



**Trinity
College
Dublin**

The University of Dublin

School of Physics

**Genetic algorithms and graph
neural network applications for
molecular magnetism**

Lion Frangoulis

Supervisor: Prof. Alessandro Lunghi

Year 2026

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Master of Science

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work. I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement. I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

This text includes published work of others, acknowledged in the thesis.

Abstract

Molecular magnets have been under active investigation for more than three decades due to their capability of storing magnetic information on a molecular level. However, their applications remain limited to this day, due to spin-phonon relaxation processes destroying this information. Calculating the full relaxation rate is an expensive task, and therefore, the discovery of high relaxation time molecular magnets is a slow process. However, it has been linked to the magnetic anisotropy of the compounds in question, allowing this property to be used as a proxy to speed up this process.

We implement two methods that, while having found widespread application inside and outside of chemistry, have not been used in the discovery of molecular magnets: In a first step, we develop two flavors of genetic algorithms that are capable of discovering new compounds on their own using on-the-fly *ab initio* methods. The chemical space investigated can either be spanned by supplying a list of ligands that are assembled into a coordination compound or a set of rules allowing the algorithm to develop new ligands on its own. The resulting algorithms are benchmarked against a previously computed dataset, proving their speedup over random search methods, and then subsequently applied to Co(II) and Dy(III) compounds, discovering new and record-breaking compounds.

In a second step, we develop different kinds of graph neural networks, capable of predicting the Kramers energy levels in Dy(III) compounds responsible for their magnetic behavior. These models are trained on a dataset specifically designed for this task, which also provides the largest dataset of Dy(III) compounds to our knowledge. The different models are fed different levels of complexity in input data, and can, in addition to the invariant Kramers energies, also predict the spin Hamiltonian itself, allowing for the computation of other magnetic properties as well.

In conclusion, this work presents two novel algorithms that, to our knowledge, have not been applied to molecular magnets before, with the capabilities to speed up calculations significantly or reduce the amount of calculations required by sampling the chemical space more efficiently.

Acknowledgements

I would like to thank my supervisor Professor Alessandro Lunghi: Not only did he support and nourish my academic growth, but also my personal development. He supported me in my decisions and future opportunities, and I could not have hoped for a better or more dedicated supervisor.

I am especially grateful to Dr. Zahra Khatibi as well. For the endless hours of support, encouragement and distraction in the office, which made many of the long days feel much shorter. Not only did she act as an academic role model, but also a dear friend and constant companion in my journey through this degree.

I would also like to thank Dr. Lorenzo Mariano, who is a near-infinite well of information and experience on electronic structure methods, and without whom I could not have performed the calculations required for this work.

Last, I would like to thank my friends, both in Dublin, at home, and across the globe, for all the support over the years, I could not have wished for better ones. Especially Valerio, Silvia, Anita, Eoin, and Camille, who always had my back during good or bad times, and everyone else at Westland Square, who created such a special community to work in.

Publications

- Lion Frangoulis[†], Zahra Khatibiz[†], Lorenzo Marianno and Alessandro Lunghi, "Generating New Coordination Compounds via Multireference Simulations, Genetic Algorithms, and Machine Learning: The Case of Co(II) and Dy(III) Molecular Magnets", JACS Au 5, 8, 2025
- Lion Frangoulis, Prajakta Jadhav, Alessandro Lunghi, "Graph Neural Networks for Spin Hamiltonian Prediction of Dy(III) Molecular Magnets", in preparation

[†] Authors contributed equally.

Conferences

- European conference on molecular magnetism, 15th-19th July 2024, Krakow (Poland). Poster presentation.
- 4th European Conference on Molecular Spintronics, 2nd-4th June 2025, Dublin (Ireland). Poster presentation.

Contents

1	Scientific background	1
1.1	Development of Molecular Magnetism	1
1.2	Anisotropy and spin Hamiltonians	2
1.3	Numerical Methods	6
1.4	Promising molecular systems for high spin lifetimes	8
1.5	Current Challenges and structure of this thesis	9
2	Results - Genetic Algorithm	13
2.1	Methodology	13
2.1.1	SMILES and SELFIES	13
2.1.2	Genetic algorithms	14
2.1.3	Ligand extraction	17
2.1.4	Compound creation	18
2.1.5	Electronic structure simulations.	18
2.2	Results	19
2.2.1	Static genetic algorithm	20
2.2.2	Semi-dynamic genetic algorithm	24
2.3	Discussion	30
3	Results - Graph Neural Networks	33
3.1	Methodology	33
3.1.1	Graph neural networks	33

3.1.2	Features	35
3.1.3	GNN Implementation	37
3.2	Results - Dataset Preparation	38
3.3	Results - GNN	41
3.3.1	Identical Components	42
3.3.2	alternative simplified DNN using chemical intuition	44
3.3.3	Two body architecture	47
3.3.4	Three body architecture	50
3.3.5	Equivariant three body architecture	53
3.4	Discussion	58
4	Conclusion and further development	61
A	Analysis of Tesseral basis	65

List of Figures

1.1	Original Mn 12. The original Mn12 molecular magnet containing 12 manganese ions (4 Mn 4+, 8 Mn 3+), connected by oxygen bridges, and corresponding magnetic hysteresis loop. (Reproduced with permission from [2])	1
1.2	Ising type anisotropy behavior. Panel a) shows the two degenerate ground states in the absence of a magnetic field with equal population. Panel b) shows the population imbalance induced by a lowering of the second eigenstates energy due to an external magnetic field. Panel c) shows the situation right after deactivation of the magnetic field, the ground state is degenerate, but the population imbalance is preserved. Panel d) shows the relaxation back to equilibrium through relaxation processes.	3
1.3	Molecular Magnets. Examples of Co(II) (left) and Dy(III) (right) molecular magnets discovered in this work.	4
1.4	Co(II) configurations. The actual and effective spin levels of Co(II) for the two cases of $D>0$ and $D<0$. Reproduced with permission from [10]. For negative D , the high $ m_S $ doublet is the groundstate, leading to an easy axis anisotropy capable of reducing spin relaxation, while positive D favors the low $ m_S $ states and leads to an easy plane anisotropy.	5
1.5	Dy(III) energy levels for a bi-pyramidal compound. The 16 states are separated into Kramers pairs, with the lower ones sorted by their z-alignment m_J . This creates an energy barrier of at least ΔE_{01} to overcome for a spin to relax from full positive to full negative alignment. Reproduced with permission from [14]	7

2.1	GA workflow. The workflow of a genetic algorithm begins by initializing a random population and evaluating their fitness function. The second part is the iterative part, where through a combination of selection, crossover and mutation new generations are created and subsequently evaluated, until the termination criteria is fulfilled. Once this criteria is fulfilled, an entire population of highly suitable solutions is returned.	15
2.2	Genetic algorithm benchmark. (a) Comparison of minimum D growth for the random sampling approach and GA with different population sizes, performed on the COMPASS test set and averaged over 1000 runs. The termination criteria is set to find one of the top three compounds in the set. The inset shows the speedup and average number of generations needed for termination of run for different population sizes. (b) Comparison of compound evaluations needed for random sampling and GA with different population sizes, versus the termination criteria—finding one of the top 1,3,5,10 or 20 compounds within the COMPASS dataset. The inset shows the speedup that GA with different population sizes enables compared to the random sampling.	21
2.3	Results for the GA with static encoding. a) Minimum and average anisotropy growth of two separate static GA runs. The minimum curves of both runs have a very jagged growth, as expected, while the average continuously goes down. However, both reach the $< -200 \text{ cm}^{-1}$ regime. (b) Temporal evolution of GA-produced compounds and their anisotropy, with the new population discovered at regular intervals of generations shown in different colors. The GA population at each generation roughly follows a skewed normal distribution with later generation concentrating and converging towards higher D values through a steady progression. (c) Cumulative distribution of compounds to compare the GA and random sampling performances, with the GA showing a higher relative population in the medium negative regime. (d) Top performing compounds and their magnetic anisotropy. None of the top compounds preserved the original tetrahedral geometry, however, all show reasonable optimized structures. (Cyan: Co, Red: O, Grey: C, White: H, Beige: F, Blue: N.)	23

2.4	Chosen static ligands. The molecular graph for the two chosen ligands used in the semi-dynamic GA run for tetrahedral Co(II) compounds. The ligands were selected by identifying the most commonly occurring ligands in the top 100 compounds of [21], after removing ligands with connecting atoms heavier than Ar.	26
2.5	Results for the GA with dynamic encoding. (a) Comparison of the minimum and average anisotropy for two dynamic GA runs. The dynamic GA presents a faster progress and convergence starting from an informed guess using two top-performing ligands as the fixed genes and using the RFC pre-screening. (b) Temporal evolution of GA-produced compounds and their anisotropy, with the new population discovered at regular intervals of generations shown in different colors. The GA population is more dispersed compared to static GA with the majority of compounds reaching $< -100 \text{ cm}^{-1}$ as early as 16th generation and a noticeable portion of the compounds accumulating in the $[-200, -100] \text{ cm}^{-1}$ region. (c) Cumulative distribution of compounds generated by the GA (blue) compared to all COMPASS entries that include the fixed ligand. (d) Top compounds and their anisotropy D , again with strong divergence from the original tetrahedral coordination. (Cyan: Co, Red: O, Blue: Nitrogen, Grey: C, White: H, Beige: Si.)	27
2.6	Results for the GA with dynamic encoding on Dysprosium. a) Minimum and average Kramers Gap growth. It very quickly found a single compound with maximum Kramer gap of 615 cm^{-1} , yet the growing average shows the number of other compounds above 550 cm^{-1} . (b) Temporal evolution of GA-produced compounds and their Kramer gap, with the new population discovered at regular intervals of generations shown in different colors. After the initial population, it rapidly grows and consistently produces compounds above 550 cm^{-1} . (c) Cumulative distribution of all compounds produced. (d) Distribution of connecting atoms in new compounds generated at each generation. The oxygen ion is quickly identified as the best candidate and all other species quickly die out completely.	30
2.7	Resulting compounds of Dy(III) GA study. All compounds within the $\Delta E_{01} > 550 \text{ cm}^{-1}$ regime show weak interactions between one or two planar water ligands and each axial ligand, often involving an electron rich group. (Cyan: Dy, Red: O, Grey: C, White: H, Ochre: Cl, Yellow: S.)	31

3.1	GNN runtime analysis. Comparison of the GNN runtime against feature number for all three models for feature number between 8 and 64 and between one and six message passing layers. The linear correlation is clearly visible, with the largest actually used model ($F = 32$, 6 message passing layers) only requiring 12.5 seconds per epoch for the entire dataset.	39
3.2	PCA analysis. First two principal components of the bispectrum components of all ligands in the QM9 and QM9 star dataset fulfilling the element and charge conditions. In black are marked the finally selected ligands.	40
3.3	Kramers Dataset Distribution. Distribution of the k -th Kramers energy for the generated dataset, separated by color. The distribution in each level follows the same trend, with a high peak of undesirable low energy compounds, an even distribution of common compounds, and a small number of high interest compounds, marked in dark blue for the first Kramers level.	41
3.4	Kramers Dataset Distribution. Distribution of the tesseral decomposition O_{lm} , separated by order as $l=2,4,6$ left to right, and separated in magnitude of m by color. The $m=0$ component shows significantly different behavior than the other components, due to the approximate z-alignment for most of the compounds.	42
3.5	DNN architecture. The deep neural network consists of 4 block made out linear layers and activation functions, using the same SiLU function as the GNN as well. The individual dimensions are shown in the figure, with the fifth activation layer without an activation function reducing down to seven outputs to predict the seven Kramers layer. Additionally, to dropout layers with a dropout rate of 0.005 are added to reduce overfitting by reducing artificial noise correlations.	44

3.6	DNN learning analysis. Top: Training (left) and Test (right) set parity plot between the real CASSCF Kramers gaps and the DNN predictions for all 7 Kramers levels, color coded by the axial ligands connecting atom species. The trend between different connecting atoms is accurately recreated, with a small number of outliers that are being underestimated. Bottom: MSE (left) and R^2 (right) learning curve for total data sizes of 1000-10000 compounds with the average and best case scenario shown for five runs. The DNN model reaches high R^2 values quickly, but stagnates rapidly due to the limited complexity of the model. Additionally, the standard deviation at each training size is much larger, showing the reduced stability of this model.	45
-----	--	----

3.7	DNN Kramers prediction. The DNN is capable of predicting the Kramers energies with decent prediction for high values within the respective level, however small values are produced by strong distortion of the bi-pyramidal geometry or ligand-breaking/swapping, leading to bad predictions of the DNN.	47
-----	--	----

3.8 GNN architecture. Panel a) shows the overall architecture of the GNN, separated into an encoding block, three message passing blocks and a readout block, each one detailed below. Each layer acts either on an atom’s or bond’s feature vector, and is shared between atoms and molecules. As such, the total number of parameters is independent of molecule size or number. Each layer apart from activation layer have the input and output dimension marked to the right. The encoding layer assigns each atom a feature vector of dimension 32 depending solely on the element, shared across all atoms and molecules. Panel b) shows one individual message passing layer, using the previous atomic features and RBF vectors as inputs. The Atom wise layers are linear layers applied to individual atom’s features, and the activation layers relate to the Sigmoid Linear Unit (SiLU) function. the messages received from the message creation layer is then post-processed and added back to the features, going from $\mathbf{x}_i^t$ to $\mathbf{x}_i^{t+1}$. Panel c) shows the two-bod message creation layer. The input RBF features are individually processed by two linear and two activation function layers, and in the end multiplied with the message sending atom’s feature. The last panel shows the Kramers readout layer. Two atom wise layers first half the feature number and than reduce it to dimension seven, after which the individual atomic contributions are summed up to predict all seven Kramers energies at once. 48

3.9 Two body GNN learning analysis. Top: Training (left) and Test (right) set parity plot between the real CASSCF Kramers gaps and the two-body feature GNN predictions for all 7 Kramers levels, color coded by the axial ligands connecting atom species. While the trend between different connecting atom species is accurately reproduced, the spread is quite large and the model not suitable to be used as a refined predictor of properties. Bottom: MSE (left) and R^2 (right) learning curve for total data sizes of 1000-10000 compounds with the average and best case scenario shown for five runs. The model reaches R^2 values of 0.8 for dataset sizes of above 5000 compounds, however saturates quickly and displays a high deviation between training runs, leading to noisy result data. While the best case scenario has converged in R^2 and mean error, the average continues to grow, with less noise for larger datasets. 49

3.10	Two body GNN Kramers prediction. Results of the two-body feature GNN model’s Kramer prediction versus the true values, with the training set on the left and test set on the right, however the colors now separate the individual Kramers levels.	50
3.11	SBF architecture. The architecture of the three-body feature is identical to the two-body one apart from the message creation and message passing block. Both SBF and SBF features are used in this layer, with each being processed by two linear and two interaction layers. The RBF results are once again multiplied element-wise with the message sending atom, while the SBF results are multiplied with two message sending atoms. The two final messages are then concatenated before being returned to the message passing layer for further processing.	51
3.12	Three body feature GNN results. Top: Training (left) and Test (right) set parity plot between the real CASSCF Kramers gaps and the three-body feature GNN predictions for all 7 Kramers levels, color coded by the axial ligands connecting atom species. The invariant three-body feature network accurately predicts the trend between different connecting atoms, and achieves high precision predictions for the entire spectrum of compounds and Kramers levels. Bottom: MSE (left) and R^2 (right) learning curve for total data sizes of 1000-10000 compounds. The maximum R^2 (minimum loss) has converged at around 8000 data points, while the average continues to improve until reaching 10000 datapoints. There is a slight overfitting of the test set over the training set, which reaches negligible amounts for large dataset sizes (0.01 R^2 score points).	52
3.13	Three body GNN Kramers prediction. For further analysis, both the train (left) and test (right) parity results of the Kramers energy prediction of the three-body feature GNN model. This time, the individual Kramers levels are separated by color to show the trend predicted: the discrepancy between prediction and true values is smallest for the highest value within each level, corresponding to the model showing the best performance on compounds that closely represent the bi-pyramidal geometry.	53

3.14	Equivariant GNN architecture. The equivariant GNN follows a similar architecture as the three-body one, with the difference in the readout block. The individual atoms' feature vector dimension is reduced to one, and afterwards multiplied with the tesseral vector of order l (and dimension $2l+1$. This guarantees the equivariance of each atom's final feature, and the final results generated through summation of all atoms in a molecule.	54
3.15	Equivariant GNN predictions. Train (top) and test (bottom) parity plot for the $l=2,4,6$ tesseral components (left to right) predicted by the equivariant three body model. The agreement in $l=2$ is excellent with an R^2 score of 0.96, while while $l=6$ falls just behind with $R^2=0.89$. The $l=4$ $m=0$ component in particular behaves visible different than the others, due to azimuthal independence of the component. As such, the atomic basis may not be sufficiently large to cover the entire spectrum of the target values, leading to a final R^2 of 0.61 for this specific component.	55
3.16	Three body GNN performance analysis. The R^2 core of the three body GNN model's tesseral prediction. Due to the different orders of magnitude, overall R^2 score fails as metric and instead is evaluated for each oder in l and m individually. It becomes clear that the prediction in higher order $m = 0$ terms leads to the mayor error, due to those terms non-gaussian distribution and basis misalignment.	55
3.17	Equivariant GNN Learning curve. Left: Overall learning curve of the R^2 score of the equivariant models prediction for the tesseral components, averaged over 5 runs with their mean shown in solid lines and the maximum in dashed. The $l=2$ component dominates the R^2 value, and for further analysis the individuals order are shown on the right in panel b-d). The bad performance in the $l=4$ components stems from the $l=4$ $m=0$ component, which has a highly non Gaussian distribution.	57

3.18	Equivariant GNN Kramers reconstruction. Training (left) and Test (right) set parity plot between the Kramers energies reconstructed from the tesseral components, separated in color by the Kramers level. The true values (x-axis) we also reconstructed from the true tesseral decomposition, since the limited order reconstruction and read-out precision of ORCA introduces additional errors that are not due to the GNN. The prediction is not as accurate as the direct three-body feature prediction, but still shows excellent agreement with an test R^2 score of 0.949.	58
A.1	FC Tesseral $l=2$ The distribution between different m components of the tesseral components of order $l=2$ for all first coordination shell atoms in the dataset. These individual components create the basis used by the GNN to predict the real tesseral components, and show the non-uniform distribution, which can potentially lead to mismatch between this basis and the target values.	66
A.2	True Tesseral $l=2$ The distribution between different m components of the real tesseral components of order $l=2$ extracted from the spin Hamiltonian. These components are the ones to be predicted, and show relatively evenly spread distributions.	67
A.3	FC Tesseral $l=4$ The distribution between different m components of the tesseral components of order $l=4$ for all first coordination shell atoms in the dataset. These individual components create the basis used by the GNN to predict the real tesseral components, and show the non-uniform distribution, which can potentially lead to mismatch between this basis and the target values. Especially the $m=0$ terms have heavily one-sided distributions, leading to an incomplete basis and worse predictions.	68
A.4	True Tesseral $l=4$ The distribution between different m components of the real tesseral components of order $l=4$ extracted from the spin Hamiltonian. These components are the ones to be predicted, and show relatively evenly spread distributions, except for the $m=-1,0,1$ parts, which favor triangular distributions. This mismatch with the basis can lead to reduced prediction accuracy.	69

A.5	FC Tesseral $l=6$	The distribution between different m components of the tesseral components of order $l=6$ for all first coordination shell atoms in the dataset. These individual components create the basis used by the GNN to predict the real tesseral components, and show the non-uniform distribution, which can potentially lead to mismatch between this basis and the target values. Especially the $m=0$ terms have heavily one-sided distributions, leading to an incomplete basis and worse predictions.	70
A.6	True Tesseral $l=4$	The distribution between different m components of the real tesseral components of order $l=4$ extracted from the spin Hamiltonian. These components are the ones to be predicted, and show relatively evenly spread distributions.	71

List of Tables

3.1	DNN results. The R^2 score, RMSE and mean error for both the test and training set prediction of the deep neural network model, separated by Kramers level. While the relative prediction shown by the R^2 score is best for the 3rd and 4th level, the absolute error is smallest for the first, as expected. The model's hyperparameter have been optimized to lead to minimum overfitting. This ensures similar performance in the test and training set, with statistical noise sometimes leading to better performance in the test set.	46
3.2	Two body GNN results. The R^2 score, RMSE and mean error for both the test and training set prediction of the two body model, separated by Kramers level. While the relative prediction shown by the R^2 score is the worst for the first Kramers level, the absolute error is smallest for the first due to the smaller absolute scale. The model's hyperparameter have been optimized to lead to minimum overfitting. This ensures similar performance in the test and training set, with statistical noise sometimes leading to better performance in the test set.	51
3.3	Three body feature GNN results. The R^2 score, RMSE and mean error for both the test and training set prediction of the three body model, separated by Kramers level. While the relative prediction shown by the R^2 score is best for the 3rd and 4th level, the absolute error is smallest for the first, as expected.	54
3.4	Equivariant GNN tesseral results. The R^2 score, RMSE and mean error for both the test and training set prediction of the equivariant body tesseral components. The value are calculated for each order in l separately due to different orders of magnitude, and also for each orders $m=0$ component individually as this term provides the dominant error.	57

3.5	Equivariant GNN results Kramers reconstruction.	The R^2 score, RMSE and mean error for both the test and training set prediction of the equivariant body model's Kramers reconstruction, separated by Kramers level. While the relative prediction shown by the R^2 score is best for the 3rd and 4th level, the absolute error is smallest for the first, as expected.	59
-----	--	---	----

Nomenclature

GNN	Graph Neural Network
GA	Genetic Algorithm
RS	Random Search
RFC	Random Forest Classifier
ZFS	Zero Field Splitting
CF	Crystal Field
SMM	Single Molecule Magnet
RBF	Radial Basis Function
SBF	Spherical Basis Function

1 Scientific background

1.1 Development of Molecular Magnetism

The definition of a molecular magnet can be described as "molecules that show magnetic memory effects arising from purely intramolecular interactions without the need for long-range order" [1]. This phenomenon was first observed for $\text{Mn}_{12}\text{O}_{12}(\text{CH}_3\text{COO})_{16}(\text{H}_2\text{O})_4$ [2] in 1991, better known as Mn_{12} . This compound, shown in figure 1.1 a), consists of four Mn^{4+} ions and eight Mn^{3+} ions, connected through oxo-group bridges, and has the rare property of showing a finite magnetic hysteresis (shown in figure 1.1 b). After applying a magnetic field, the compound retains a finite magnetization along the chosen axis even after the magnetic field is removed, below the magnetic blocking temperature.

