, 49 (1994), pp. 2932–2941.

[128] J. D. Walecka, *Theoretical Nuclear and Subnuclear Physics; 2nd ed.*, World Scientific, Singapore, 2004.

[129] M. Wang, G. Audi, A. H. Wapstra, F. G. Kondev, M. MacCormick, X. Xu, and B. Pfeiffer, *The ame2012 atomic mass evaluation (ii). tables, graphs and references*, Chin.Phys.C, 36 (2012), p. 1603.

[130] L. Weissman, O. Arndt, U. Bergmann, J. Cederkall, I. Dillmann, O. Hallmann, L. Fraile, S. Franchoo, L. Gaudefroy, U. Köster, K. L. Kratz, A. F. Lisetskiy, B. Pfeiffer, O. Sorlin, and S. Tabor, *β decay of* ^{26}Ne, Phys. Rev. C, 70 (2004), p. 057306.

[131] K. Wendt, B. Carlsson, and A. Ekström, *Uncertainty quantification of the pion-nucleon low-energy coupling constants up to fourth order in chiral perturbation theory*, arXiv preprint arXiv:1410.0646, (2014).

[132] K. A. Wendt, C. Forssén, T. Papenbrock, and D. Sääf, *Infrared length scale and extrapolations for the no-core shell model*, Physical Review C, 91 (2015), p. 061301.

[133] S. Wesolowski, R. Furnstahl, J. Melendez, and D. Phillips, *Exploring bayesian parameter estimation for chiral effective field theory using nucleon–nucleon phase shifts*, Journal of Physics G: Nuclear and Particle Physics, 46 (2019), p. 045102.

[134] T. Whitehead, T. Poxon-Pearson, F. Nunes, and G. Potel, *Prediction of (p, n) charge-exchange reactions with uncertainty quantification*, arXiv preprint arXiv:2112.14256, (2021).

[135] J. Xu, Z. Zhang, and B.-A. Li, *Bayesian uncertainty quantification for nuclear matter incompressibility*, Physical Review C, 104 (2021), p. 054324.

[136] D. Yarotsky, *Error bounds for approximations with deep relu networks*, Neural Networks, 94 (2017), pp. 103–114.

[137] Y. Yazici, C.-S. Foo, S. Winkler, K.-H. Yap, G. Piliouras, and V. Chandrasekhar, *The Unusual Effectiveness of Averaging in GAN Training*,

arXiv e-prints, (2018), p. arXiv:1806.04498.

[138] S. Yoshida and N. Shimizu, *Constructing approximate shell-model wavefunctions by eigenvector continuation*, (2021).

[139] S. Yoshida, N. Shimizu, T. Togashi, and T. Otsuka, *Uncertainty quantification in the nuclear shell model*, Phys. Rev. C, 98 (2018), p. 061301.

[140] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, arXiv e-prints, (2017), p. arXiv:1703.10593.

[141] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, *Toward Multimodal Image-to-Image Translation*, arXiv e-prints, (2017), p. arXiv:1711.11586.

.1 The `SHMUQ` codes: Shell Model Uncertainty Quantification

Doing UQ for nuclear shell model required I develop a system to automate the calculations. `SHMUQ` is a module and series of codes written in Python with the objective of providing basic building blocks for complicated calculations using multiple production codes. Some capabilities are as follows.

- For a given effective Hamiltonian with the form $\hat{H} = \sum_i \lambda_i \hat{o}_i$ and[1] a set of states $\{|\psi_\alpha; J, T, n\rangle\}$ for many nuclides, `SHMUQ` performs all `BIGSTICK` calculations required to get the expectation values $\langle\psi_\alpha|\hat{o}_i|\psi_\alpha\rangle$. These are then used for sensitivity analysis (SA) of $\boldsymbol{\lambda}$.
- The sensitivity analysis (SA) consists of a principle component analysis, which allows us to determine the most important linear combinations of matrix elements with respect to energies of the states $E_\alpha = \langle\psi_\alpha|\hat{H}|\psi_\alpha\rangle$.
- Given the resulting sensitivity analysis (SA), approximate the distribution $P(\boldsymbol{\lambda})$ with a Gaussian and sample new interactions $\boldsymbol{\lambda}' \sim \mathcal{N}(\text{average} = \mu_{\boldsymbol{\lambda}}, \text{covariance} = C_{\boldsymbol{\lambda}})$.
- Given the resulting sampled interaction $\boldsymbol{\lambda}'$, compute a set of states $\{|\psi_\alpha(\boldsymbol{\lambda}')\rangle\}$ for many nuclides (potentially different than the original set used for SA). Note that the wavefunction $|\psi_\alpha(\boldsymbol{\lambda})\rangle$ is smooth in $\boldsymbol{\lambda}$: small perturbations in the Hamiltonian yield small changes in the eigenvectors. In principle one should be able to "track" a state as $\boldsymbol{\lambda}$ changes. However, given some solutions evaluated at two points, $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}'$, matching pairs of states $|\psi_\alpha(\boldsymbol{\lambda}); J, T, n\rangle \leftrightarrow |\psi_\alpha(\boldsymbol{\lambda}'); J, T, n\rangle$ is nontrivial. I will elaborate on this problem, "state matching", later in this section.
- With the states produced by the sampled interaction, compute other observables. This could include transition matrix elements of one-body operators, $M_{i\to f} = \langle\psi_f|\hat{O}|\psi_i\rangle$, or expectation values of two-body operators $M_\alpha = \langle\psi_\alpha|\hat{O}|\psi_\alpha\rangle$. This provides the mapping from interactions to observables, $P(\boldsymbol{\lambda}) \to P(O|\boldsymbol{\lambda})$.
- If the observable depends on parameters of interest, $O = O(\theta)$, then we model θ as well.

.1.1 Challenges

---

[1] All Hamiltonians we deal with have this form, a linear combination of density operators. As such the terms "Hamiltonian" and "interaction" are mostly interchangeable. The Hamiltonian *matrix*, however, depends on the basis, number of particles, angular momentum, etc.

.1.1.1 State matching

The eigenvectors and eigenvalues the Hamiltonian $\hat{H}(\lambda)$ are perturbative with respect to the parameters λ: a small change in λ produces a small change in the solutions, and thus also a small change in the integrals upon which solutions depend. The individual mappings $\lambda \to \langle\psi(\lambda)|\hat{O}|\psi(\lambda)\rangle$ are usually not very complicated, and a polynomial fit does well to emulate the model around a point. However, when λ is chosen randomly, it is possible to "lose track" of the solution due to state mixing and/or level crossings. If the solution of interest has mixed too much with another state, or has changed order in the list of total excitations of the system as shown in Fig. 1, then it can be very difficult or impossible to connect within the overall framework.

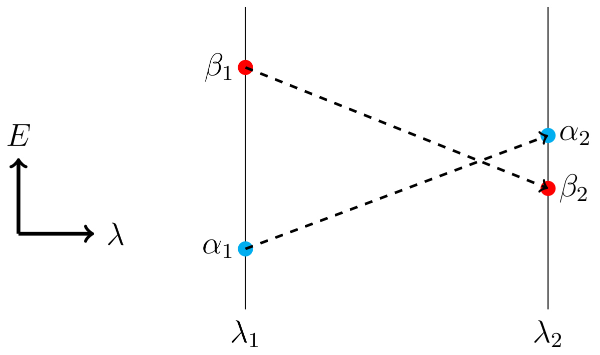


Figure 1. For two discrete samples of parameters λ_1 and λ_2 the energies of states α and β change order.

This is handled within `SHMUQ` using overlaps. Wavefunctions $|\psi\rangle$ in the calculation are tracked not by J^π_n but rather by computing the overlap, $|\langle\psi|\psi_{\text{ref}}\rangle|^2$, with a reference state $|\psi_{\text{ref}}\rangle$. The reference states are eigenstates of the unperturbed Hamiltonian. This system is not too slow, and ensures that states are tracked even when they may switch order with one another. Of course, this requires one set a lower limit on the acceptable overlap; for overlaps less than around 0.5, we do not accept the result. If this occurs, `SHMUQ` makes a new sampled Hamiltonian and the calculation starts over. During the work in Chapter 5, this occurred with very low probability, less than 0.01% of the time.

.1.1.2 Extreme I/O

A practical challenge faced when sampling large calculations for UQ is management of output files and disk storage. Simulation codes are typically designed with verbosity as a high priority, and this makes a lot of sense: delivering more information to the user is rarely a bad thing, especially if that information can be conveniently organized into multiple text and/or data files. However, it became apparent early during the development of `SHMUQ` that some built-in file management would be required. The problem is that disk space required for UQ increases *geometrically* as we broaden the analysis.

$$N_{\texttt{SHMUQ}} \sim N_{\text{observables}} N_{\text{samples}} N_{\text{experiment}} N_{\text{states}} \cdots \tag{1}$$

Here $N_{\texttt{SHMUQ}}$ is some abstract quantity representing the size of the UQ, and is related to disk space required and/or the number of files produced. That is, our analysis scales like the product of number of observables considered, number of samples for statistics, number of experimental data points, number of solutions required to produce good data, etc. For UQ of operator parameters in the *sd*-shell (for E2, M1, and GT, and the sum-rule version of each), we have the following (rough) evaluation of this expression.