This property enables a range of potential applications, including nano-magnetism [3], spintronics [4], and quantum technologies [5]. However, similar to most other modern quantum platforms, the current applications are severely limited by the timescale of magnetic retention: The moment the magnetic field is removed, exponential decay of the magnetization sets in, according to the compound's magnetic relaxation rate τ .

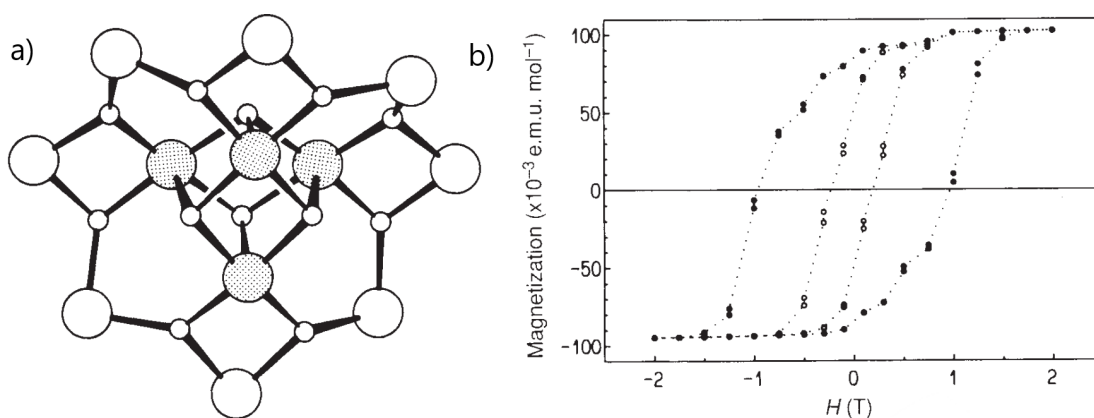


Figure 1.1: **Original Mn 12.** The original Mn12 molecular magnet containing 12 manganese ions (4 Mn 4+, 8 Mn 3+), connected by oxygen bridges, and corresponding magnetic hysteresis loop. (Reproduced with permission from [2])

To understand the origin of the magnetic hysteresis and the following spin relaxation, a model of the ground state energy of a theoretical toy compound dependent on the rotation with respect to a chosen z-axis is shown in figure 1.2 a). The model shows two equally high local minima for perfectly parallel or anti-parallel alignment. In the presence of a magnetic field B (panel b), the state in alignment with the magnetic field is lowered in energy due to the interaction of the spin magnetic moment and the magnetic field, while the other one is raised. The breaking of this degeneracy leads to a population imbalance between the two states. Once the magnetic field is removed, the population imbalance remains (panel c) before decaying back to the original state (panel d).

The magnetic relaxation rate, defined as the inverse of the relaxation time τ , has been shown to follow an Arrhenius-type law:

$$\tau^{-1} = \tau_0^{-1} \cdot \exp(-U_{eff}/k_bT), \quad (1.1)$$

with τ_0 and U_{eff} being compound-specific parameters, with the latter known as the effective energy barrier. For Mn_{12} , the effective barrier is $U_{eff}/k_b=61$ K, and the magnetic hysteresis has been observed for temperatures of up to 4 K, making applications outside of laboratory conditions nonviable. Since then, a race toward the design of compounds with large U_{eff} and blocking temperature has taken place. Over more than three decades of active research, a wide range of open-shell coordination compounds has been investigated, expanding to most transition metals and lanthanides, and a large number of different coordination geometries [6–8]. The number of spin centres has been varied as well, eventually leading to compounds with only a single spin center, known as single-ion magnets (SIMs), and a blocking temperature up to 80 K [9]. In this work, we chose to focus on this class of compounds and, in particular, on tetra-coordinated Co(II) and pentagonal bi-pyramidal Dy(III) complexes (shown in 1.3 a) and b)), which are known to exhibit record magnetic properties.

1.2 Anisotropy and spin Hamiltonians

Given a magnetic ion in vacuum, the different orientations of its magnetic moment, or spin, are perfectly degenerate. The preference of the ground state towards a certain direction, known as magnetic anisotropy, must therefore have its origin outside of the metal ion, more specifically in the surrounding ligands and crystal structure. A framework to model the impact of these external influences on the spin energy

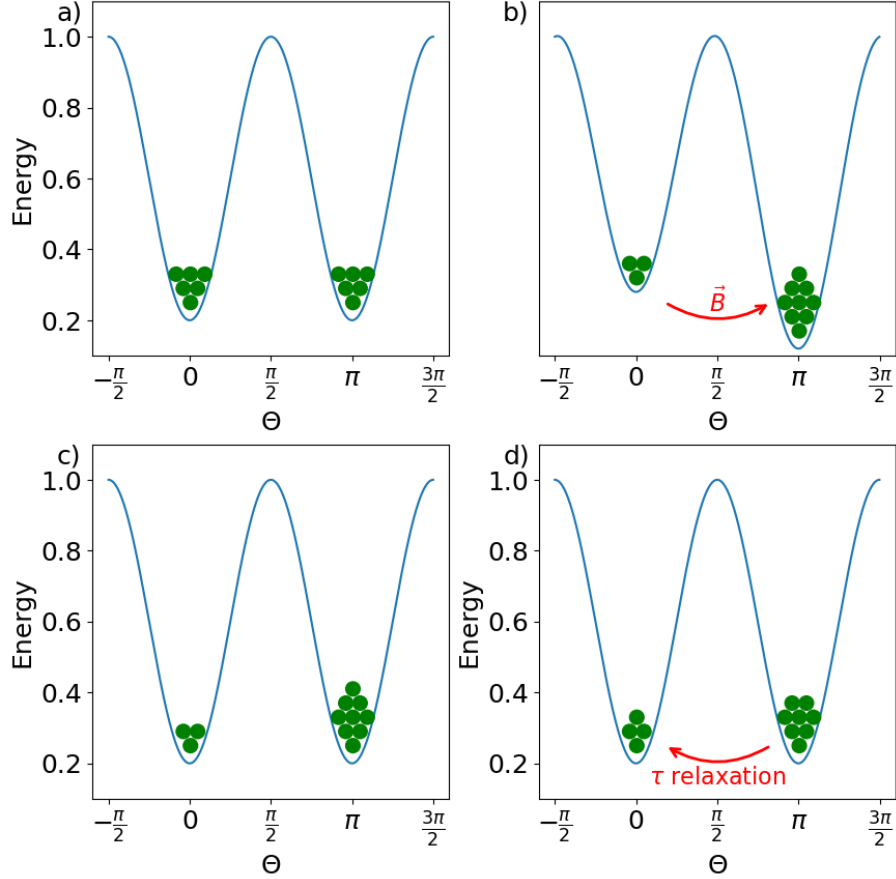


Figure 1.2: **Ising type anisotropy behavior.** Panel a) shows the two degenerate ground states in the absence of a magnetic field with equal population. Panel b) shows the population imbalance induced by a lowering of the second eigenstates energy due to an external magnetic field. Panel c) shows the situation right after deactivation of the magnetic field, the ground state is degenerate, but the population imbalance is preserved. Panel d) shows the relaxation back to equilibrium through relaxation processes.

states can be modeled through the Hamiltonian

$$\hat{H} = \sum_i^{N_s} \hat{H}_Z(i) + \sum_i^{N_s} \hat{H}_{ZF}(i) + \sum_{i \neq j}^{N_s} \hat{H}_{EE}(i, j). \quad (1.2)$$

The first term corresponds to each spin center's interaction with an external magnetic field, namely the Zeeman term, and vanishes in the absence of magnetic fields. The last term includes spin-spin interactions and is only relevant for compounds containing multiple spin centers. The second term, however, relates to the so-called zero-field splitting, and is the crucial part that lifts the degeneracy of the electronic spins and only arises in the presence of ligands/crystal structures. We can expand the zero-field splitting term as an ordered expansion of operators acting on the combined angular momentum operator $\hat{J}$, with the basis states $|J, m_J\rangle$ describing both

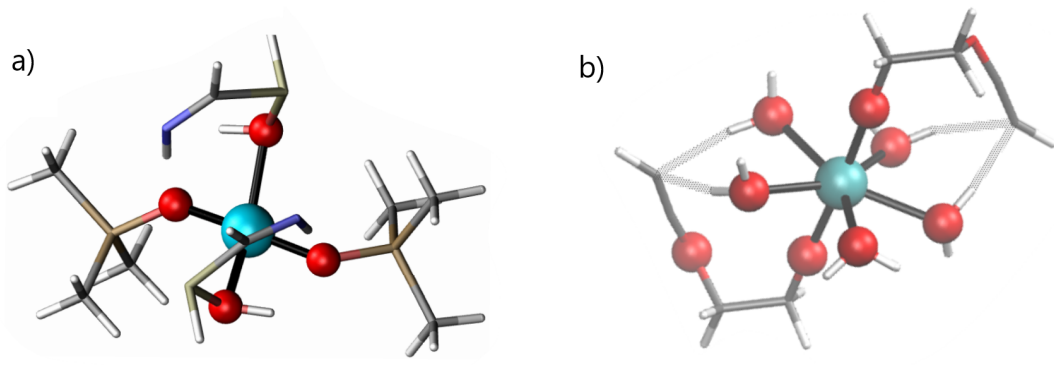


Figure 1.3: **Molecular Magnets.** Examples of Co(II) (left) and Dy(III) (right) molecular magnets discovered in this work.

the total angular momentum and its z alignment:

$$\hat{H} = \sum_i^{N_s} \beta_i \vec{B} \cdot \vec{g} \cdot \vec{J}_i + \sum_i^{N_s} \sum_{l=2,4,6}^{2J_i} \sum_{m=-l}^l B_{l,m} \hat{O}_{l,m}(\vec{J}_i) + \frac{1}{2} \sum_{i \neq j}^{N_s} \vec{J}_i \cdot \vec{J}_{ij} \cdot \vec{J}_j. \quad (1.3)$$

The zero-field term in this expansion is written in terms of the Stevens operators $\hat{O}_{l,m}$ and their corresponding pre-factors $B_{l,m}$, allowing for a fitting of the Hamiltonian to given basis states energies. The Stevens operators here are tesseral operators, closely related to the well-known spherical basis functions $\hat{Y}_{lm}$ under the relation:

$$\begin{aligned} \hat{Y}_{lm} & \quad \text{for } m = 0 \\ \hat{O}_{lm} = \frac{1}{\sqrt{2}}(\hat{Y}_{l,-m} + (-1)^m \hat{Y}_{lm}) &= \sqrt{2}(-1)^m \text{Re}(\hat{Y}_{lm}) \quad \text{for } m > 0 \\ \frac{1}{\sqrt{2}}(\hat{Y}_{l,m} - (-1)^m \hat{Y}_{l,-m}) &= \sqrt{2}(-1)^m \text{Im}(\hat{Y}_{lm}) \quad \text{for } m < 0 \end{aligned} \quad (1.4)$$

This basis is chosen due to the fact that the tesseral functions are purely real and so are their coefficients, allowing for an easier fitting and interpretation of the components. The indices l and m describe their behavior under an arbitrary rotation R , which follows directly from the spherical basis functions' behavior:

$$\hat{Y}_{lm}(R) = \sum_{m'=-l}^l D_{l,m,m'}(R) \hat{Y}_{lm'}, \quad (1.5)$$

with $D_{l,m,m'}(R)$ being the Wigner-D matrix corresponding to the rotation R .

In the expansion, only even-order terms are considered, due to the symmetry of the zero-field splitting, and only terms of order up to $2J$ contribute. This leads to a separation of the $2J + 1$ degenerate orientation of J into $\frac{2J+1}{2}$ double degenerate

states, so-called Kramers doublets.

While this gives a universal model whose parameters can be adjusted to fit experimental data, it does not allow an intuitive view of the anisotropy of a compound and its consequences. Instead, this image can be simplified for systems with quenched orbital angular momentum, which is often the case for transition metals. In this case we can describe molecular spin states by their spin quantum numbers S and m_S . In the case of Co(II), the Stevens operators are now operators acting on $\vec{S}$, and for $S = \frac{3}{2}$ the zero-field splitting Hamiltonian reduces to

$$\sum_{m=-2}^2 B_{l,m} \hat{O}_{l,m}(\hat{S}) = \hat{S}^T \mathbf{D} \hat{S}, \quad (1.6)$$

with $\mathbf{D}$ being a 3x3 matrix diagonalized in its own eigenbasis into D_x, D_y, D_z . Using this Hamiltonian's eigenbasis, we can decompose this relation into

$$D(S_z^2 - \frac{1}{3}S(S+1)) + E(S_x^2 + S_y^2), \quad (1.7)$$

with $D = \frac{3}{2}D_z$ and $E = \frac{1}{2}(D_x - D_y)$ being directly derived from the eigenvalues of $\mathbf{D}$. These terms can be directly calculated from the prefactors of the Stevens operators:

$$D = 3B_{2,2}. \quad (1.8)$$

$$E = B_{2,2}. \quad (1.9)$$

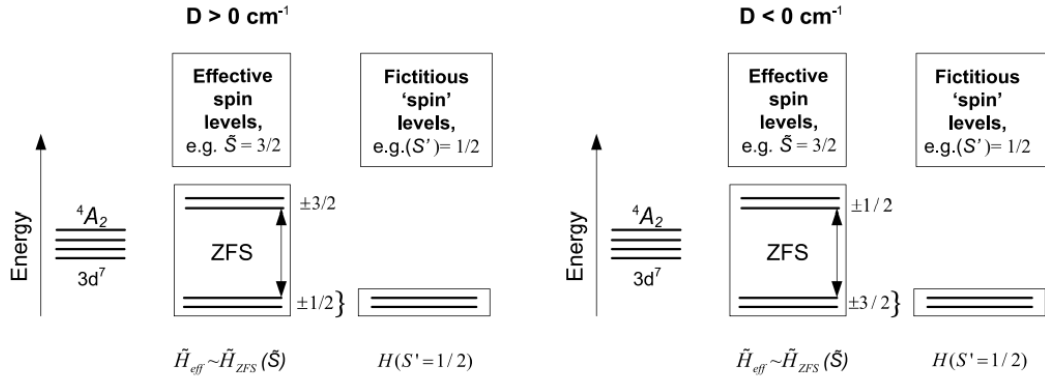


Figure 1.4: **Co(II) configurations.** The actual and effective spin levels of Co(II) for the two cases of $D > 0$ and $D < 0$. Reproduced with permission from [10]. For negative D , the high $|m_S|$ doublet is the groundstate, leading to an easy axis anisotropy capable of reducing spin relaxation, while positive D favors the low $|m_S|$ states and leads to an easy plane anisotropy.

The resulting energy state configuration of Co(II) can be seen in figure 1.4. The zero-field splitting divides the four-fold degenerate ground state into two doublets depending on the magnitude of their spin z-alignment, with the order depending on the sign D : For negative D , the compound exhibits what is known as "easy axis" anisotropy: The lowest eigenstate corresponds to the maximally z-aligned state, and the spin needs to traverse the energetically less favorable x-y plane to relax. For positive D , it exhibits "easy plane" anisotropy, with the lowest eigenstate having minimum magnetic moment in the z-direction, and the spin therefore precessing in the x-y plane easily. The energy gap between the excited state is equal to exactly $2D$, with this energy separation then corresponding to the effective energy barrier $U_{eff} = D(S^2 - \frac{1}{4})$ for the Arrhenius behavior of τ . Therefore, by maximizing the magnitude of negative D , we optimize the exponential behavior of τ [11,12] without having to calculate the full relaxation process, which is a lengthy and expensive calculation [13].

As mentioned, for non-quenched orbital angular momentum, the spin-orbit coupling prevents a treatment in $\vec{S}$ and instead they need to be treated using their total angular momentum operator $\vec{J}$. As the second class of compounds in this work is Dy(III), we want to further display its electronic structure as an example. With a total $J = \frac{15}{2}$, Stevens operators of order $l=2-14$ are required for an exact representation. In this case, the energy 16 degenerate eigenstates are separated by the zero-field splitting into 8 Kramers doublets. One example of this distribution is shown in figure 1.5 for a recently discovered Dy(III) molecular magnet. The Kramers doublets are ordered by their m_J magnitude, and separated by their Kramers gaps ΔE_{ij} between the i -th and j -th levels. The most important one for this work is the gap between the first two levels ΔE_{01} , as it represents the minimum energy gap that needs to be covered to invert the spin orientation and relax to equilibrium, effectively playing a similar role to the factor $2D$ for Co(II) compounds.

1.3 Numerical Methods

Calculating the electronic ground state wavefunction and its energy for a molecular system is a complex task, and large amounts of effort have been invested into finding efficient solutions for this problem. Even under the Born-Oppenheimer approximation, which allows us to remove the nuclear movement by assuming their location to be fixed, the solution to an N -electron Schrödinger equation requires a $3N$ -dimensional wavefunction, with exact solutions being prohibitively expensive. Instead, we rely on density functional theory (DFT): It has been shown that the

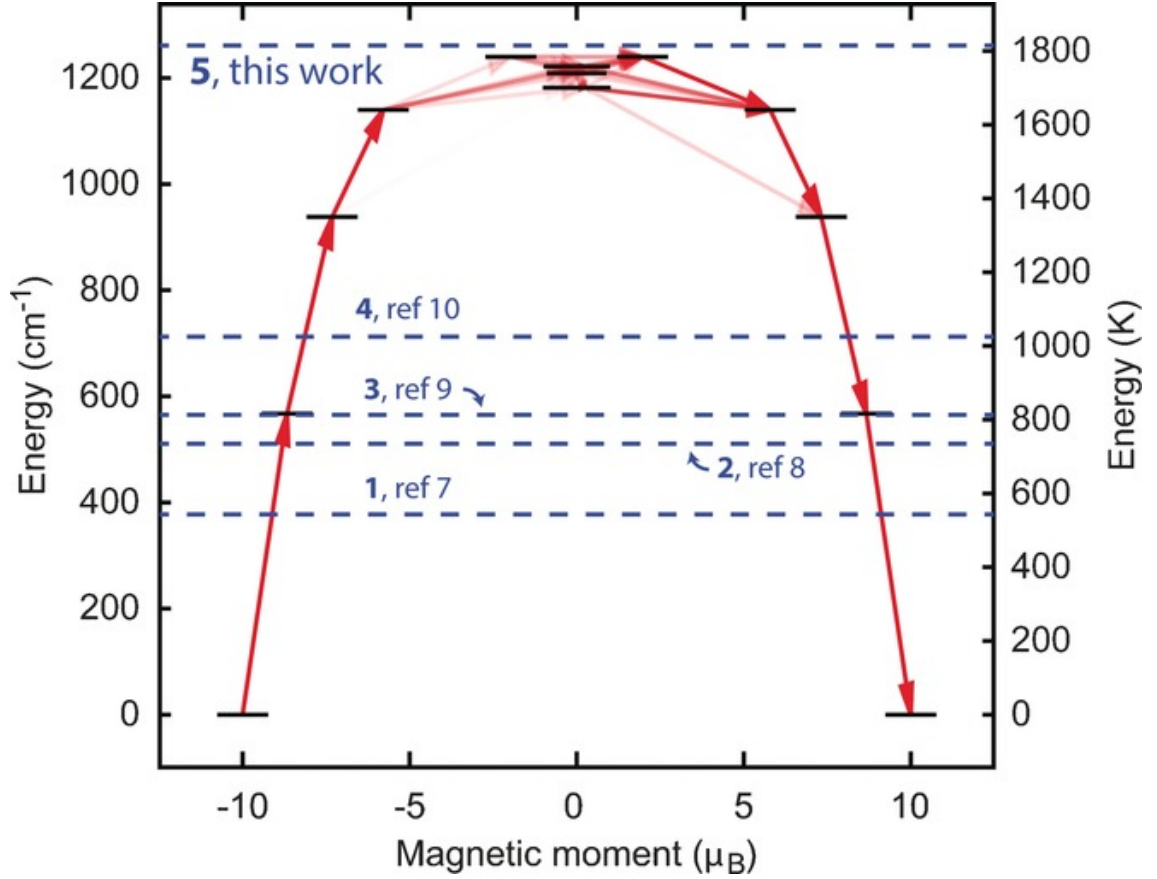


Figure 1.5: **Dy(III) energy levels for a bi-pyramidal compound.** The 16 states are separated into Kramers pairs, with the lower ones sorted by their z -alignment m_J . This creates an energy barrier of at least ΔE_{01} to overcome for a spin to relax from full positive to full negative alignment. Reproduced with permission from [14]

wavefunctions are not required to calculate the groundstate energy, and instead it is possible to work with the single-particle electron density. The Kohn-Sham equations allow for the translation of the full electronic Schrödinger equation into a number of single particle, non-interacting wave functions. The contribution of the interactions enters the calculation in the form of an electronic function E [15]. The goal is then to optimize the single-particle electron density $\rho(\mathbf{r})$ to minimize the energy given by E , which then leads to the optimization criterium

$$\delta[E[\rho(\mathbf{r})] - \mu(\int \rho(\mathbf{r})d\mathbf{r} - N)] = 0. \quad (1.10)$$

While the existence of a universal functional E has been shown, the exact form of it is not determined. As such, a number of different functionals have been developed, all with the goal to use some initial guess $\rho_0(\mathbf{r})$ and optimize it to find an electron density as close as possible to the true ground state, with the computational complexity of this procedure generally scaling as N^3 . The result of this method is the

ground state energy, used for geometry optimization, and an optimized set of states expressed in the atomic orbitals basis supplied.

While this method allows fast calculations of the ground state energy, it comes with two significant issues: First, it lacks the precision to calculate magnetic properties such as the anisotropy, as these are extremely sensitive to small perturbations in the wavefunction. Second, approximate functionals are known to struggle with strongly correlated systems, in this example localized f-like electrons. Additionally, the existence of multiple near-degenerate states near the ground state leads to additional convergence issues.

Instead, we utilize Multi-configurational self-consistent field methods, more specifically, the complete active space self-consistent field (CASSCF) method. This method allows for the computation of highly accurate, multi-reference eigenstates. While a complete optimization of all orbitals would be prohibitively expensive, the active space is defined by a number of electrons, the number of orbitals they are distributed over, and their spin state. The method then optimizes the electronic configuration within this active space based on an initial guess (oftentimes supplied by previous DFT calculations). While this method is much cheaper than a full configuration interaction (full CI) and available for large molecules, it is still too expensive for efficient geometry optimization and is usually used for property extraction.

1.4 Promising molecular systems for high spin lifetimes

Before presenting the methods and results in the later sections, it is important to understand the current state of the field. Co(II) compounds are chosen due to their intuitive description using the $\mathbf{D}$ matrix formalism, introducing a single parameter D that describes their magnetic anisotropy. The ideal theoretical compound would be a linear coordination compound, which have shown effective potentials of 413 cm^{-1} [16] and even 450 cm^{-1} [17]. However, compounds of this class are known to be unstable due to their tendency to accept additional ligands [18]. Instead, we focus on four coordinate compounds, covering a wide range of pseudo-tetrahedral compounds. Experimentally verified compounds contain among others $(\text{HNEt}_3)_2[\text{Co}^{II}(\text{L}^{-2})_2]$ with a magnetic anisotropy D of -115 cm^{-1} and effective energy barrier U_{eff} of 230 cm^{-1} [19]. Three years later the two compounds $(\text{HNEt}_3)_2[\text{Co}(\text{L}^1)_2]\cdot\text{H}_2\text{O}$ (1) and $(\text{Bu}_4\text{N})_2[\text{Co}(\text{L}^2)_2]\cdot\text{H}_2\text{O}$ were experimentally synthesized and reported a magnetic anisotropy D of -144 m^{-1} and -131 cm^{-1} respectively [20]. Meanwhile, a recent study numerically optimized and analyzed a large

number of compounds using DFT and CASSCF and reported theoretical values of D of up to -253 m^{-1} [21], reaching record values far above the experimentally known. While it is important to note that theoretical values can oftentimes overestimate the real results, the number of experimentally verified compounds is in the few dozens, while this study is the largest database of molecular magnets, analyzing over 15,000 compounds numerically, drastically expanding the known chemical space. While individual high interest compounds will still require experimental verification, this increase makes theoretical predictions important to guide experimental work in a more efficient way.

Additionally, we also investigate bi-pyramidal Dy(III) compounds in this work, and wish to create a reference frame for the evaluation of these as well. Due to the unquenched orbital angular momentum, separating the spin Hamiltonian in D and E is not possible, and instead we use ΔE_{01} as our figure of merit. While there is no simple relation between ΔE_{01} and U_{eff} , it still acts as a proxy value for the fast prediction. This property is usually only available from *ab initio* calculations, and therefore we limit ourselves to the theoretical values of ΔE_{01} , which are computed using the CASSCF methods explained in the previous section and the geometry received from experiments. In 2016, a bi-pyramidal Dy(III) compound was synthesized using a single Cl or Br ligand and one hexa-dentate ligand with both oxygen and nitrogen connecting atoms. The *ab initio* calculations revealed a first Kramers gap of between 354 and 448 cm^{-1} , depending on the basis set used, and an experimental effective energy barrier of $708(\text{Cl})$ or $1025(\text{Br}) \text{ cm}^{-1}$ [22]. A more recent study of the bi-pyramidal $[\text{Dy}(\text{L})_2(\text{py})_5]^+$ Dy(III) compound revealed a computed Kramers gap 524 or 550 cm^{-1} [23], depending on the specific axial ligand L , and an experimental energy barrier of 1128 to 1188 cm^{-1} . These compounds both give a guideline on the current state of the art of Dy(III) single-ion magnets, but also the relation between ΔE_{01} and U_{eff} . For non-bipyramidal geometries, dysprosium metallocene compounds show great promise. In Ref. [24], the authors synthesized and analyzed two metallocene compounds, with first Kramers separations computed at 365 and 520 cm^{-1} , leading to an effective barrier measured at 747 and 1410 cm^{-1} .

1.5 Current Challenges and structure of this thesis

After three decades of work, the field of molecular magnetism has come a long way pushing the relaxation rates and maximum temperatures in experiments, however there is still a long way to go. With the rapid improvement in numerical *ab initio* methods and computational resources available, the numerical treatment of

electronic structure problems has been enabled, and together with the analytical development of spin-phonon relaxation methods [11], we can now predict the spin lifetime directly.

However, this full treatment comes at great computational cost, and has only been applied to a handful of compounds. Using the magnetic anisotropy or first Kramers energy as a proxy value speeds up this process, and yet still there is few databases of large size that contain the spin Hamiltonian information, with most publications focusing on less than 10 compounds at a time. The recent rise in machine learning methods and other novel approaches however has lead to a speed-up in many fields, yet has found very limited application in molecular magnets [25, 26]. We want to group the challenges and inherently also the possible solutions into two groups:

- **Acceleration of computation.** The price of individual treatment of compounds is too high to create databases of the size usually known in chemistry, such as the Cambridge structural database [27] or QM9 [28], which range in millions of compounds and oftentimes include experimental results. Machine learning methods have successfully been used to predict molecular properties and show great promise in either predicting the proxy quantities directly using neural networks [25] or using reinforcement learning to guide the calculations in a more efficient way [26]. The challenges here lie in the absence of large homogeneous training datasets, the high complexity of the spin Hamiltonian and very sensitive dependency on the final geometry.
- **Generation of new compounds.** Even if computations can be speed up, the question which compounds to investigate is complex on its own. Even within one chosen metal ion and geometry, the space spanned by potential ligands is practically infinite, and discovery of new compounds is often based on intuition or prior experience of the acting researcher. A more structured approach, free of biases and capable of learning from previous results instead of sampling the space randomly, would allow a drastic speedup in analysis. Different kinds of generative machine learning methods such as variational auto encoders [29, 30] or generative adversarial networks [31–33] have been used in the past, as well as genetic algorithms [34–39]. The challenges here come from the vastness of potential chemical space, as well as a careful balance between control and randomness in many algorithms suitable for this solution.

In this thesis, we are taking on both challenges on their own, with different techniques.

In **chapter 2**, we propose genetic algorithms as a potential approach to the efficient traversal of the chemical space. This kind of heuristic algorithm iteratively creates generations of new compounds, each one based on the most suitable ones

found before. We implement and analyze the performance of two different versions of genetic algorithms: The first, labeled static ligand genetic algorithm, creates new compounds with a specific metal ion and geometry based on a list of ligands supplied, combining them in optimal ways. The algorithm is benchmarked for its speedup against an already created dataset of tetrahedral Co(II) compounds, and later on run on a larger list of ligands extracted from the Cambridge Structural Database. The second kind is capable of creating its own ligands, therefore labeled dynamic ligand genetic algorithm, and creates compounds with ligands that aren't found in any database on the fly. It is tested on both tetrahedral Co(II) and bi-pyramidal Dy(III) compounds, with the resulting compounds being competitive with the current state of the art.

In **chapter 3**, we approach the challenge of speeding up the individual calculation of spin Hamiltonians using graph neural networks. For this purpose, 9889 Dy(III) compounds have been created, optimized, and their spin Hamiltonian extracted using DFT and CASSCF. We implement two different graph neural networks and one deep neural network for comparison, each capable of predicting all seven Kramers energies directly, and provide a detailed analysis of their performance. The prediction of the spin Hamiltonian itself provides significantly more flexible applications in the future, such as the prediction of derivatives on spatial distortions for spin relaxation methods, the g-factors describing the compound's behavior in a magnetic field, or the EPR-spectrum. However, contrary to the scalar energy gaps, the spin Hamiltonian is subject to change under rotation of the molecule, and as such requires a graph neural network capable of reproducing this behavior in its output, namely behaving equivariant under rotation. Using additional equivariant features inspired by [25], we present a fourth model capable of using these features to predict the entire spin Hamiltonian's tesseral decomposition into Stevens operator coefficients.

2 Results - Genetic Algorithm

The work in this chapter has been published in [40] and was a cooperative undertaking. Lion Frangoulis designed and implemented the genetic algorithm itself, executed the *ab initio* calculations and performed all simulated benchmarkings. The random forest classifier was implemented and trained by Dr. Zahra Khatibi.

2.1 Methodology

2.1.1 SMILES and SELFIES

For the later application of genetic algorithms to chemical problems, an accurate and computer-readable way of describing molecules is required. The Simplified Molecular Input Line Entry System (SMILES) has been developed specifically for this reason [41], with each SMILES string fully describing one molecule’s chemical composition and bond topology. They are capable of describing arbitrary elements, including the number of connected hydrogen atoms and assumed charge state, as well as single to quadruple bonds, aromatic bonds and, with later extensions, also dative bonds. They can also include stereo-chemical information, both for single bonds adjacent to double bonds, and tetrahedral carbon stereo-isomers. While each SMILES string uniquely maps to a specific molecule, the inverse is not true since the SMILES string can begin at any atom in the molecule and can traverse the graph in any order. This later lead to the addition of canonical SMILES [42], which are unique for each molecules.

Rings and branches in a molecule are described by numbers and parentheses at the beginning and end of either, which leads to the main disadvantage of SMILES for generative chemical methods: These non local parts of the string lead to invalid SMILES strings if separated into two SMILES strings or modified without care, which lead to the development of Self-Referencing Embedded Strings (SELFIES) [43]. This new representation based on finite automata uses only local operators to describe branches and rings, and automatically enforces valence rules by dropping atoms that would violate it. They have been used in generative AI applications for

chemistry with a 100% valid string reconstruction [43], and extended to include the usage of prior defined fragments in the form of so called group SELFIES [44]. While to our knowledge there is no canonical SELFIES format making them unique for each molecule, the python package selfies [43] allows for the quick translation between SELFIES and SMILES, enabling the comparison between two SELFIES.

2.1.2 Genetic algorithms

Genetic algorithms are a class of heuristic algorithms first developed in 1992 [45] by JH Holland. The underlying idea is to mimic a natural selection process that is assumed to lead to highly efficient adaptations to a given problem. The application of this class of algorithm is incredibly wide and includes among others economic optimization problems [46], the optimization of machine learning hyper parameters [47], physics and engineering design [48,49] and also various different chemistry applications. This method is by design highly general, and can be applied to a number of different problems with very little requirements on the problem in question. However, on the downside its heuristic nature makes a guarantee of success impossible [50]. In the field of chemistry, they have found broad application, especially in the design of either organic and biological compounds [37–39] or solid-state inorganic crystals [51–54]. Coordination compounds are highly complex and offer a large degree of freedom for design of novel compounds, and as such have received less attention so far, mostly limited to structural optimization of catalysts [34–36,55–57].

The requirements come naturally from the explanation of a general GA workflow shown in figure 2.1. The initial step is to form an initial set, furthermore related to the initial population, of solutions to the problem at hand. The individual solutions, also known as specimen, can be chosen randomly or from prior knowledge, and need to be described by a genome, meaning a description build up from small building blocks, referred to as genes, that fully describe a unique compound and can be split and modified to create new compounds. The shape of this genome can vary drastically for different applications, and can be made up of among other binary encodings, value encodings, set encodings or string encodings [50,58]. It is important for this encoding to be a two way function: The genome needs to contain enough information to fully describe a solution to the problem at hand, while we also need to be able to go from a certain solution to a genome describing it.

The second step in the workflow is the evaluation of fitness: Any genetic algorithm requires a so called fitness function, which can be defined as a one way function that takes the genome of a specimen as an input, and returns a scalar value of suitability to the problem at hand. These fitness functions can range from a simple single value [59], to a weighted average of multiple properties, and can even change during

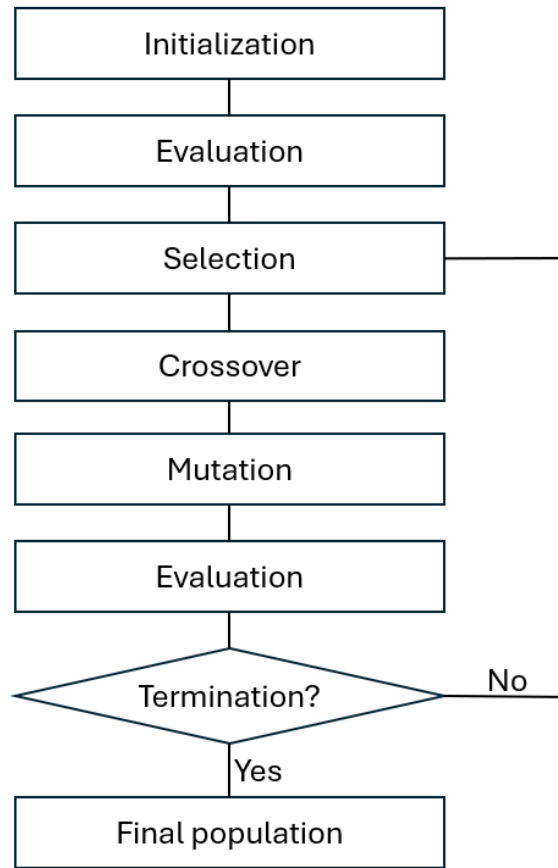


Figure 2.1: **GA workflow.** The workflow of a genetic algorithm begins by initializing a random population and evaluating their fitness function. The second part is the iterative part, where through a combination of selection, crossover and mutation new generations are created and subsequently evaluated, until the termination criteria is fulfilled. Once this criteria is fulfilled, an entire population of highly suitable solutions is returned.

the optimization run [60]. The one way nature of this function is intrinsic from any optimization problems: Many possible solutions are allowed to have the same suitability, and having an inversion of this function would defeat the purpose of the algorithm as one could simply pick a desired target value and return a solution that delivers it.

The third part contains a termination criteria, at which the iterative part of the algorithm stops: oftentimes this can simply be reaching a certain threshold in the fitness function, however it can also prove wise to apply secondary conditions such as a maximum amount of resources used, or a minimum amount of diversity left in the population to allow for future discovery of novel solutions [61]. If one of the requirements is reached, the algorithm stops with the current population being the final set of solutions, if not the optimization loop begins/continues: A number of highly suitable specimen is selected to continue to the next generation of solutions based on their fitness. Various different selection types have been developed, with

the most common ones being a selection of the top compounds (elites) [58], a tournament selections where subsets of specimen are created and the winner of each subsets continues to the next generation [62], essentially applying a higher probability to higher ranking solutions, and a value-based selection which picks specimen based on a probability function according to their fitness values [63]. This creates a new, smaller population of specimen that act as parents to create new ones.

To fill this population back to the original number, a series of operations are applied on randomly selected pair of parents: the first step, crossover, takes the genome of two parents, splits each one according to some rule, and recombines the fragments into one or more new genomes, creating a number of offspring specimen that ideally have properties similar to both their parents. Intuitively, this can be understood as sampling the solution space in the mid point between to solutions known to be of high fitness. In a second step each of these offsprings has a chance to undergo mutation: A single gene of the genome is removed, added or replaced with a different one, allowing the algorithm to sample the solution space nearby to increase diversity.

Subsequently, the new compounds are evaluated using the fitness functions, and the first iteration of the optimization is complete. If this new population composed of both parents and offspring fulfills the termination criteria the algorithm ends, if not this cycle is repeated until it does. The final population is then assumed to contain a number of highly suitable solutions to the problem at hand.

The hard requirements for a GA to be applicable to a given problem can therefore be summed up as having:

1. An encoding, a two way function that connects each solution to a genome that contains all the information required to uniquely describe a solution.
2. A fitness function, a one way function that maps each genome to a scalar value that describes its suitability as a solution to the problem at hand.
3. A defined set of crossover and mutation operators that maps parent genomes to valid offspring genomes. While these are the only hard requirements, they do not guarantee the GA to be an efficient way of solving the problem. While it is impossible to predict the convergence of a GA [50], guidelines on how to implement it well have been established [62]. Key results are the closeness of interacting genes on the genome, increasing the chances that they are inherited together, and a high mutation rate to keep diversity high.

The main disadvantages of genetic algorithms are their inherent risk of being heuristic algorithms, and the fact that they tend to struggle with problems containing a single high suitability solutions, since no systematic optimization is performed. However, their independence of a training set enables them to be

applied where computational cost makes the generation of large datasets unviable, adding to their high versatility in applications. Additionally, other than classical optimization schemes, GA are designed to deliver a large population of highly suitable specimen instead of a single one, making them especially suited for problems where variety is required, or experimental implementation of each solution is not guaranteed, such as theoretical chemical problems.

2.1.3 Ligand extraction

The static GA requires a list of ligands to select from to create new compounds. Since the goal is to cover a chemical space as diverse as possible, this requirement translates into the ligand selection as well, and as such a very diverse database was required. The Cambridge structural database (CSD) contains a large number of different crystal structures containing metal coordination compounds that have all been experimentally verified, and such are known to exist. This has the advantage, while not guaranteeing the synthesizability of our compounds, that the ligands themselves exist in some coordination compound. We extract all crystals containing a compound with only a single metal and discard all compounds that are flagged to contain multiple possible confirmations or X-ray errors. Afterwards we split them into their individual molecules of the unit cell using MolForge. In a second step, the metal ion is removed from those molecules, leading to the individual ligands being separable and used as the first stage of the ligand list. This list is then filtered to only contain monodentate, closed shell (even number of electrons is assumed to guarantee closed shell states due to the absence of metallic elements) ligands with less than 26 atoms and no connecting atom heavier than Bromide. Additionally, we require a negatively or neutral-charged connecting atom to facilitate the dative metal-ligand bond.

To create a sampling of this list that is as chemically diverse as possible, we calculate the bispectrum components ($J_{max} = 8$, $r_{cut} = 4$) [64] of each ligand based on the connecting atom. This results in a 55-dimensional vector for each ligand. To reduce this complexity a principal component analysis over this entire set was performed. On this two dimensional PCA space, 678 ligands with maximum distance to each other were selected and verified by hand, to ensure no artifacts remain, creating a dataset of maximum chemical diversity within the space created by the database.