$$N_{\texttt{SHMUQ}} \sim (6)(10,000)(150)(40) = 360,000,000 \tag{2}$$

This is certainly an order-of-magnitude calculation, but clearly we can reach the scale of millions of files and millions of MB of data on disk. Not all of those files must be kept, but implementing “clean-up” functions in the code itself takes much care and caution on the part of the developer. The typical method used in `SHMUQ` is to stage unnecessary files for deletion, then archive the important files using `tar` along with compression program `pigz`, which is a multithreaded version of `gzip`. A Python script then copies the necessary files out of archive, and only after that returns are the trash files finally deleted. While this strategy ultimately reduces $N_{\texttt{SHMUQ}}$ significantly, archiving files on this scale can be extremely slow.

.2 Quantum mechanical formalism

I include this section for the interested reader not already familiar with the formalism of quantum mechanics, but with basic knowledge of linear algebra and calculus.

This work is concerned with quantum mechanical systems of many particles (> 3) at low energy. Low energies implies non-relativistic theory, so the language of non-relativistic quantum mechanics (QM) is usually appropriate, although a little bit of non-relativistic field theory is used as well. Quantum mechanics for a fixed number of particles is often formally done in **Hilbert spaces**, which are infinite-dimensional vector spaces with an inner product rule. In practice though, we work in finite bases – I will explain those in detail later. In the Hilbert space, states of the many-body system are represented by vectors, so working with these states is in large part linear algebra, altogether forming the invaluable *matrix formulation of quantum mechanics.*

.2.1 The matrix formulation of quantum mechanics: kets, bras, operators, and all that

A vector in our Hilbert space is written in Dirac's **bra-ket notation**: $|\psi\rangle$ is a state with the label ψ. [2] A vector $|\psi\rangle$ is a ***ket* vector**, meaning it is oriented like a column vector. A vector $\langle\psi|$ is a ***bra* vector**, meaning it is oriented like a row vector. Transforming $|\psi\rangle \rightarrow \langle\psi|$ is the same as a *conjugate transpose.* Just like regular vectors,

---

[2] We often label states with its quantum numbers. So, if the state $|\psi\rangle$ is an eigenvector of an operator, we can specify according to that information. Say some vectors $|\psi\rangle$ have spin $S = 2$, then I could write those as $|\psi; S = 2\rangle$. Or, if we want to write a general eigenstate of the spin operator, we may just write $|S\rangle$. Labels are flexible to accommodate whatever properties we need to keep track of.

kets and bras are closed under addition and scalar multiplication, although they technically exist in two distinct vector spaces. For a Hilbert space $\mathcal{H}$ defined over the complex numbers, a bra (dual) space $\mathcal{H}^*$, vectors $|a\rangle$ and $|b\rangle \in \mathcal{H}$, and a complex number $z \in \mathbb{C}$, we have the following general relations.

$$\begin{aligned} |a\rangle + |b\rangle &\in \mathcal{H} \\ \langle a| + \langle b| &\in \mathcal{H}^* \\ z|a\rangle &\in \mathcal{H} \\ z\langle a| &\in \mathcal{H}^* \end{aligned} \tag{3}$$

The Hilbert space is also equipped with a number of products: inner and outer come up the most often. Just like a dot product between vectors $\mathbf{x}, \mathbf{y}$ can be written like $\mathbf{x} \cdot \mathbf{y}$ or $\mathbf{x}^T\mathbf{y}$, or for complex vectors like $\mathbf{x}^*\mathbf{y}$ or $\mathbf{x}^\dagger\mathbf{y}$ or $\mathbf{x}^H\mathbf{y}$, the **inner product** in our Hilbert space is written $\langle x|y\rangle$. The usual inner product between two vectors is defined as the integral over their product, but with one complex-conjugated.

$$\int_{\text{All possible } q} dq\, \psi_i^*(q)\psi_j(q) = \langle \psi_i|\psi_j\rangle \tag{4}$$

Here, q can be anything: coordinates, frequencies, spins, etc. Notice I had to introduce a new variable q for the integral, but not in the inner product in Dirac notation. Indeed, computing an integral requires a **basis**, but the inner product is independent of basis. Put another way, the numerical value of the inner product does not depend on what computational basis is used. Our bra and ket vectors are *without representation*, and they only gain a representation when we put them in a basis.

For example, consider a 1D coordinate basis $\{|x_i\rangle\}_{i=1,\ldots,N}$ versus a 1D Fourier basis $\{|k_i\rangle\}_{i=1,\ldots,N}$, both with N basis vectors[3]. That means, each $|x\rangle$ is localized at one point in coordinate space, and each $|k\rangle$ is localized at one point in Fourier space. Just

---

[3] I realize I introduced the idea of infinite-dimensional bases, then gave an example of a finite dimensional one. This is intentional. It is easier to define concepts like an inner product formally as integrals first, then in practice define them again as sums. Computationally, all our bases are finite.

like coordinate vectors are distinguished by their position x, Fourier modes are distinguished by their wavenumber k, and the process of going between these two is called a Fourier transform (or inverse Fourier transform). The components representing some arbitrary vector $|\psi\rangle$ are obtained by the inner product of it with each basis vector: $\psi(x_i) = \langle x_i|\psi\rangle$ and $\psi(k_i) = \langle k_i|\psi\rangle$. Remember: while the coefficients of the vector are basis dependent, the vector $|\psi\rangle$ itself does not have any preferred representation. It is helpful to express both these bases in the $|x\rangle$ basis: $\langle x|x'\rangle = \delta(x - x')$ and $\langle x|k\rangle = \cos kx + i \sin kx$.

To be clear, the quantity $\psi(x_i) = \langle x_i|\psi\rangle$ is not a vector, it is a number. In order to represent the vector in a basis (that is, in terms of other vectors), we will apply a **completeness relation**, shown in Eq. 5.

$$\sum_{i=1}^{N} |x_i\rangle\langle x_i| = I. \tag{5}$$

This is an extremely handy object. Note that $|x_i\rangle\langle x_i|$ is an **outer product** of a vector with itself, and thus is a matrix. Algebraically, we can always insert an identity matrix without changing the value of an expression. By the completeness relation, if we know a useful basis, we can always find a way to evaluate the expression in that basis. There is an integral version of completeness too,

$$\int dq\, |q\rangle\langle q| = \hat{1} \tag{6}$$

where $\hat{1}$ is the identity in infinite dimensions. One should be careful to distinguish though – these statements are similar but not interchangeable. With this knowledge, we can return to Eq. 4 and see how the inner product and completeness relation are related.

$$\langle \psi_i|\psi_j\rangle = \langle \psi_i|\hat{1}|\psi_j\rangle = \int dq \langle \psi_i|q\rangle\langle q|\psi_j\rangle = \int dq\, \psi_i^*(q)\psi_j(q) \tag{7}$$

Bases used for computation often have an additional property called *orthonormality*: the inner product of two basis vectors is only nonzero if the vectors are equal, in which case the inner product is 1. We use the Kronecker delta: $\delta_{ab} = 1$ only if $a = b$, otherwise = 0.

$$\langle x_i | x_j \rangle = \delta_{ij} \tag{8}$$

In nuclear theory, the state vectors and associated objects get called by different names, and any nuclear theorist would admit these can be a little confusing for newcomers. Usually state vectors represent one or more particles, so they are *wavefunctions*. If they are a solution to an eigenvalue problem, which comes up often, they are called *eigenvectors* or *eigenfunctions* or *eigenstates*. The inner product of two states $\langle x_i | x_j \rangle$ may be called an *overlap*, because we often use it as a measure of similarity. If an operator $\hat{O}$ is inserted between, then the number $\langle x_i | \hat{O} | x_j \rangle$ is typically called a *matrix element*, because it is the coefficients of the matrix representation of $\hat{O}$ in the basis $\{|x_i\rangle\}$.

Operators represent what we call *external fields*, although I think it is helpful to think of them as representing properties of the quantum mechanical system (like position), or processes the system can undergo (like decay). We denote operators with a hat: $\hat{O}$. Calculating a property of a state $|\psi\rangle$ is equivalent to an **expectation value**; position may be represented by the operator $\hat{x}$ and measurement of position is calculated as a modified inner product like in Eq. 9.

$$\langle \hat{x} \rangle_\psi = \langle \psi | \hat{x} | \psi \rangle = \int dx \, \psi^*(x) x \psi(x) = \int dx \, x |\psi(x)|^2 \tag{9}$$

From here, we can take a closer look at expansion of a wavefunction in a basis, and statistical interpretations of the wavefunction.