2.1.4 Compound creation

In this work two kinds of compounds were investigated: bipyramidal Dy(III) compounds with water as the planar ligands and tetrahedral Co(II) compounds. In order to perform the optimization and extract magnetic properties, an initial guess for the complex is needed. We use the python software MolSimplify [65] to assemble the compounds. For Dy(III) compounds, we first assemble the core surrounded by water, and optimize the bond distances while keeping bond angles static using DFT methods. This pre-optimized core can then be used in a second stage to assemble the two axial ligands. For Co(II) however, all four ligands are assembled one after another without optimization steps in between. The semi-dynamic genetic algorithm on the other hand creates its own ligands based on a SELFIES string, which is translated into a SMILES string using the python package selfies [43]. The initial guess of the individual ligand geometry is then either created using rdkit [66] (in the case of Co(II) GA), or directly created by MolSimplify from the SMILES string (in the case of Dy(III) GA). In all four cases, once the initial guess of the coordinates is created, this geometry is optimized using ORCA 5 to achieve a realistic conformation and extract its magnetic properties.

2.1.5 Electronic structure simulations.

We use the quantum chemistry software ORCA 5 [67] to perform both DFT and state-averaged CASSCF calculations. Scalar relativistic effects are treated using the Douglas–Kroll–Hess (DKH) method, with picture-change effects included up to second order to account for DKH corrections in the spin–orbit coupling operator. The DKH-def2-TZVPP basis set is employed for all atoms, except for elements heavier than Kr, for which the SARC-DKH-TZVPP basis set is used. Energies and forces are converged to 10^{-9} a.u. and 10^{-6} a.u., respectively. The active space for the CASSCF wavefunction consists of seven electrons in the five 3d cobalt orbitals. This active space is automatically selected based on the Löwdin orbital composition from DFT calculations, evaluating the contribution of Co 3d atomic orbitals to each Molecular Orbital (MO). If any of the five highest occupied MOs contain less than 30% Co 3d-character, they are replaced with the highest-energy occupied orbitals that exhibit more than 30% Co 3d-character. Magnetic anisotropy is extracted from the CASSCF Hamiltonian using a mean-field spin–orbit coupling operator within the framework of Quasi-Degenerate Perturbation Theory (QDPT). The state-averaged CASSCF procedure includes 10 quartet and 40 doublet states. Compounds that fail to converge in either the geometry optimization or CASSCF calculation are assigned a default value of $D = 0 \text{ cm}^{-1}$ and automatically discarded by the genetic algorithm.

For Dysprosium, D is not well defined, and instead we calculate the Kramers gaps ΔE_{0j} and the spin-orbit Hamiltonian. To achieve numerical convergence, a more intricate optimization scheme has been used, designed by Dr. Lorenzo Marianno. DFT methods are known to fail to converge on Lanthanide elements, and instead the Dysprosium core is replaced with an Y(III) core. Apart from the Dysprosium f-orbitals, which have negligible impact on the ligand bonds, they behave similarly and are expected to converge to a similar geometry. We optimized the compounds in two stages: First by fixing the water molecules’ oxygen atom in a perfect plane with variable distances, and only relax those in a second step. This allows to extract magnetic properties independent on the water molecules behavior for other studies, and improves convergence rate in the optimization.

Afterwards, we replace the Y(III) core again with Dysprosium. For the extraction of magnetic properties, CASSCF is used due to higher precision, which can deal well with Lanthanides. For the CASSCF calculation, the orbital of the Dysprosium core is evaluated individually in a combination of DFT and CASSCF calculation, as well as the DFT orbitals of the ligands with the core removed, and combined as the initial guess for the CASSCF calculation. The active space contains 9 electrons in 7 orbitals, with 21 roots and a final multiplicity of 6. Due to the large separation of f orbitals and the rest, the majority of calculations remain within this active space, allowing the cutoff point to be set to 60% f-orbital behavior. The CASSCF calculation returns both the Kramers energies, as well as the full spin Hamiltonian.

2.2 Results

After describing the general workflow of a genetic algorithm, we now want to show two different implementations of such an algorithm specifically for the discovery of novel single ion magnets. The two algorithms follow different methods to source their ligands, and as such have different targets: the first one, dubbed static genetic algorithm, requires a list of ligands whose charge state and connecting atom are known, and is capable of scanning the chemical space spanned by combinations of these ligands around a central metal ion. The second one, dubbed semi-dynamic algorithm, instead is based on SELFIES: by fixing one ligand and describing the other one as a SELFIES string, the modification of individual atoms or bonds within the second ligand are possible, leading to a virtually infinite chemical space controlled by the boundaries set to the SELFIES themselves. All algorithms in this work used a mutation rate of 1, meaning they apply the mutation operation on every offspring created. The population size is fixed to 128 compounds, allowing for a good trade off between parallelization and efficiency, and the parent selection is a simple elitism scheme, retaining the best 64 compounds while discarding the other

half. Due to this scheme, if a compound is discarded once, there is no way for it to survive the selection later on if rediscovered, and as such an additional step is implemented after the mutation stage: The new compound is compared against a list of previously discovered compounds and discarded if already known, to avoid wasting computation times.

2.2.1 Static genetic algorithm

The static genetic algorithm’s encoding is based on a pre-compiled list of ligands, and the genome consists of a list of numbers corresponding to the ligands used to create the compound. As such, this algorithm is limited in the chemical space searched, but also more controllable in direction. The crossover operation is performed by using a number of genes (ligand identifiers) from one parent, and the rest of the ligands from the other parent, which combined give a new genome with the same number of genes (ligands). It is important to note that the numbering of ligands is chosen arbitrarily, and relative positions within this numbering does not relate to chemical similarity. The mutation is performed by taking a single ligand and switching it for a random different one.

The compounds investigated in this section are all tetrahedral Co(II) compounds, chosen due to their potential of high magnetic anisotropy D while remaining relatively small in ligand number. However, due to the complexity of synthesizing a tetrahedral compound with four different ligands, we limit ourselves to two different ligands, each appearing twice, leading to a genome length of only two genes. However, the developed algorithm is written fully adaptable to arbitrary numbers of ligands. Tetrahedral Symmetry in the compound leads to symmetry between the two genomes, and as such when comparing a new compound against the known compounds, both orders need to be verified.

The fitness function in this case is the magnetic anisotropy D , with the goal of minimizing it, leading to compounds with high magnitude, negative anisotropy D , which corresponds to strong axial anisotropy. For this purposes, starting from the genome containing the ligand IDs, a compound needs to be created by assembling the ligands around the central Co(II) ion, optimizing the compound using DFT and calculating its magnetic properties, including D , using CASSCF. The details of this are layed out in section 2.1.4, however it is important to note here that not all calculations converge, and as such not all compounds return a value for D . For this case, there is a placeholder value of 200 cm^{-1} applied to those, essentially marking them as unsuitable to the selection scheme.

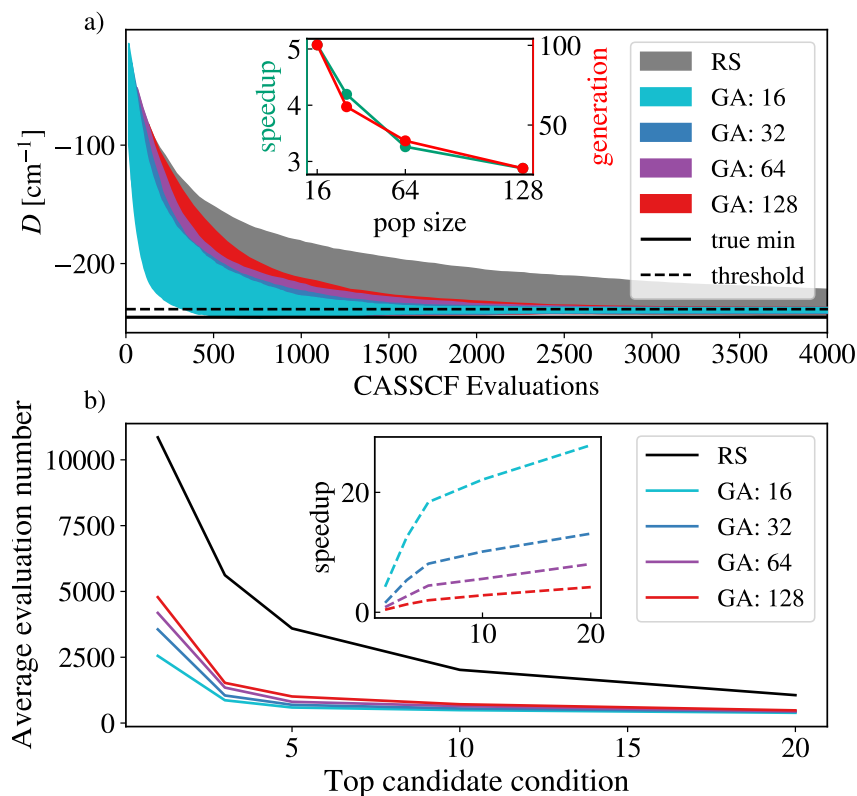


Figure 2.2: **Genetic algorithm benchmark.** (a) Comparison of minimum D growth for the random sampling approach and GA with different population sizes, performed on the COMPASS test set and averaged over 1000 runs. The termination criteria is set to find one of the top three compounds in the set. The inset shows the speedup and average number of generations needed for termination of run for different population sizes. (b) Comparison of compound evaluations needed for random sampling and GA with different population sizes, versus the termination criteria—finding one of the top 1,3,5,10 or 20 compounds within the COMPASS dataset. The inset shows the speedup that GA with different population sizes enables compared to the random sampling.

COMPASS Benchmarking

To verify the functionality of the algorithm and get a quantification of the speedup provided, it is important to first benchmark it against a simple random search. Due to the fixed list of ligands, the chemical space is well defined, and therefore a dataset can be used to perform this benchmark. The COMPASS [21] dataset extracted 208 ligands for the Cambridge Crystallographic Database (CSD), leading to a total number of 21,736 possible tetrahedral Co(II) compounds created from them. Out of those, 15,547 compounds were successfully optimized and their magnetic properties extracted, creating a database we can use to benchmark the static genetic algorithm against.

We can run a large number of genetic algorithm runs, each trying to optimize D within this list of 208 ligands, without having to run any calculations and instead looking up the fitness values in the database and marking unavailable compounds as unsuitable, allowing us to perform a statistical comparison between a random search (RS) and genetic algorithm search within this chemical space. The termination criteria is set to end at either 500 generations or when finding one of the top N compounds, whichever happens first. We evaluate the computational cost using the number of compounds queried, as this usually would each require an expensive calculation, assuming equal cost for every compound. We additionally define the speed up the ratio of GA and RS evaluations required:

$$S_{GA} = \frac{\# \text{ of unique compounds queried by GA}}{\# \text{ of compounds queried by the random sampling}}. \quad (2.1)$$

The results are shown in figure 2.2 for a termination criteria of finding one of the top three compounds. The top panel shows the average growth and its standard deviation of genetic algorithm runs with various different population sizes in color, the random search results as a comparison in gray, and the true minimum and threshold in black. It can be seen that for all investigated population sizes, there is a clear speedup using GA over RS in this theoretical framework, with higher population sizes leading to less efficiency. The inset shows the actual speed in green and the number of generations required in the GA for different population sizes with the same termination criteria. The speedup here relates to the saving of computational time, while the number of generations correlates to the real time it would take to run the GA, since each generation can only be run once the previous is finished, showing a trade-off of the two for different population sizes. This trade-off is due to the fact that small population sizes carry a smaller number of medium quality specimen with them, leading to a more efficient calculation. Large population sizes however carry a larger amount of undesirable solutions, diluting the pool of parents and leading to reduced efficiency, however also allowing for more calculations to be performed in parallel per generation, with the limit of extreme population sizes essentially turning into a random search, having no speedup but maximum parallelization. In the bottom half we instead show the average number of evaluations required to find one of the N -th best compounds for different population sizes and a random search. While the genetic algorithm clearly becomes faster for less harsh termination criteria, so does the random search. We therefore plot the speedup S_{GA} in the inset, showing that it increased with more relaxed termination criteria for all different population sizes. This behavior is expected due to the strength of genetic algorithms of looking for problems with a number of good solutions instead of a single optimal solution.

On the fly Static Genetic Algorithm

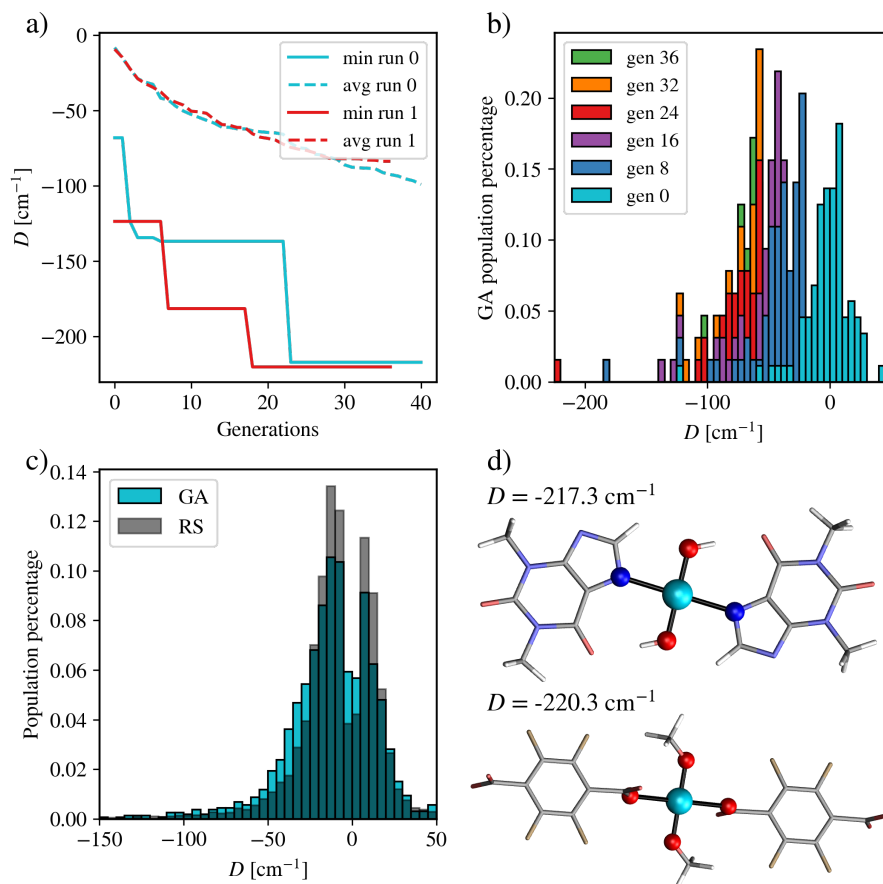


Figure 2.3: **Results for the GA with static encoding.** a) Minimum and average anisotropy growth of two separate static GA runs. The minimum curves of both runs have a very jagged growth, as expected, while the average continuously goes down. However, both reach the $< -200 \text{ cm}^{-1}$ regime. (b) Temporal evolution of GA-produced compounds and their anisotropy, with the new population discovered at regular intervals of generations shown in different colors. The GA population at each generation roughly follows a skewed normal distribution with later generation concentrating and converging towards higher D values through a steady progression. (c) Cumulative distribution of compounds to compare the GA and random sampling performances, with the GA showing a higher relative population in the medium negative regime. (d) Top performing compounds and their magnetic anisotropy. None of the top compounds preserved the original tetrahedral geometry, however, all show reasonable optimized structures. (Cyan: Co, Red: O, Grey: C, White: H, Beige: F, Blue: N.)

After demonstrating the viability of this simple implementation of a genetic algorithm, it is utilized to create novel Co(II) compounds. For this purpose, crystals containing metal coordination compounds from the CSD are extracted and investigated for their ligands, with the details shown in section 2.1.3. This resulted in a list of 678 monodentate ligands, creating a chemical space of up to 229,503 compounds, significantly outstretching than any high throughput study on magnetic

compound performed to our knowledge. Since there is no database available at this scale, calculations are performed on the fly for each generation. The results of two independent runs with random initial populations are shown in figure 2.3. The first panel shows the minimum and average anisotropy of each generation after the parent selection. This is done to ignore the impact of unconverged compounds on the average, with the minimum being unaffected. The behavior is as expected, with the average growing continuously, while the minimum has rare breakthrough event at which a new best compound is found. The termination criteria was set to either 50 generations or a threshold of -250 cm^{-1} , neither of which were triggered by the time the runs were manually stopped due to no more growth being observed. As the GA is designed to find a large number of decent solutions, the average growing is important, and is further analyzed in figure 2.3 b), where we show a histogram of the new compounds discovered at regular intervals of generations. While the first generation is completely random and approximately normal distributed around 0, the later generations keep moving towards negative values and begin to populate the areas. The cumulative distribution is shown in panel c), which shows the GA compared to the overall random search results (normalised for total compound number in each set). It can be seen that the GA populates the left flank significantly more than the RS, while still struggling to reach the far end of negative anisotropy values, showing a less significant advantage compared to the per generation view. This is due to the large number of unsuitable compounds found in the initial generations, that are immediately discarded and therefore don't contribute to panel b), but still add computational cost and are included in panel c). Panel d) shows the top compound of each of the two runs with their respective magnetic anisotropy. It is easy to see that the tetrahedral geometry has been completely replaced by a square planar one, maximizing magnetic anisotropy.

2.2.2 Semi-dynamic genetic algorithm

In a second step we further refine the algorithm to use a dynamic encoding of ligands using a SELFIES based encoding. Differently from the static encoding already discussed, genes do not correspond to the entire ligands but to fractions of them, as identified by the tokens of the strings of text used to represent them. In this implementation, ligands do not come from a pre-existing list but are generated by the algorithm in real time, leading to the exploration of a much broader, effectively infinite, chemical space.

While this algorithm is significantly more powerful, both in the size of the chemical space but also in its capability of making minor adjustments to the ligands and instead of swapping them out completely, the technical implementation is more

complex. Especially due to the complex relation between the two ligands creating a crystal field that maximizes or minimizes D , the combination of ligands is what really matters and not just the individual ones. As such, we fix one of the two ligands in the beginning of the run, while allowing the second one to be modified at the atomic level, hence the naming semi-dynamic genetic algorithm. However, we want to clarify that the software developed is completely adaptable to arbitrary numbers of both fixed and dynamic ligands.

The encoding is fundamentally different from the static algorithm one, with the fixed ligands SMILES string making up the first part of the genome, and the dynamic ligand’s SELFIES string making up the rest, with each token corresponding to one gene. One particular attribute is that each SELFIES string always begins with the connecting atom, which is limited to be either oxygen or nitrogen in a neutral or single negative charge state. As the first ligand is static, the semi dynamic GA really tries to optimize the second ligand to synergize as good as possible with the first to create a ligand field leading to high axial anisotropy. The different encoding also gives rise to new crossover and mutation operations: in the crossover stage, the first ligand is retained as all specimen within one GA run share it. The second ligand of both parent specimen however is split into individual SELFIES tokens, and split into two parts making up between 20 and 80% of the total token numbers. Afterwards, the first part of the first parent is combined with the second part of the second parent and vice versa, creating two offspring specimen per parent pair. The nature of SELFIES string meanwhile ensures a valid string to be generated in both cases, with the connecting atom remaining in the first token. In the mutation stage, a random token is either removed or replaced, or a new token added in a random location, with the exception being the very first token, which is always held constant to ensure the nature of the connecting atom. The fitness function in this case depends on the nature of the compound the GA is applied to.

With the chemical space covered by SELFIES being essentially endless, boundaries need to be put in place to steer the genetic algorithm in a sensible direction. The number of tokens in the SELFIES string is limited to between 5 and 15 tokens. While the selection of elements is random, it is unwise to keep the ratio of them to be equal, as certain elements facilitate long chains much better than others. As such, the chance of a single bond carbon atom token appearing is increased 15 times compared to others, as well as five times for double bond carbon tokens. This increases the chance of a robust organic skeleton, instead of just pairs of first or seventh group elements. The connecting atom element of the dynamic ligand is fixed to be either nitrogen or oxygen.

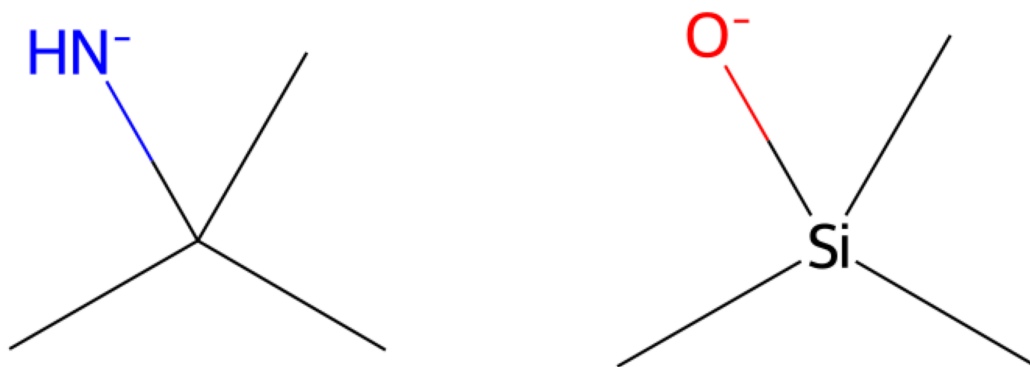


Figure 2.4: **Chosen static ligands.** The molecular graph for the two chosen ligands used in the semi-dynamic GA run for tetrahedral Co(II) compounds. The ligands were selected by identifying the most commonly occurring ligands in the top 100 compounds of [21], after removing ligands with connecting atoms heavier than Ar.

Co(II) results

We first apply this new algorithm to the same class of compounds as before, tetrahedral Co(II) coordination complexes with two ligands appearing twice each. In order to select the fixed ligands, all ligands in the COMPASS dataset were analyzed by their probability of appearing in the top 50 compounds, and the two best performing ones that do not contain metal heavier than Krypton were selected, with their molecular graph shown in figure 2.4. We perform one GA run with each of those ligands, allowing the other ligand to be optimized within the SELFIES constraints. Additionally, a random forest classifier (RFC) model has been created by Dr. Zahra Khatibi. This model is operating based on the SMILES string created from the SELFIES string of the dynamic ligand, as well as the known SMILES string of the fixed one. The combined string is then transformed into binary vector representation using molecular fingerprints, using the Extended Connectivity Fingerprint (ECFP) family [68]. Based on these, the model classifies a potential offspring into two categories, above and below -10 cm^{-1} , with the model previously being trained on the real anisotropy values. Only compounds of category 1 (below -10 cm^{-1}) are used for the next generation, while category 0 ones are discarded, leading to an increase of speedup. Due to the requirement of a large dataset for random forest training, the model was only available for Co(III) compounds using the COMPASS [21] dataset, and was not used for Dy(III) compounds.

While a lower threshold of separation between the categories would have lead to an improved speedup, the imbalance in datapoints in each category grows rapidly

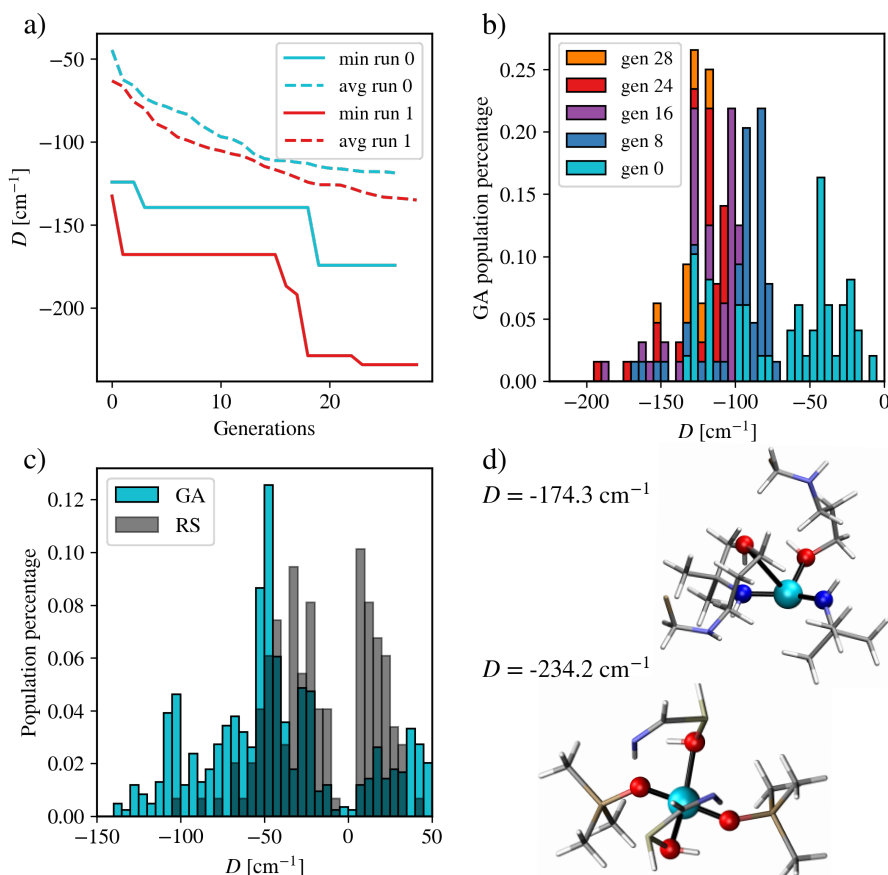


Figure 2.5: **Results for the GA with dynamic encoding.** (a) Comparison of the minimum and average anisotropy for two dynamic GA runs. The dynamic GA presents a faster progress and convergence starting from an informed guess using two top-performing ligands as the fixed genes and using the RFC pre-screening. (b) Temporal evolution of GA-produced compounds and their anisotropy, with the new population discovered at regular intervals of generations shown in different colors. The GA population is more dispersed compared to static GA with the majority of compounds reaching $< -100 \text{ cm}^{-1}$ as early as 16th generation and a noticeable portion of the compounds accumulating in the $[-200, -100] \text{ cm}^{-1}$ region. (c) Cumulative distribution of compounds generated by the GA (blue) compared to all COMPASS entries that include the fixed ligand. (d) Top compounds and their anisotropy D , again with strong divergence from the original tetrahedral coordination. (Cyan: Co, Red: O, Blue: Nitrogen, Grey: C, White: H, Beige: Si.)

and reduces precision and accuracy. Unfortunately, the exact speedup can not be measured, as the GA is exploring the chemical space statistically, and as such we can not compare one run with and one without the RF model.

For comparability we perform the same analysis as for the static algorithm in figure 2.5, with the first panel showing the average and minimum D growth for both compounds. Compared to the static run, the minimum behavior is very similar for the second ligand shown in 2.4, the breakthrough points are again between 15-20

generations, with the first ligand struggling to breach the below -200 cm^{-1} regime. However, the average reaches significantly better values at a faster pace, showing the advantage of this encoding. Once it finds a good ligand partner, it can produce similar ligands and further optimize the compound. The GA run containing the second ligand is used for the other three panels, with figure 2.5b showing the per generation distribution of new compounds, confirming this interpretation: There is a strong growth in the overall distribution of D values towards large negative values very rapidly with the passing of generations, with compounds after the 16th generation all being within the below -100 cm^{-1} regime. Next, we take a look at the distribution of all the resulting compounds in Figure 2.5c, normalized with the full population and compared to the subset of all COMMPASS compounds containing the second selected ligand in gray. It is clearly visible that the combination of pre-selecting one of the two ligands and the usage of the RFC model predictions leads to a shift to the left in early generations, with the GA leading to significant growth in the $< -100\text{ cm}^{-1}$ regime, marking the advantage of this approach over brute force visibly. The GA efficiently populates the low anisotropy regime with significantly higher ratios. Finally, similarly to what observed in the static run, the GA naturally identifies ligands able to stabilize a distorted square-planar or see-saw structure (see Figure 2.5d), which is now known to lead to an unquenched angular momentum and record-large negative D values.

Dy(III) results

While the previous results demonstrate the potential of the genetic algorithm to optimize Co(II) compounds, they do not support the claim of generality. To show the versatility, we apply it to a different kind of compound: Pentagonal Bipyramidal Dy(III) complexes. This class of compounds has received a large attention over the last ten years thanks to their incredibly large magnetic anisotropy and slow magnetic relaxation times combined with air stability [14, 69, 70], and represents one of the most promising avenues to deploy single-molecule magnets for technological applications. In this geometry, Dy(III) generally exhibits a very high $J = 15/2$ ground state, generating eight Kramers doublets, each one degenerate in energy in the absence of external magnetic fields. The concept of a magnetic anisotropy D breaks down in this more complex energy state diagram and instead the figure of merit is the energy gap between the Kramers states i and j ΔE_{ij} , more specifically the first excited states energy ΔE_{01} . While all seven energy separations are important for relaxation mechanisms, the first one needs to be overcome by any one of them in order to jump from the $m=15/2$ to the $m=-15/2$ state. ΔE_{01} can therefore be taken as a measure of magnetic anisotropy and is a key metric related to slow

magnetic relaxation, regardless of the latter being dominated by Orbach or Raman processes [11]. The fitness function is therefore chosen to be ΔE_{01} with the goal of maximizing this energy barrier, as it is positive by definition. Inspired by the compound discovered by [71] in 2016, we focus this application on a Dy(III) atom surrounded by five water molecules as the planar ligands, acting as the static ligand. Water has been shown to act as a weak binding ligand, allowing us to optimize the axial ligand towards a strongly binding one, acting as the dynamic ligand optimized on the SELFIES string level. To our knowledge there is no large scale database for this class of compounds magnetic properties, and as such no RFC model can be trained to speed up the algorithm.

The results of this algorithm can be seen in figure 2.6, with the first panel once again showing the average and maximum growth of ΔE_{01} . The algorithm very rapidly identifies its best candidate in generation 3 with a Kramers gap of 615 cm^{-1} , however we are mostly interested in creating a collection of highly suitable compounds to maximize the probability of synthesizability. The average growth curve shows the rapid discovery of compounds with a separation of over 500 cm^{-1} and even reaching an average in the final generation of 550 cm^{-1} , providing 64 compounds of high quality. This trend is recreated in figure 2.6 panel b), where we show the per generation discovery of compounds, rapidly populating the above 400 cm^{-1} regime to the point where the total distribution shown in panel c) is heavily distorted towards high ΔE_{01} values, which are usually a minority.

The compounds discovered here can compete with the top end of the literature, and require additional analysis. In figure 2.6 panel d) we instead of the two best compounds show the distribution of the connecting atom element and charge at various different generations. All except the O- species are wiped out within six generations, as O- is expected to be the strongest binding connecting atom, being able to deliver the highest axial ligand field, a trend already discovered in the Co(II) study.

To understand what kind of ligands the GA use to optimize ΔE_{01} , four of the best performing compounds are shown in figure 2.7 with their first Kramers gap. The theoretical expectation is that the separation is maximized for an as perfect as possible bipyramidal geometry, with strongly bonded axial ligands and weakly bonded planar ligands. While the planar ligand is fixed and can not be directly influenced by the GA, it can influence its orientation in the final optimized geometry indirectly through the electro-static interaction between the axial and planar ligands. A clear trend can be seen in 2.7, the axial ligand having an O- connecting atom, but also having a long chain bending around and interacting with the water molecules, stabilizing their position in the plane. This is in most cases realized by a large charge localization on the tail of the axial ligand, i.e. carbon-carbon triple bonds or ni-

trogen double bonds. It is important to stress that none of this information was provided to the GA, it acts completely without a database or a classifier model as in the Co(II) case.

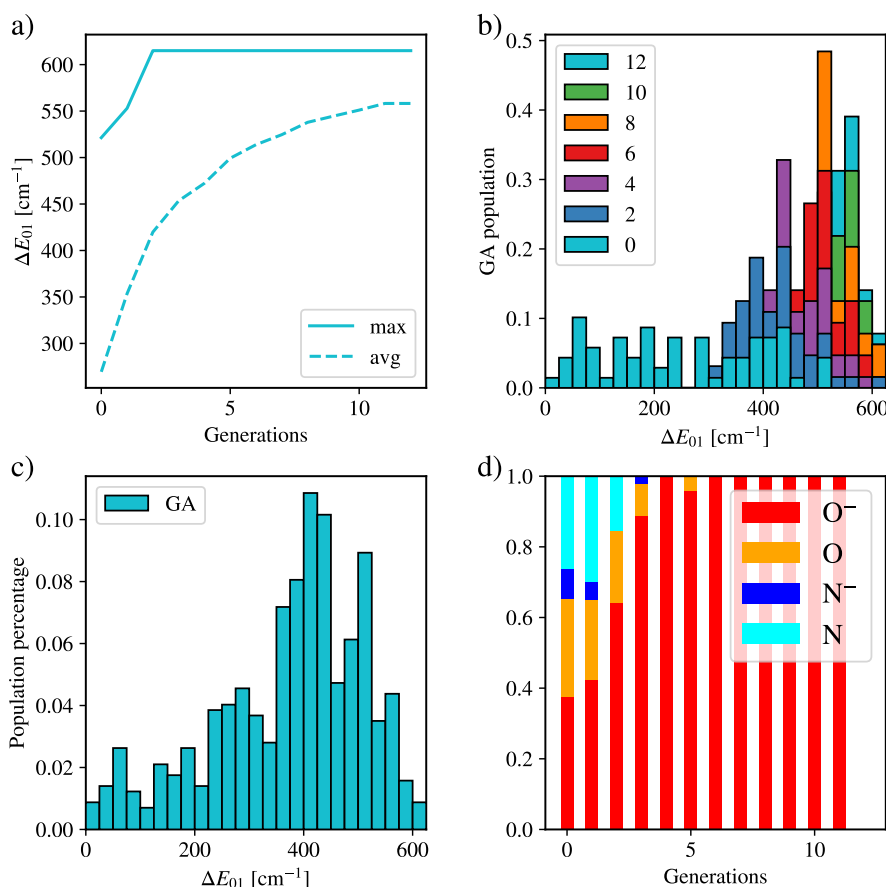


Figure 2.6: **Results for the GA with dynamic encoding on Dysprosium.** (a) Minimum and average Kramers Gap growth. It very quickly found a single compound with maximum Kramer gap of 615 cm^{-1} , yet the growing average shows the number of other compounds above 550 cm^{-1} . (b) Temporal evolution of GA-produced compounds and their Kramer gap, with the new population discovered at regular intervals of generations shown in different colors. After the initial population, it rapidly grows and consistently produces compounds above 550 cm^{-1} . (c) Cumulative distribution of all compounds produced. (d) Distribution of connecting atoms in new compounds generated at each generation. The oxygen ion is quickly identified as the best candidate and all other species quickly die out completely.

2.3 Discussion

Before moving on to the next chapter, we want to briefly summarize the results in this chapter. In this work, two individual implementations of genetic algorithms were created, capable of either exploring a chemical space defined by a list of ligands or creating its own ligands on the fly resulting in an essentially infinite chem-

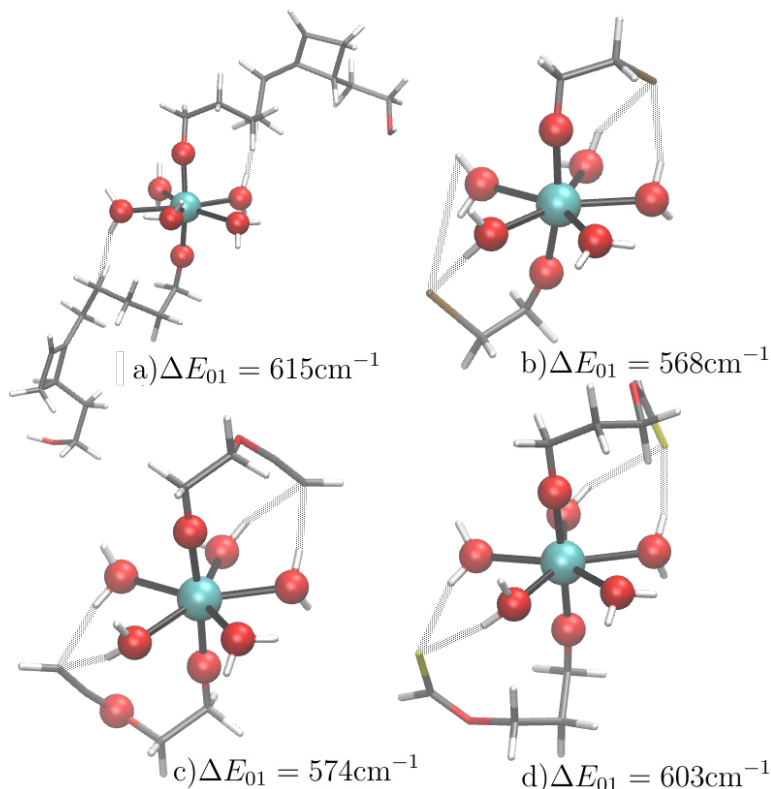


Figure 2.7: **Resulting compounds of Dy(III) GA study.** All compounds within the $\Delta E_{01} > 550\text{ cm}^{-1}$ regime show weak interactions between one or two planar water ligands and each axial ligand, often involving an electron rich group. (Cyan: Dy, Red: O, Grey: C, White: H, Ochre: Cl, Yellow: S.)

ical space to be explored. While both algorithms are kept completely general in regards to ligand number and metal core, they were tested on tetrahedral Co(II) and bi-pyramidal Dy(III) compounds with on the fly calculations, and benchmarked against a previously computed dataset. The benchmarking has shown a computational speedup ranging from a cost reduction factor of 2 to 10, depending on the size of the population and the termination requirement. The static algorithm has created Co(II) compounds with a maximum anisotropy of -220 cm^{-1} , and while not breaching the record of $D=-253\text{ cm}^{-1}$ set by [21], it showed the capability of this simple implementation to identify ligands that break the initial geometry into a planar or seesaw like geometry. The dynamic algorithm in the Co(II) case has resulted in a compound with $D=-234\text{ cm}^{-1}$, again breaking the initial geometry and recreating the trend of negatively charged oxygen being the ideal strong ligand's connecting atom. Additionally, it resulted in an entire generation of 64 specimen with an average anisotropy below -125 cm^{-1} , drastically increasing the chance of experimental synthesis. However, both these applications were limited in their efficiency due to the initial geometry not being ideal for high magnetic anisotropy compounds. Instead, the Dy(III) application created record breaking compounds

with first Kramers gaps of 615 cm^{-1} , and creating a population of 64 specimen with an average of 550 cm^{-1} , drastically increasing the number of known highly anisotropic Dy(III) compounds. Additionally, the compounds analyzed revealed a trend to design axial ligands to enhance weak interactions between the ligands, leading to a stabilization of the axial ligands without modifying them directly, possibly improving future direct approaches of designing ideally suited axial ligands.

3 Results - Graph Neural Networks

3.1 Methodology

3.1.1 Graph neural networks

Graph neural networks (GNNs) are an advanced form of deep neural networks, with the distinct difference of using topological information of the input data. While classical neural networks feed their data unordered into usually fully connected linear layers and activation functions, a graph neural network first restructures their input data into a graph like structure, consisting of nodes, representing usually some real world physical object, and edges, representing the interactions between them. This data split has several advantages: First, with the right implementation this allows for the analysis of datasets of varying size with one model, while classical neural networks have a fixed input size and often require padding to be applied to the data, resulting in wasted resources and unwanted dynamics. Second, this implementation allows for a geometrical interpretation, since the interaction in the model only exists for real world connected objects. Thirdly, there is no ordering of the input data, and as such no artificial correlations arise that can lead to trouble when using different datasets.

GNNs have been used in a number of different fields, such as social sciences representing members of social networks and their connections [72], material science and chemistry [73–77], with each node representing an atom and the edges their bonds/interactions [73], traffic analysis [78–80] and even language processing [81]. Their specific task varies heavily, but can usually be grouped into one of the groups:

- Node/graph prediction: A specific property is predicted based on the input data. This can be either on a node level, with each nodes output feature representing some real world property, or on a graph level

- Node/graph classification: Similar to image recognition, this task requires the identification of either the entire graph or individual nodes into some categories, which usually requires labeled data. On a node level, this can also be achieved with only partially labeled nodes.
- Link prediction: Predicting links between nodes based on edges between their neighbors is closely related to social network analysis/predictions, and allows the network to learn from potentially incomplete data.
- Clustering: Similar to simpler algorithms such as k-means and Bayesian clustering, GNNs can be used to group nodes into clusters based on their “distance”, which can be a more complex set of edge features than just a one dimensional distance.

The first step to any implementation is some sort of embedding, where the raw data is sorted in a way understandable and efficient for a graph representation. This requires the grouping of features into one body features (e.g. atomic numbers in chemistry, personal data in social networks etc...) for the nodes and at least two-body features for the edges (e.g. bond length/type in chemistry, interpersonal connections in social networks). These node and edge features need to be of constant size, but the number can vary between samples. The exact embedding of input data into features can take various different forms:

- One hot encoding: the feature vector is a list of ones and zeros, each one representing whether or not a binary feature is active or not. This implementation, while less general than ordinal encoding, has the advantage of not introducing any artificial relationships between the values of the encoding.
- Ordinal encoding: Each feature is represented by a list of possible integer numbers, with each value corresponding to a possible state of the feature. This allows non-binary features, however can lead to artificial relationships due to assignment of values to properties.
- Trainable encoding: A randomized vector of fixed length is generated for each feature in the initialization of the model, which is constant across nodes, but trainable between epochs. This allows for complex features without artificial relationships, however introduces additional training parameters, requiring larger datasets for convergence.

At the core of any graph neural network is the message passing: an iterative process is performed, where at each step t , a message is calculated for each edge connecting atoms i and j , based on the edge’s features and the feature of the atom j . This message is then combined with the feature of atom i to create the feature at step

$t+1$:

$$\mathbf{F}_i^{t+1} = \mathbf{F}_i^t + \sum_j W(\mathbf{F}_j^t, \mathbf{B}_{ij}), \quad (3.1)$$

where $\mathbf{F}_i^t$ is the feature of node i at timestep t , $\mathbf{B}_{ij}$ is the feature of the edge connecting nodes i and j . The function W here can be of various different shapes, combining the node and edge features and usually contains multiple trainable linear layers acting on the features of individual edges or nodes, keeping the model adaptable to input size. The sum over j represents another important design choice in GNNs: It can either run over all nodes in the network, or only over a specified "neighborhood" of atom i , based on some real world distances between the elements represented by the nodes.