$$|\psi\rangle = I|\psi\rangle = \sum_i |x_i\rangle\langle x_i|\psi\rangle = \sum_i c_i |x_i\rangle \tag{10}$$

where $c_i = \langle x_i|\psi\rangle$. For now, imagine ψ is a single particle in a 1D space coordinate x, so the basis $\{|x\rangle\}$ fully spans a 1D coordinate space and this expression poses our wavefunction $|\psi\rangle$ as a scalar function of space $\psi(x)$. The values of the function may be computed as $\psi(x = x') = \langle x'|\psi\rangle$. In this case, measuring the x position of our particle delivers an eigenstate of $\hat{x}$, $|x\rangle$, with probability given by the square of the wavefunction at that point: $\psi^2(x = x') = |\langle x'|\psi\rangle|^2$. For M independent preparations of the state $|\psi\rangle$, the measurement of $\hat{x}$ will "reveal" the particle at $x = x'$ a total $M|\langle x'|\psi\rangle|^2$ times, on average. Upon measurement, the wavefunction becomes an eigenstate of the operator; for $\hat{x}$, every $|x\rangle$ is an eigenstate, so the act of measurement has taken our system from an admixture of many positions to a single position. In other words, if we measure the position of the particle to be $x = x'$, then the wavefunction immediately after the measurement *is* $|x'\rangle$. For more on statistical interpretations see an introductory quantum mechanics text like those by Griffiths or Shankar.

By the probability interpretation we can clearly see that the *scale* of wavefunction coefficients c_i is important – since the probabilities of all possible states must equal 1, we enforce that our wavefunction be **normalized**.

$$\sum_i \Pr(\langle\psi|\hat{x}|\psi\rangle = x_i)_i = \sum_i \Pr(\langle\psi|\hat{k}|\psi\rangle = k_i)_i = 1 \tag{11}$$

The process of normalization is to divide the wavefunction by the square root of its inner product with itself.

$$|\psi'\rangle = \frac{|\psi\rangle}{\sqrt{\langle\psi|\psi\rangle}} \tag{12}$$

So, $|\psi'\rangle$ is properly normalized, and thus we can think of $|\psi'(x)|^2$ as a **probability density function** in the coordinate basis. This can be generalized to other bases which are formed by eigenstates of an operator, such as the Fourier basis with the wavenumber operator $\hat{k}$ or spin basis with spin operator $\hat{S}$.

.2.2 Quantum angular momentum

Recall from classical mechanics the formula for calculating angular momentum from position and translational momentum: $\boldsymbol{L} = \boldsymbol{r} \times \boldsymbol{p}$. For quantum mechanics, we must also account for *intrinsic* angular momenta of particles, commonly called **spin**. Systems with spin have angular momentum but may be point particles with no structure to be rotating through space, so the classical definition does not work. The quantum analogue to classical $\boldsymbol{L}$ is **orbital angular momentum** $\hat{\boldsymbol{L}} = (\hat{L}_x, \hat{L}_y, \hat{L}_z)$, and we define intrinsic spin as a new quantity $\hat{S} = (\hat{S}_x, \hat{S}_y, \hat{S}_z)$ which follows all the same rules as $\hat{\boldsymbol{L}}$. In nuclear physics especially these two types of angular momentum will combine to give **total angular momentum** $\hat{\boldsymbol{J}} = \hat{\boldsymbol{L}} + \hat{\boldsymbol{S}} = (\hat{L}_x + \hat{S}_x, \hat{L}_y + \hat{S}_y, \hat{L}_z + \hat{S}_z) = (\hat{J}_x, \hat{J}_y, \hat{J}_z)$.

Quantum angular momentum follows the SU(2) commutation relations in Eq. (13).

$$[\hat{J}_x, \hat{J}_y] = i\hbar\hat{J}_z \qquad [\hat{J}_z, \hat{J}_x] = i\hbar\hat{J}_y \qquad [\hat{J}_y, \hat{J}_z] = i\hbar\hat{J}_x \tag{13}$$

Since individual components don't commute, they cannot be known simultaneously to arbitrary precision, and so we often work in terms of two operators which do commute: $\hat{J}_z$ and $\hat{J}^2 = \hat{J}_x^2 + \hat{J}_y^2 + \hat{J}_z^2$. Consider a state $|J, J_z\rangle$ which is an eigenvector of both. We often label a vector with its quantum numbers so it is obvious these conditions are satisfied.

$$\begin{gathered} \hat{J}_z|J, J_z\rangle = J_z|J, J_z\rangle \\ \hat{J}^2|J, J_z\rangle = J(J+1)|J, J_z\rangle \end{gathered} \tag{14}$$

As expected, the eigenvalue of the operator $\hat{J}_z$ is the z-component of the total angular momentum of the system, J_z. However, the eigenvalue of the operator $\hat{J}^2$ is not J^2 but $J(J+1) = J^2 + J$. The extra factor of J is generated by the non-commutativity in Eq. 13.