The last block of any GNN is the readout block: the features of individual atoms $\mathbf{F}_i$ need to be mapped to whatever quantity is to be extracted from the network. The most common way to achieve this is by using neural networks that condense the feature into a vector space of desired dimension.

3.1.2 Features

The exact architecture of the developed graph neural network is shown in the result section, however to understand it it is crucial to know what type of features are used to generate the graph information. The raw coordinates of the compounds in theory contain all the information (apart from charge information) required to extract the spin Hamiltonian and its eigenvalues ΔE_{0j} using multi reference methods, and therefore must be sufficient to reproduce them using an advanced enough machine learning model. However, this data is in fact subject to change under translation and rotation, while the resulting Hamiltonian is invariant under translation, and its scalar eigenvalues even under rotation. As such, it is crucial to split the features into two categories: Invariant one-, two- and three-body features that do not change under rotation or translation of the coordinate data and equivariant tesseral features, that do not transform under translation and transform the same way under rotation as the spin Hamiltonian itself.

One body features

We aim to keep the versatility of the model as high as possible, and therefore reduce the amount of chemical information to a minimum. While the performance of the model could potentially be improved by introducing information on the partial charges of each atom, their dipole moments or other DFT based information, we restrict ourselves just to the atomic number as an invariant one body feature. This

allows the network to be trained on data without running any kind of calculations, which would also include experimental data received from x-ray crystallography, such as the CSD.

Two body features

In order to incorporate the interaction between two atoms, their pairwise distance is evaluated and expanded using radial basis functions for all atoms with a distance of below 3.3 Å. This means instead of feeding the scalar distance $d_{ij} = |r_i - r_j|$ to the network, it is expanded as an R -dimensional vector, with R being a freely selectable hyperparameter. The expansion is performed by sampling an exponential function around the actual distance d_{ij} in equal intervals μ_k , resulting in the vector $\mathbf{rbf}_{ij} = \exp(-\gamma|d_{ij} - \mu_k|)$, with μ_k ranging from 0 to 3.3 Å. This results in a gaussian vector, which improves learning rate by introducing additional non-linearity and intuitively transforms the distance between two atoms from a scalar quantity to a position within this R -dimensional vector, allowing the network to process this information more easily. Additionally, a list of bonds is created to facilitate the message passing efficiently.

Three body features

Another important information the network requires about the compound is the angle between bonds. For this purpose, the angle between the vector $\mathbf{r}_{ij}$ and $\mathbf{r}_{ik}$ is evaluated as θ_{ijk} , and introduced as a three body feature. We once again want to introduce additional non-linearity to improve learning, and as such expand this angle in the associated Legendre polynomials of rank one to rank A , leading to an A -dimensional vector $\mathbf{sbf}_{ijk}$. The zero-th order Legendre polynomial is a constant and discarded as such. Only angles of a central atom i and neighboring atoms j and k within 3.3 Å of atom i are considered, leading to angle of atoms that are at most 6.6 Å apart from each other. Additionally, a list of angles is created, with the indices of the central and both neighboring atoms, to facilitate the message passing efficiently.

Tesseral operators In contrary to the Kramers energies, the full spin Hamiltonian is equivariant under spatial rotation of the molecule, and can not be described purely by invariant objects. However, for Dysprosium the Hamiltonian is described by a 16x16 matrix, which contains significant amounts of redundant information. Instead, a more condensed representation is advised if one wishes to efficiently predict this Hamiltonian, given by the Stevens operator decomposition shown in 1.3 and their definition by spherical harmonics shown in 1.4.

As shown early, these spherical harmonics behave under a rotation R by mixing

terms of various m but equal l :

$$\hat{H}(R) = \sum_{l=0}^{\infty} \sum_{m=-l}^l B_{lm} Y_{lm}(R) = \sum_{l=0}^{\infty} \sum_{m=-l}^l B_{lm} \sum_{m'=-l}^l D_{l,m,m'}(R) Y_{lm'}, \quad (3.2)$$

with $D_{l,m,m'}(R)$ being the Wigner-D matrix corresponding to the rotation R . The behavior of the tesseral operators $\hat{O}_{l,m}$ under rotation R can directly be derived from combining equations 1.5 and 1.4.

Being able to model the single ion magnet’s zero-field splitting Hamiltonian as

$$\hat{H}_{ZF} = \sum_{l=2,4,\dots}^{2J} \sum_{m=-l}^l B_{l,m} \hat{O}_{l,m}, \quad (3.3)$$

a straight forward possibility to predict the coefficients $B_{l,m}$ in a way that is equivariant under rotation has been developed in [25]: using the tesseral function $O_{l,m}(\theta, \psi)$, defined the same way based on the spherical harmonic function $Y_{l,m}(\theta, \psi)$ as the tesseral operators and ψ and θ being the azimuthal and polar angle of a vector $\vec{r}$, we evaluate this tesseral function for each atom i based on the relative position to the central Dy atom $\vec{r}_i - \vec{r}_{Dy}$. This relative vector is invariant under translation of the molecule, and follows the same behavior as the molecule under any rotation R , leading to an equivariant tesseral vector $\vec{T}^i$ containing the tesseral components $T_{l,m}^i$. Any linear combination of the atomic tesseral vectors again behaves equivariant under rotation and allows for the prediction of the spin Hamiltonian itself.

3.1.3 GNN Implementation

The graph neural network is implemented using PyTorch. Activation functions and both atom-wise and filter layers can be created using standard PyTorch functionalities, while embedding and message passing require custom layers. The data is loaded in batches of 100 molecules for training purposes, leading to a decent trade-off of training speed and vector operation usage which is favored by large batches versus training efficiency due to smaller batches, allowing for speedup on GPU systems. Additionally to the RBF and SBF features, a list of bonds and angles needs to be computed to facilitate the message passing. The bonds list tensor contains tuples of length 2, which merely contain the index of both atoms, and the angle list tuples of length 3, with first the index of the target/middle atom of the angle, followed by the two outer atoms. All features and lists are prepared before training and stored to facilitate fast training purposes, however when loading the data into a dataset, the indices of both the the bond and angle list need to be adjusted for the

combination of molecules in each batch: the second molecule’s indices are shifted by the total number of atoms in the first molecule, the third one’s indices by the combined number of the first to molecules etc. This is required for the efficient implementation of the network, with the entire batch’s atomic features stored within one PyTorch tensor, as well as their RBF and SBF features in their own.

We will later on introduce three different models, with their details architecture explained in sections 3.3.3, 3.3.4 and 3.3.5. The first one only uses two-body features, the second uses both that and three-body ones, while the last also incorporates tesseral features to enable equivariant predictions. While higher complexity generally leads to better accuracy in these models, the cost of training is also higher and needs to be taken into consideration.

To analyze the cost efficiency of the models, it is important to know the total number of trainable parameters. Assuming a fixed number of elements N_E , feature number F , RBF number R , SBF number S and message passing number N_M , the total cost for the different models is:

$$P_{\text{Two-body}} = F(N_E + N_M(4F + 5 + R) + \frac{F}{2} + 4) + 7. \quad (3.4)$$

$$P_{\text{Three-body}} = F(N_E + N_M(6F + 7 + R + S) + \frac{F}{2} + 4) + 7. \quad (3.5)$$

$$P_{\text{Equivariant}} = F(N_E + N_M(6F + 7 + R + S) + \frac{F}{2} + 1) + 1. \quad (3.6)$$

For the models used in the results section, this gives a total of 16711, 23716 and 46465 trainable parameters for the three models. For comparison, the simple DNN model only has 1399 parameters, but is also trained over 20,000 epochs to ensure convergence versus the up to 150 epochs of the GNN. The runtime analysis for the GNN is seen in figure 3.1, where we show the averaged runtime per epoch and its uncertainty for training on the maximum size dataset on an Intel(R) Core(TM) i7-10700. The model is implemented with a focus on hardware efficiency, allowing a single epoch for the three-body model on $F=32$, $N_M=3$ to an average epoch time of 6.3 seconds, and for the equivariant model with $N_M=6$ 12 seconds, allowing for efficient training of various sizes to optimize the hyper parameters.

3.2 Results - Dataset Preparation

Contrary to genetic algorithms, graph neural networks require a dataset to be trained on. The number of experimentally investigated molecular magnets is small,

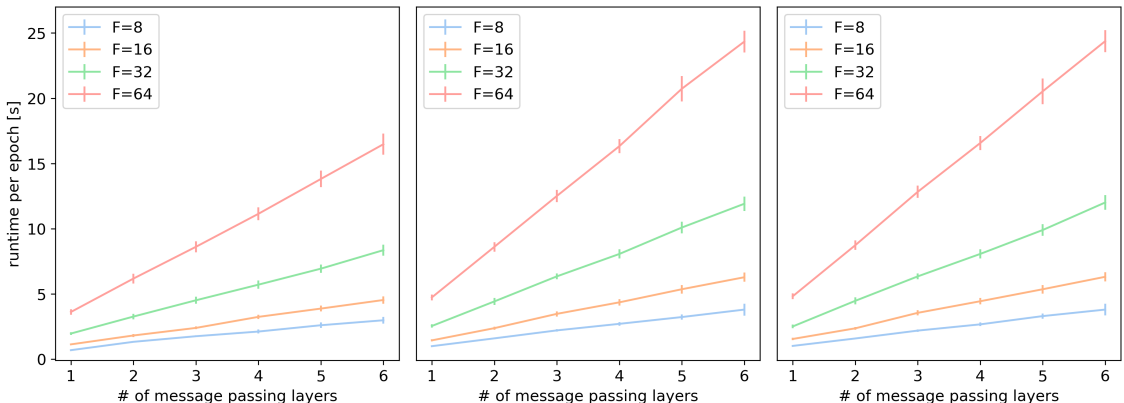


Figure 3.1: **GNN runtime analysis.** Comparison of the GNN runtime against feature number for all three models for feature number between 8 and 64 and between one and six message passing layers. The linear correlation is clearly visible, with the largest actually used model ($F = 32$, 6 message passing layers) only requiring 12.5 seconds per epoch for the entire dataset.

and datasets like COMPASS too diverse for their size to be trained on, as such a custom dataset is needed. Pentagonal bi-pyramidal Dysprosium compounds have shown high Kramers energy splitting in the past, an indicator for slow spin relaxation mechanism due to a high energy barrier. The bi-pyramidal structure is subject to distortions based on ligand interactions, yet some highly geometry-preserving compounds have been found. One example is a Dy(III) metal ion surrounded by five water ligands as the weakly binding planar ligand, and a strongly binding monodentate ligand is attached on both ends as the axial one. This type of compounds has already been shown in the genetic algorithm study to result in high Kramers energies, and is already well understood from said study, therefore representing an ideal candidate for further analysis.

Once again a list of ligands is required. However, this time the requirements are reversed compared to the GA study. Here, we require a large number of relatively homogeneous ligands to enable the GNN to properly learn and extrapolate. As such, we selected the well-established QM9 [28] and QM9 star [82] database of ligands to create a dataset. The extraction of ligands from these datasets has been performed by Dr. Zahra Khatibi, and is aimed to curate a highly homogeneous dataset of ligands: All ligands in QM9 were filtered by their chemical composition and charge, and limited to those only contain H, C, O, N, S, P, and have N or O, either neutral or negatively charged, as the connecting atom. The anions in the QM9 star dataset are originally created from the neutral ligands of QM9 by removing individual Hydrogen atoms.

This final dataset of ligands contains 135695 neutral and 72687 charged ligands,

which then follow the same pipeline as in the static ligand extraction: their bispectrum components are calculated, and a max distance selection algorithm is run on their first two principal components, after which a max distance search is performed to sample the space evenly, textcolorredleading to a final 9889 ligands selected. The result of this can be seen in figure 3.2, with the connecting atoms separated by color, and the finally selected ligands marked in black.

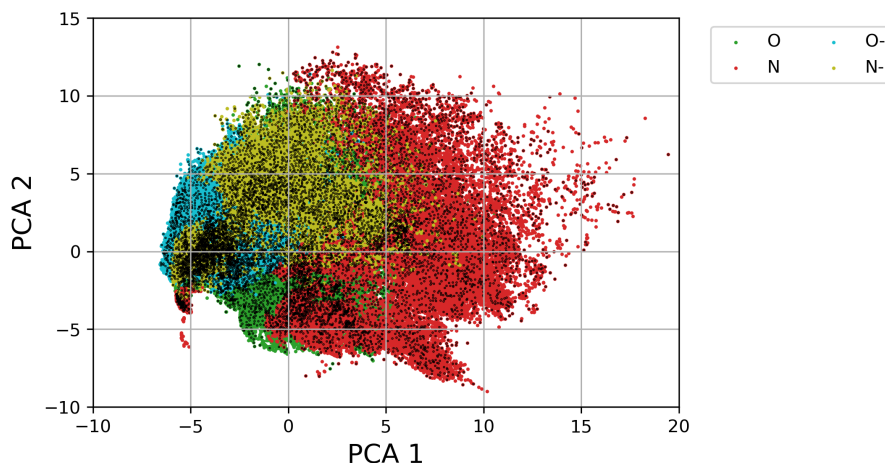


Figure 3.2: **PCA analysis.** First two principal components of the bispectrum components of all ligands in the QM9 and QM9 star dataset fulfilling the element and charge conditions. In black are marked the finally selected ligands.

We create a full list of the compounds created by attaching these ligands to a pre-optimized Dy(III) core with water attached in the plane using MolSimplify. These compounds are then optimized using DFT, and their magnetic properties extracted using CASSCF, which in this case include their full Kramers spectrum, the complete spin-orbit-Hamiltonian and, using the extraction of MolForge, a spherical tensor decomposition of the spin-orbit-Hamiltonian. This results in a final dataset of 9889 compounds, their corresponding 3D-structure and spin Hamiltonian and its eigenvalues. For training and analysis purposes, we use randomly selected subsets of varying sizes, each split into a training set containing 80% of compounds and a test set containing the remaining 20%.

We first show the distribution of Kramers values in figure 3.3. The relative rarity of high values in each level is clearly visible, reinforcing the inefficiency of random search methods in this very unforgiving search. Each level has a strong peak for very small values, containing all unsuited axial ligands that either break the geometry due to ligand interactions, have broken ligands or simply don't bind strongly enough to create a strong anisotropy. The majority of compounds exist in the middle section of relatively constant density, with the small number of high value compounds

creating a small tail, marked in dark blue for the first Kramers level to show the proportion of compounds of interest.

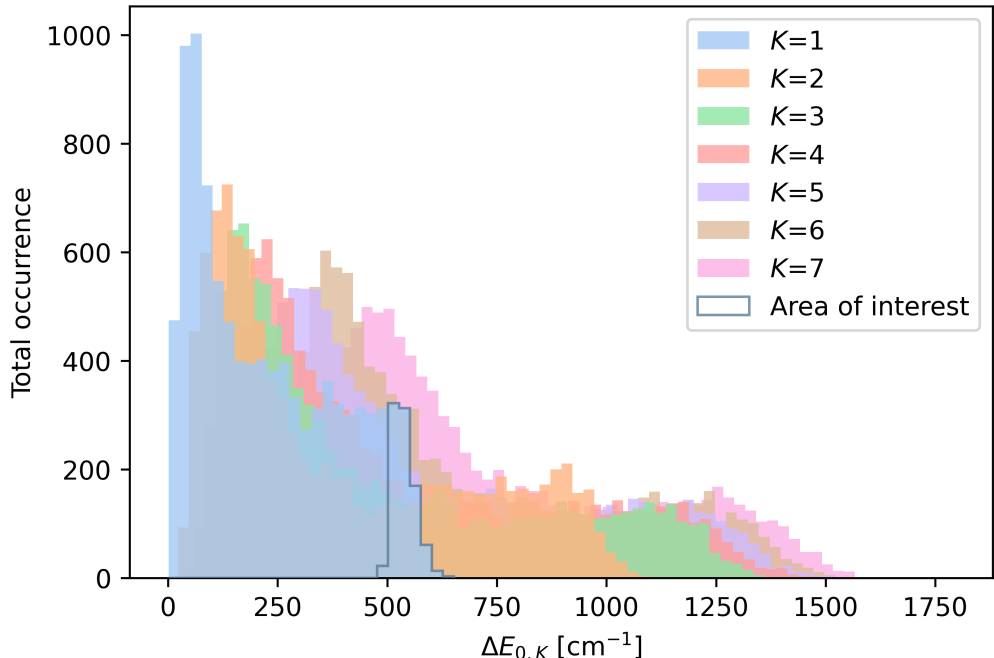


Figure 3.3: **Kramers Dataset Distribution.** Distribution of the k -th Kramers energy for the generated dataset, separated by color. The distribution in each level follows the same trend, with a high peak of undesirable low energy compounds, an even distribution of common compounds, and a small number of high interest compounds, marked in dark blue for the first Kramers level.

We also present the distribution of tesseral components in figure 3.4. However, due to the much less gaussian spread, logarithmic y representation is required. We show them separated by their order in l in three separate plots, and color separated by their magnitude of m . It is clearly visible that the $m=0$ term behaves significantly different from the rest, due to the similar orientation of all compounds along the chosen z axis. This is relevant for the predictive capabilities of the GNN: While we normalize the datasets with respect to their mean value and standard deviation, a highly asymmetric distribution still tends to lead to worse precision.

3.3 Results - GNN

At first we show the result of predicting Kramers energies using a deep neural network (DNN) trained on chemically refined data: instead of feeding the coordinates or distances directly, we select only particular features that are known to be relevant for the magnetic properties and train on those, drastically simplifying the job

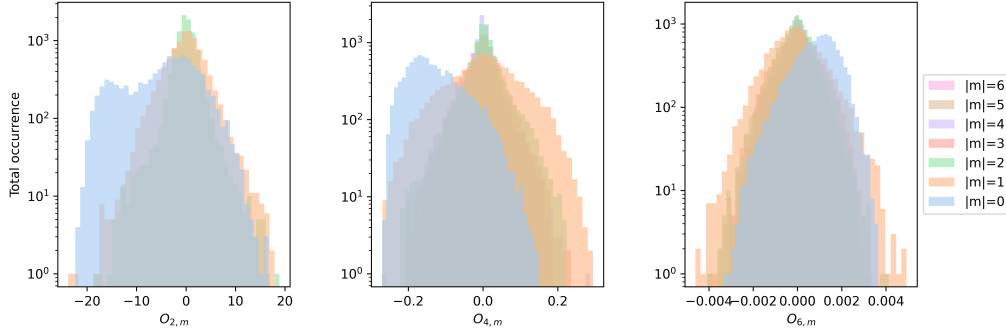


Figure 3.4: **Kramers Dataset Distribution.** Distribution of the tesseral decomposition O_{lm} , separated by order as $l=2,4,6$ left to right, and separated in magnitude of m by color. The $m=0$ component shows significantly different behavior than the other components, due to the approximate z-alignment for most of the compounds.

for the DNN. The performance of this model can then be used to benchmark the capabilities of the GNN: afterwards we present three different levels of complexity of GNN architectures in this section: the first used only invariant one- and two-body features as defined early, and is designed to predict the invariant Kramers energies directly. The second architecture follows the same target, however also includes three-body features to further improve precision and show the superiority of higher order features. The third architecture additionally also includes the equivariant tesseral information, and is capable of predicting the tesseral decomposition of the spin-orbit Hamiltonian, giving access to the full magnetic information, and presenting a new level of complexity and challenges. Each architecture is described individually in this section and shown with their corresponding performance, with their shared components that are identically between architectures first described in a general fashion. Lastly, we investigate the capability of the trained models to generalize to a different tasks, predicting the change in properties under small displacements, which allows for the efficient calculation of derivatives required for the calculation of spin-phonon relaxation times.

3.3.1 Identical Components

Embedding Every atom is only described by its atomic number Z , which is encoded by a randomly initiated vector $\mathbf{x}_i^0 = \mathbf{x}_{Z_i}^0$ of length $F=32$. The encoding is identical for all atoms of the same atomic number, across molecules and training batches. This vector itself is a parameter, and is trained with each epoch to allow for internal understanding of the GNN of each atom type available in the data. It is important to mention that there is no correlation between the atomic number and the chosen vector, and therefore elements not appearing in the training set can not be understood by the network.

Activation Layer Each activation layer applies an elementwise activation function, in our case chosen Sigmoid Linear Unit defined as

$$(\mathbf{x}_i^l)' = \mathbf{x}_i^l * \frac{1}{1 + \exp(-\mathbf{x}_i^l)}, \quad (3.7)$$

in order to introduce non linearity. The SiLU activation function has been chosen due to the best performance among a number of different activation functions.

Atom-wise layer A crucial component, as shown in [73], is the atom-wise layer. It is a linear layer with weights W^l and biases $\mathbf{b}^l$ that are applied on the feature vector of each atom in the dataset individually:

$$(\mathbf{x}_i^l)' = W^l \mathbf{x}_i^l + \mathbf{b}^l. \quad (3.8)$$

As seen, the weights and biases are independent of the atom index i , leading to a number of parameters independent of molecule number and size, and a runtime linear in the total number of atoms in the dataset. Unless further specified, these layers keep the feature dimension constant at $F=32$.

Filter layer These layers are functionally the same as atom-wise layers, however they aren’t applied to atom features, but bond or triplet features and are responsible for mapping these R - or A -dimensional inputs to F -dimensional messages.

Dropout Dropout layers are a common functionality in deep neural networks. With probability $p=0.005$ individual elements of the input data are zeroed out during training, and the final output is normalised with $1/(1-p)$. This has been shown an effective technique to regularize a neural network and prevent overfitting, by effectively randomly switching of the artificial noisy correlation the network might overfit on,

Message passing The message passing layer is where interactions between the individuals atom’s features and their connecting edges are combined together to create a message $\mathbf{M}^l$, which is added element wise to update the features of each atom by one step:

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \mathbf{M}_i^l. \quad (3.9)$$

This is achieved by a combination of atom-wise layers, activation layers and a message creation layer which depends on the features used in the individual architectures. Weights and biases are not shared between layers, allowing for the repetition of Message passing layers as often as required.

3.3.2 alternative simplified DNN using chemical intuition

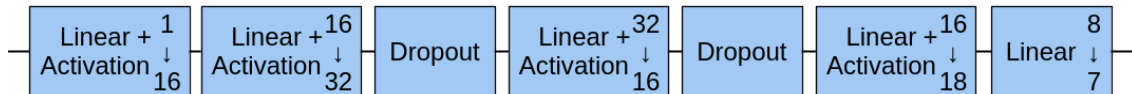


Figure 3.5: **DNN architecture.** The deep neural network consists of 4 block made out linear layers and activation functions, using the same SiLU function as the GNN as well. The individual dimensions are shown in the figure, with the fifth activation layer without an activation function reducing down to seven outputs to predict the seven Kramers layer. Additionally, to dropout layers with a dropout rate of 0.005 are added to reduce overfitting by reducing artificial noise correlations.

To achieve a reasonable level of comparison and investigate the need for advanced graph neural network architectures over standard deep neural networks, a simple DNN is created and trained on the same data. However, since the input dimension is fixed, we can not train it on coordinate data directly, and it is expected to struggle when only feeding certain atom positions since the order matters. Instead, it is trained on geometric properties. While this is common practice in atomistic machine learning, in this case we use the knowledge gathered from previous studies to limit ourselves to properties that are known to be important to maximize the energy separation: the angle between the connecting atoms of the axial ligand and the metal core θ_{Dy} , the distances between them d_1, d_2 , and two types of molecular planarity scores. We fit a plane through the center of mass of the planar oxygen atoms using least square fitting, and calculate the distances from atom to plane d_i , and their maximum and minimum d_{max} and d_{min} , to calculate two scores:

$$MPP = \sqrt{\frac{1}{5} \sum_{i=1}^5 d_i^2}. \quad (3.10)$$

$$SDP = d_{max} - d_{min}. \quad (3.11)$$

An ideal compound would be described by perfectly planar water ligands, orthogonal to this plane and opposing axial ions close to the metal center, creating perfect axial symmetry, corresponding to $MPP=SDP=0$, $\theta_{Dy} = \pi$, $d_{OH}^i = d_{OH}^j$ for all i, j , and small d_1, d_2 . This set of data contains most of the information of the first coordination shell, which is assumed to be the dominant contribution to the energy splitting. This architecture is much simpler, but designed to use only relevant information and therefore expected to still perform well enough to benchmark the GNNs performance. The exact final architecture is shown in figure 3.5, using a combination of activation functions, linear layers to process the data and dropout

layers to reduce overfitting.

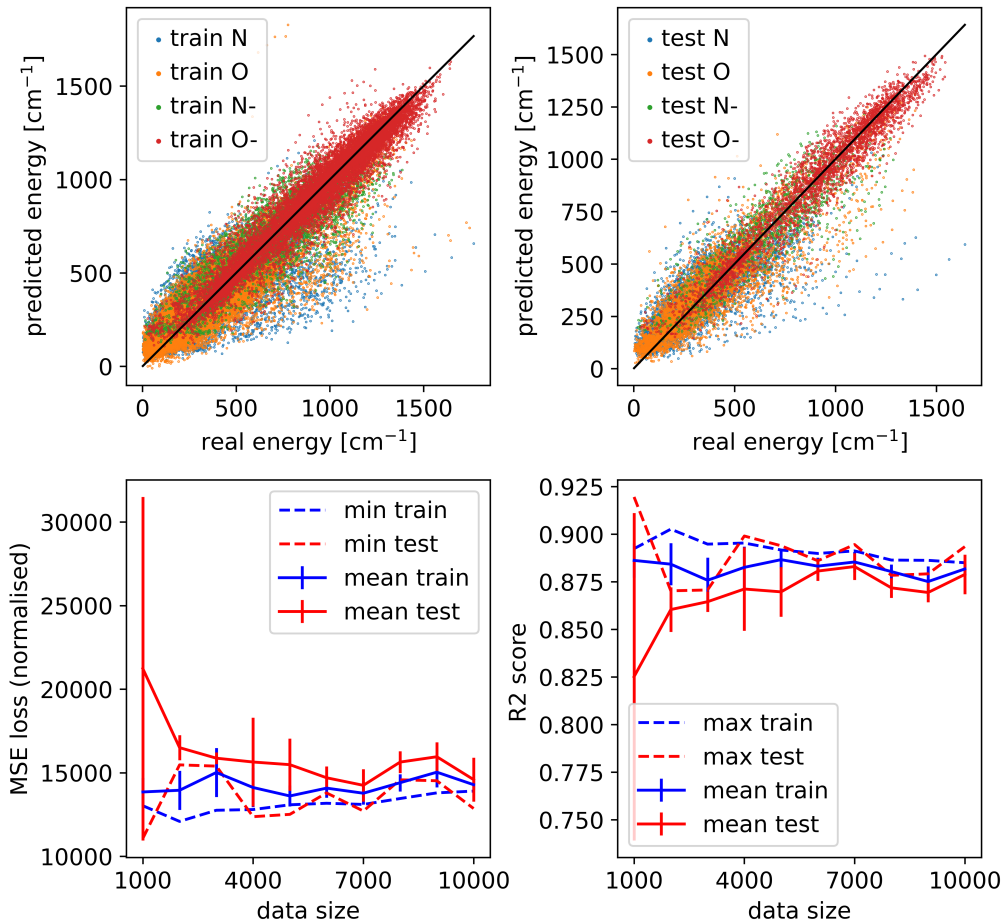


Figure 3.6: **DNN learning analysis.** Top: Training (left) and Test (right) set parity plot between the real CASSCF Kramers gaps and the DNN predictions for all 7 Kramers levels, color coded by the axial ligands connecting atom species. The trend between different connecting atoms is accurately recreated, with a small number of outliers that are being underestimated.

Bottom: MSE (left) and R^2 (right) learning curve for total data sizes of 1000-10000 compounds with the average and best case scenario shown for five runs. The DNN model reaches high R^2 values quickly, but stagnates rapidly due to the limited complexity of the model. Additionally, the standard deviation at each training size is much larger, showing the reduced stability of this model.

We first show the parity plots of the training and test set in the top side of figure 3.6. It becomes clear that the model can give good predictions at a test R^2 level of up to 0.88, however, the parity plot makes it clear that the prediction is significantly worse for small Kramers gaps. We analyze the R^2 score, RMSE, and mean error of this prediction in table 3.1, both calculated per Kramers level and in total, averaged over five runs. The important figure of merit is not only the total R^2 score, but also the mean error on the first Kramers energy, as this ultimately dictates how well one

can use this model to aide in automated analysis of this proxy property. With only 51.2 cm^{-1} , this simple model is already capable of being used as a good predictor of this property. To reflect this predictive capabilities, we also show the parity plot in figure 3.7, however, this time separated by color into the different Kramers levels. It can be seen that, within each Kramers level, the prediction becomes worse for low values. To understand this, it is important to note that the initial geometry of bi-pyramidal compounds is expected to lead to high Kramers gaps, assuming the geometry is conserved, which oftentimes is not the case for the low gap compounds. The DNN appears to simply measure how far away a compound is from the ideal bi-pyramidal geometry and predicts the Kramers energies based on this distortion. This approach works well for small distortions, but breaks down for large ligand movements.

While this model is an important step to create a baseline used to benchmark the newly developed models, it lacks complexity to lead to high accuracy prediction. It only utilizes information within the first coordination sphere, and even within that is limited to a fixed set of angles and distances deemed important. Instead of creating an internal logic of how the Kramers level depend on the geometry, it merely approximates a distance to an ideal configuration with perfect symmetries and distances. Additionally, this approach contains no direct possibility to be extended to equivariant predictions, and is only suitable as a benchmark for the GNN model or a fast reference for on-the-fly calculation speed-up. The significantly larger standard deviation at each point shows the reduced stability of this model and a higher dependency of the initial parameter set. However, this is not a big issue in this case since the training time of all models investigated in this work is below one hour, allowing for the training of multiple models and using the best one / a combination of multiple ones.

Kramers level	1	2	3	4	5	6	7	all
Train R^2	0.83	0.86	0.87	0.86	0.85	0.84	0.83	0.88
Test R^2	0.83	0.86	0.87	0.86	0.85	0.84	0.82	0.88
Train RMSE [cm^{-1}]	69.7	108.1	124.5	128.1	129.7	130.0	133.4	119.5
Test RMSE [cm^{-1}]	69.3	108.0	125.4	129.5	131.0	131.6	135.9	120.7
Train mean error [cm^{-1}]	51.8	77.5	89.6	91.4	92.2	92.7	98.1	84.8
Test mean error [cm^{-1}]	51.2	77.0	89.7	91.6	92.4	92.5	98.5	84.7

Table 3.1: **DNN results.** The R^2 score, RMSE and mean error for both the test and training set prediction of the deep neural network model, separated by Kramers level. While the relative prediction shown by the R^2 score is best for the 3rd and 4th level, the absolute error is smallest for the first, as expected. The model’s hyperparameter have been optimized to lead to minimum overfitting. This ensures similar performance in the test and training set, with statistical noise sometimes leading to better performance in the test set.

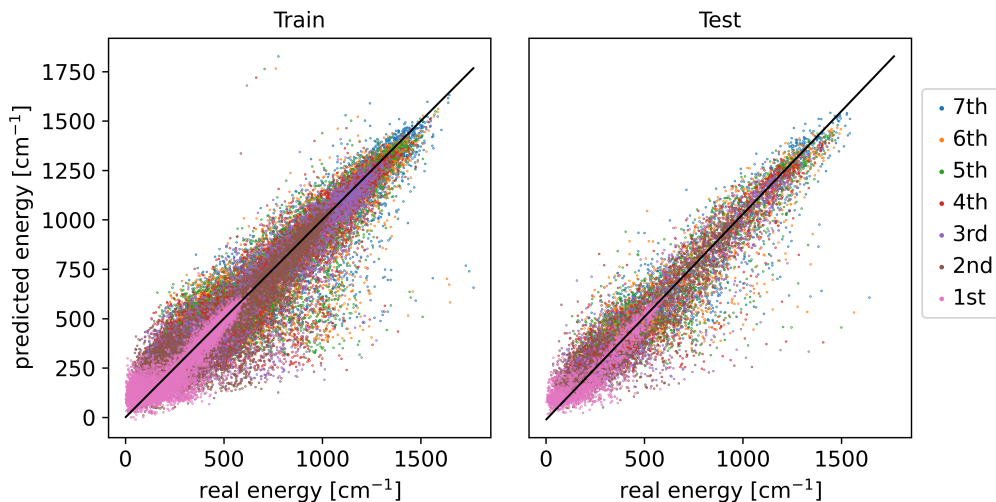


Figure 3.7: **DNN Kramers prediction.** The DNN is capable of predicting the Kramers energies with decent prediction for high values within the respective level, however small values are produced by strong distortion of the bi-pyramidal geometry or ligand-breaking/swapping, leading to bad predictions of the DNN.

3.3.3 Two body architecture

The first GNN architecture uses only atomic numbers and two-body RBF features. The encoding layer assigns each atom a randomly initiated vector based on its atomic number, which is trained with each epoch. The message creation layer is shown in figure 3.8 b), and requires the pre-processed atomic features $(\mathbf{x}_i^0)'$ and bond features $\mathbf{rbf}_{ij}$ and is composed of two filter layers, two activation layers and a dropout layer resulting in the filters $(\mathbf{rbf}_{ij}^l)'$. The final output is a message object M^l , composed of the message vectors $\mathbf{M}_i^l$, which are calculated from the filters multiplied with the corresponding feature vectors $(\mathbf{x}_i^l)'$ as

$$\mathbf{M}_i^l = \sum_j ((\mathbf{rbf}_{ij}^l)' * (\mathbf{x}_j^l)'), \quad (3.12)$$

with the sum over j summing over all atoms j that fulfill $d_{ij} < 3.3$ Å. Intuitively, this element-wise multiplies the features of all neighbors j of atom i with the filter created based on the distance between the two atoms, and adds them up as the message to atom i . This resulting message is then post-processed within the message passing block as seen in figure 3.8 c), and added to the individual atoms features.

The readout block consists of an atom-wise layer which halves the feature dimension, an activation layer and a final layer that reduces the dimension to the final output dimension, in this case seven to predict the seven Kramers level energies, as seen in figure 3.8. The final prediction is then the sum over these vectors across all atoms within a molecule.

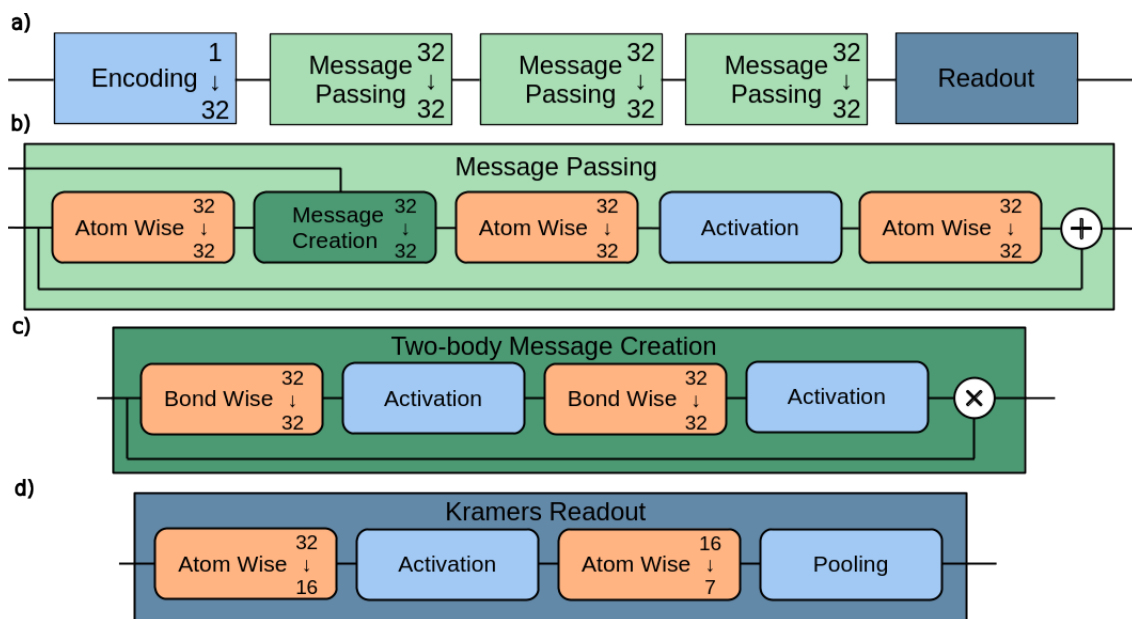


Figure 3.8: **GNN architecture.** Panel a) shows the overall architecture of the GNN, separated into an encoding block, three message passing blocks and a readout block, each one detailed below. Each layer acts either on an atom’s or bond’s feature vector, and is shared between atoms and molecules. As such, the total number of parameters is independent of molecule size or number. Each layer apart from activation layer have the input and output dimension marked to the right. The encoding layer assigns each atom a feature vector of dimension 32 depending solely on the element, shared across all atoms and molecules.

Panel b) shows one individual message passing layer, using the previous atomic features and RBF vectors as inputs. The Atom wise layers are linear layers applied to individual atom’s features, and the activation layers relate to the Sigmoid Linear Unit (SiLU) function. the messages received from the message creation layer is then post-processed and added back to the features, going from $\mathbf{x}_i^t$ to $\mathbf{x}_i^{t+1}$.

Panel c) shows the two-bod message creation layer. The input RBF features are individually processed by two linear and two activation function layers, and in the end multiplied with the message sending atom’s feature.

The last panel shows the Kramers readout layer. Two atom wise layers first half the feature number and than reduce it to dimension seven, after which the individual atomic contributions are summed up to predict all seven Kramers energies at once.

We first show the parity plots of the training and test set in the top side of figure 3.9. The network was trained on the entire 9889 compounds in the training set, and reached a final R^2 score of 0.847 on the training set and 0.852 on the test set. The hyper-parameters (number of message passing layers, feature dimension) were chosen to have the maximum R^2 score without leading to overfitting. The parity plot shows two main results: First, the dominance of ligands with oxygen ions as the connecting atoms in the high ΔE_{01} regime. This is expected, as oxygen ions are known to have a shorter bond distance to the metal core, acting as a strong axial ligands. Second, we also see a higher spread in the low Kramers gap regime. This

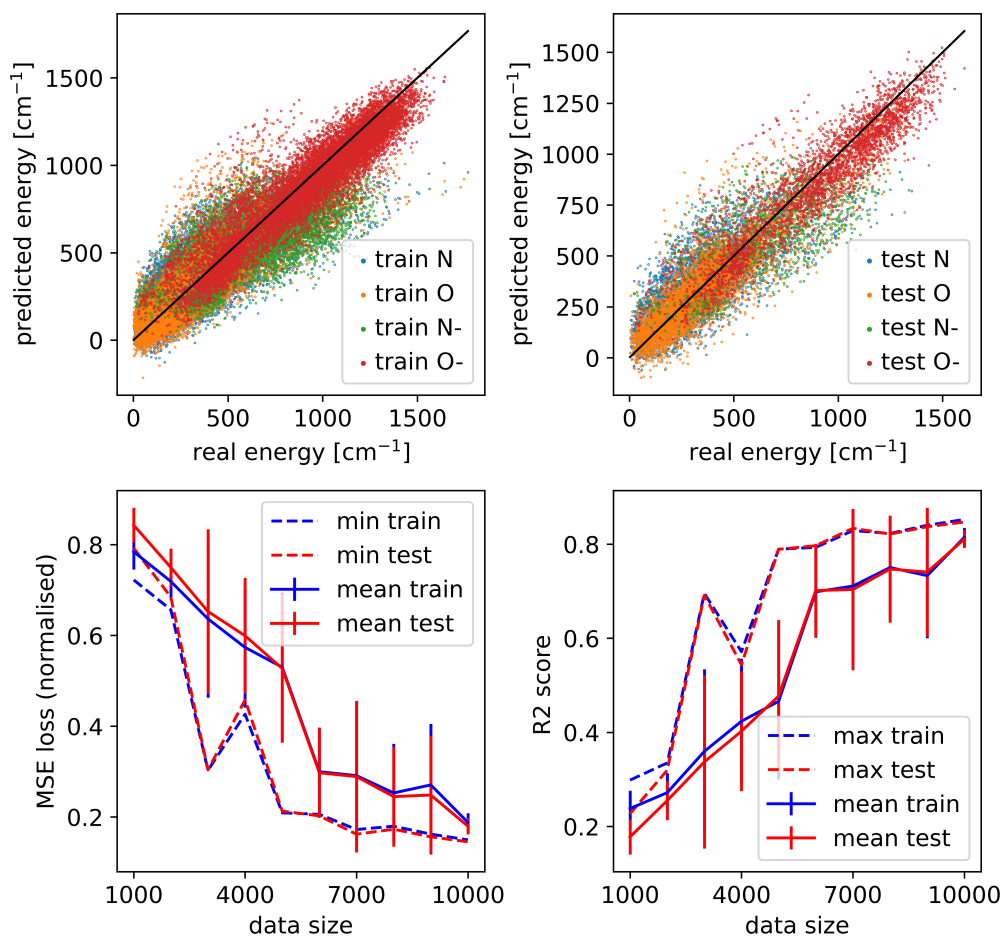


Figure 3.9: **Two body GNN learning analysis.** Top: Training (left) and Test (right) set parity plot between the real CASSCF Kramers gaps and the two-body feature GNN predictions for all 7 Kramers levels, color coded by the axial ligands connecting atom species. While the trend between different connecting atom species is accurately reproduced, the spread is quite large and the model not suitable to be used as a refined predictor of properties.

Bottom: MSE (left) and R^2 (right) learning curve for total data sizes of 1000-10000 compounds with the average and best case scenario shown for five runs. The model reaches R^2 values of 0.8 for dataset sizes of above 5000 compounds, however saturates quickly and displays a high deviation between training runs, leading to noisy result data. While the best case scenario has converged in R^2 and mean error, the average continues to grow, with less noise for larger datasets.

can be explained by the impact of structure distortion on the Kramers gap: the maximum gap is expected for perfect bi-pyramidal structures, and the GNN learns to distinguish how far away a compound is from this perfect structure. In some cases, the original bi-pyramidal structure can be drastically changed due to rotating, swapping or breaking of ligands. These compounds lead to very small Kramers gaps, and are too far away from the training set to be accurately understood based on distances alone. To verify this, the same parity plots are shown in figure 3.10, color separated by Kramers levels. Within each level, the spread becomes larger

the smaller the values are, confirming the lower performance on highly distorted geometries.

In the lower part of figure 3.9 we then show the normalized MSE loss and R^2 score test set learning curve, with the minimum and mean value with its standard deviation of five runs per data size. It becomes clear that overfitting is minimal with the chosen hyper-parameters, and that the best case mean error loss and R^2 value has converged with the current dataset size and would not benefit from further addition of compounds. The model predicts the test data with an average error of 112.4 cm^{-1} (RMSE of 148.0 cm^{-1}), which already allows for the application as a classifier, but is not good enough to perform fully CASSCF independent analyses, which requires a more sophisticated model. The full error analysis in total and per Kramers level is shown again in table 3.2, and shows the same trend of worst R^2 score for the first Kramers level, yet also smallest mean error due to smaller scale.

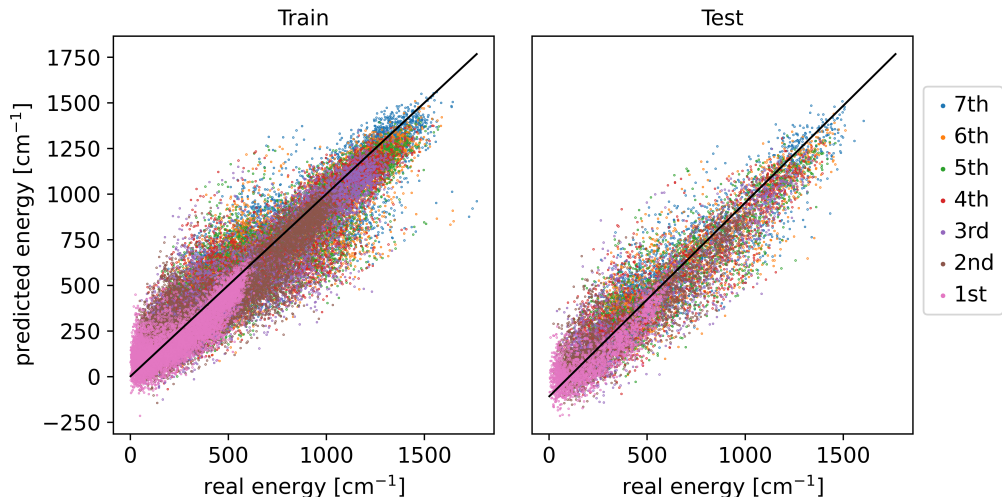


Figure 3.10: **Two body GNN Kramers prediction.** Results of the two-body feature GNN model’s Kramer prediction versus the true values, with the training set on the left and test set on the right, however the colors now separate the individual Kramers levels.

3.3.4 Three body architecture

The second architecture is a direct upgrade of the first: it improves the prediction capability by also including angular information in addition to the distance based ones. This way, the model receives additional information, while also essentially doubling the range of interactions known to the model: Angles of all atoms within $3.3\text{ \AA}$ of a central atom i are incorporated, allowing angles of atoms that are up to $6.6\text{ \AA}$ apart.

The majority of the network stays untouched, with only the message creating block

Kramers level	1	2	3	4	5	6	7	all
Train R^2	0.66	0.74	0.78	0.79	0.77	0.78	0.76	0.81
Test R^2	0.64	0.73	0.78	0.78	0.77	0.77	0.76	0.81
Train RMSE [cm^{-1}]	98.8	147.5	160.6	159.5	160.9	156.4	158.3	150.3
Test RMSE [cm^{-1}]	98.7	145.6	156.5	155.7	157.3	151.5	154.3	147.0
Train mean error [cm^{-1}]	77.0	111.8	120.1	119.2	121.1	116.8	121.2	112.5
Test mean error [cm^{-1}]	77.2	112.3	120.3	119.5	121.0	116.0	120.3	112.4

Table 3.2: **Two body GNN results.** The R^2 score, RMSE and mean error for both the test and training set prediction of the two body model, separated by Kramers level. While the relative prediction shown by the R^2 score is the worst for the first Kramers level, the absolute error is smallest for the first due to the smaller absolute scale. The model’s hyperparameter have been optimized to lead to minimum overfitting. This ensures similar performance in the test and training set, with statistical noise sometimes leading to better performance in the test set.

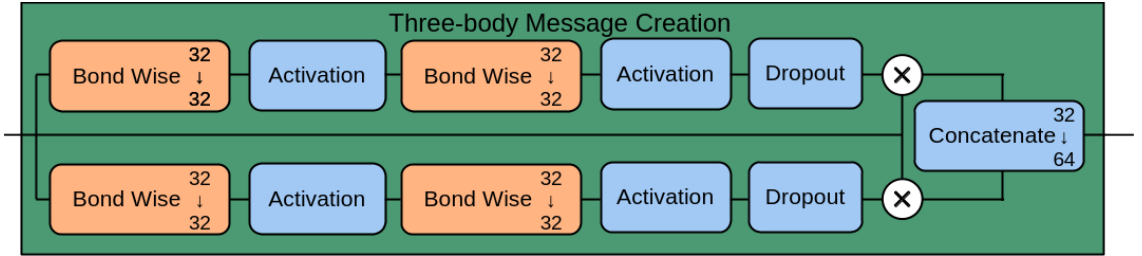


Figure 3.11: **SBF architecture.** The architecture of the three-body feature is identical to the two-body one apart from the message creation and message passing block. Both SBF and SBF features are used in this layer, with each being processed by two linear and two interaction layers. The RBF results are once again multiplied element-wise with the message sending atom, while the SBF results are multiplied with two message sending atoms. The two final messages are then concatenated before being returned to the message passing layer for further processing.

being changed as shown in figure 3.11. It now receives the distance information $\mathbf{rbf}_{ij}$ and angular information $\mathbf{sbf}_{ijk}$, with the first index always being the receiving atoms index. Both are filtered using two filter, two activation and one dropout layer to gain the filters $(\mathbf{rbf}_{ij}^l)'$ and $(\mathbf{sbf}_{ijk}^l)'$, which result then in the messages:

$$M_{1i}^l = \sum_j ((\mathbf{rbf}_{ij}^l)' * (\mathbf{x}_j^l)'), \quad (3.13)$$

$$M_{2i}^l = \sum_j ((\mathbf{sbf}_{ijk}^l)' * (\mathbf{x}_j^l)' * (\mathbf{x}_k^l)'), \quad (3.14)$$

with the final message $\mathbf{M}_i^l$ being the concatenation of the two messages. It is important to mention that the resulting message has twice the dimension as the atomic feature vectors, and is subsequently cut down to the appropriate dimension in the message passing block.

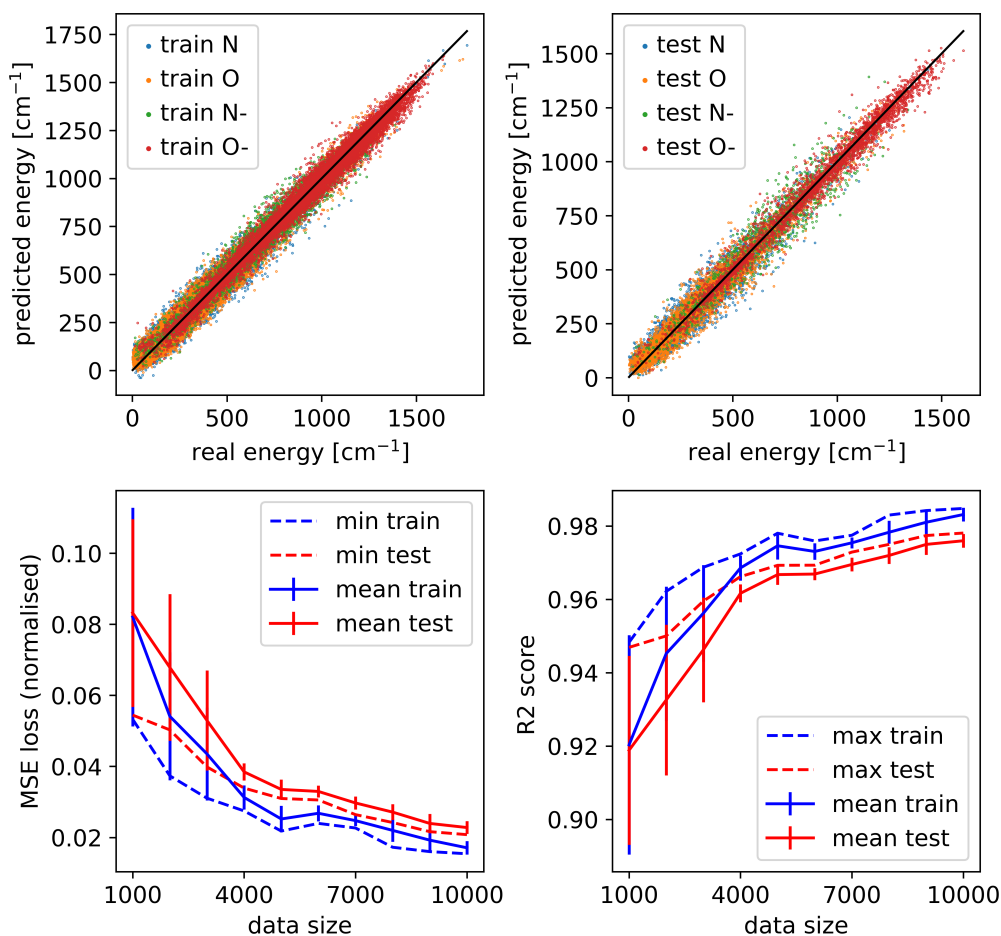


Figure 3.12: **Three body feature GNN results.** Top: Training (left) and Test (right) set parity plot between the real CASSCF Kramers gaps and the three-body feature GNN predictions for all 7 Kramers levels, color coded by the axial ligands connecting atom species. The invariant three-body feature network accurately predicts the trend between different connecting atoms, and achieves high precision predictions for the entire spectrum of compounds and Kramers levels. Bottom: MSE (left) and R^2 (right) learning curve for total data sizes of 1000-10000 compounds. The maximum R^2 (minimum loss) has converged at around 8000 data points, while the average continues to improve until reaching 10000 datapoints. There is a slight overfitting of the test set over the training set, which reaches negligible amounts for large dataset sizes (0.01 R^2 score points).

We perform the same analysis as in the two body feature architecture in figure 3.12, with the parity plots on top comparing the real and predicted Kramers gaps for all seven levels. The model again captures the trend between different connecting atom species, however it predicts the energies with much higher precision, especially in the low energy regime. The addition of three-body features drastically improves the understanding of the model of disturbances of the ideal geometry, and even allows it to accurately predict the behavior for below 50 cm^{-1} , which is usually created by ligand breaking or swapping (axial ligands entering the plane). It appears that

the model developed a proper understanding of molecular magnetism in the case of planar water ligands in this specific geometry. This is also reflected by the lower panel of figure 3.12, where we show the normalized loss and R^2 growth for different total dataset sizes. The R^2 score of the test set reaches near perfect values of 0.975. The mean test error of the prediction reaches an average of 38.9 cm^{-1} (RMSE of 52.4 cm^{-1}). However, it is important to note that this error is not constant across Kramers levels, which is important to see the actual accuracy for the lower level predictions. We therefore show the parity plot separated by Kramers level in figure 3.13, where it becomes clear that the average discrepancy between real value and prediction depends on the scale of the real value. The R^2 , average error and RMSE error averaged over the trained models is shown in table 3.3, with the model being able to predict the first Kramers energy with an average error of just 27.4 cm^{-1} .

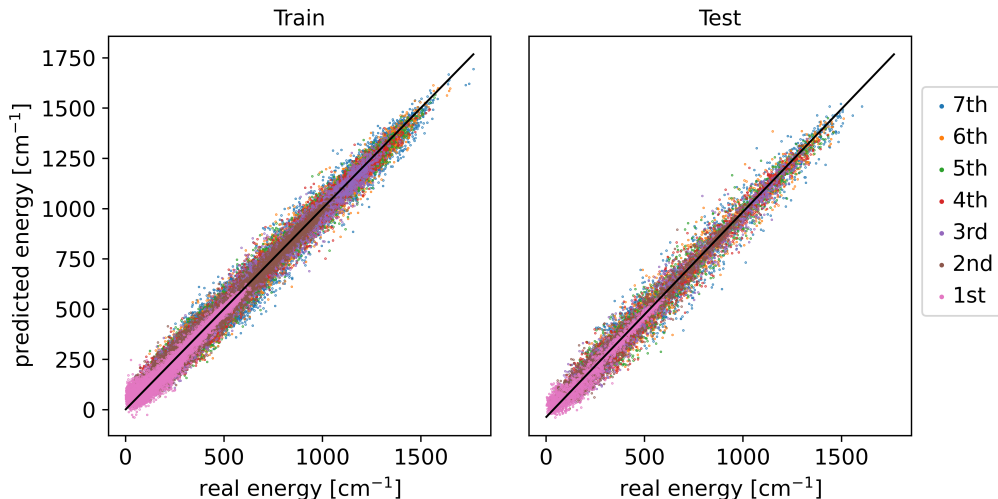


Figure 3.13: **Three body GNN Kramers prediction.** For further analysis, both the train (left) and test (right) parity results of the Kramers energy prediction of the three-body feature GNN model. This time, the individual Kramers levels are separated by color to show the trend predicted: the discrepancy between prediction and true values is smallest for the highest value within each level, corresponding to the model showing the best performance on compounds that closely represent the bi-pyramidal geometry.

3.3.5 Equivariant three body architecture

The third architecture introduces a third level of features in the form of the tesseral operators calculated for each atom up to order $l=6$. This feature is, as shown earlier, equivariant under rotation and allows for the prediction of the tesseral decomposition of the spin Hamiltonian. The encoding and message passing block are identical to the three-body feature architecture, however six message passing layers were required to achieve accuracy. The difference is in the target property and

Kramers level	1	2	3	4	5	6	7	all
Train R^2	0.96	0.98	0.98	0.98	0.98	0.98	0.97	0.98
Test R^2	0.95	0.97	0.98	0.98	0.97	0.97	0.95	0.98
Train RMSE [cm^{-1}]	34.3	39.7	42.4	42.9	46.4	48.1	59.6	45.4
Test RMSE [cm^{-1}]	36.5	44.6	49.3	50.7	55.1	56.7	68.2	52.4
Train mean error [cm^{-1}]	25.6	30.7	32.7	33.1	35.9	37.0	46.1	34.5
Test mean error [cm^{-1}]	27.4	33.7	37.3	38.3	41.7	42.3	51.7	38.9

Table 3.3: **Three body feature GNN results.** The R^2 score, RMSE and mean error for both the test and training set prediction of the three body model, separated by Kramers level. While the relative prediction shown by the R^2 score is best for the 3rd and 4th level, the absolute error is smallest for the first, as expected.

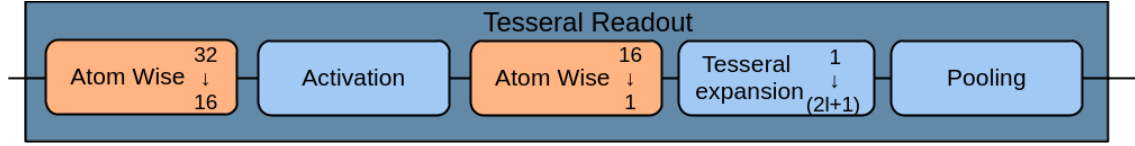


Figure 3.14: **Equivariant GNN architecture.** The equivariant GNN follows a similar architecture as the three-body one, with the difference in the readout block. The individual atoms’ feature vector dimension is reduced to one, and afterwards multiplied with the tesseral vector of order l (and dimension $2l+1$). This guarantees the equivariance of each atom’s final feature, and the final results generated through summation of all atoms in a molecule.

readout block: during the dataset preparation, we calculate the tesseral component vector $O_{lm}(\theta_i, \phi_i)$ of each atom i based on its relative position to the Dysprosium core, expressed in the spherical coordinates θ_i, ϕ_i . The models readout dimension is reduced to one, with each atom therefore predicting a scalar quantity P^i which is subsequently multiplied with the tesseral component vector $O_{lm}(\theta_i, \phi_i)$. The total tesseral components of the molecule are then computed as the sum of these atomic tesseral vectors

$$B_{lm} = \sum_{i=1}^{a_N} T_{lm}^i = \sum_{i=1}^{a=N} P^i O_{lm}(\theta_i, \phi_i), \quad (3.15)$$

which results in the ability to reconstruct the Hamiltonian as:

$$H = \sum_{l=0}^{\infty} \sum_{m=-l}^l \sum_{i=1}^{i=N_a} P^i O_{lm}(\theta_i, \phi_i) \hat{O}_{lm}(\vec{J}). \quad (3.16)$$

The equivariance of this model stems directly from the equivariance of the spherical harmonics, as shown in equation 1.4.

We analyze the model’s performance in two stages: First, we show the direct prediction of the tesseral decomposition components in figure 3.15 for one individual training run. Due to the different orders of magnitude of different orders in l , we

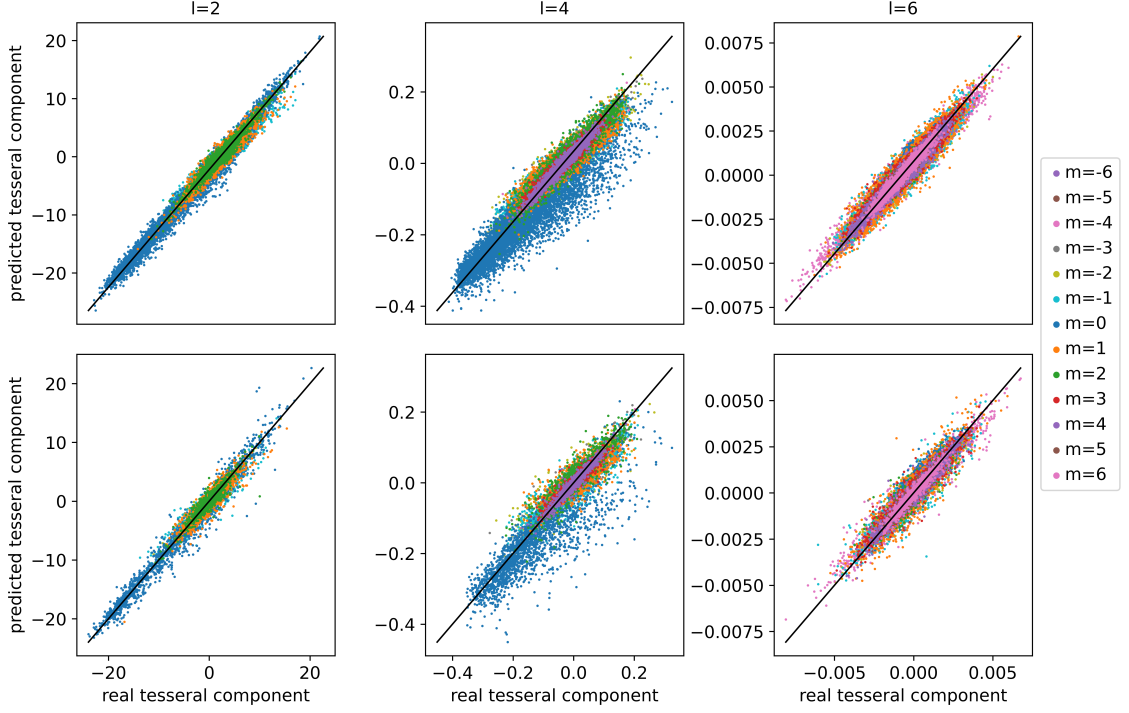


Figure 3.15: **Equivariant GNN predictions.** Train (top) and test (bottom) parity plot for the $l=2,4,6$ tesseral components (left to right) predicted by the equivariant three body model. The agreement in $l=2$ is excellent with an R^2 score of 0.96, while while $l=6$ falls just behind with $R^2=0.89$. The $l=4$ $m=0$ component in particular behaves visible different than the others, due to azimuthal independence of the component. As such, the atomic basis may not be sufficiently large to cover the entire spectrum of the target values, leading to a final R^2 of 0.61 for this specific component.

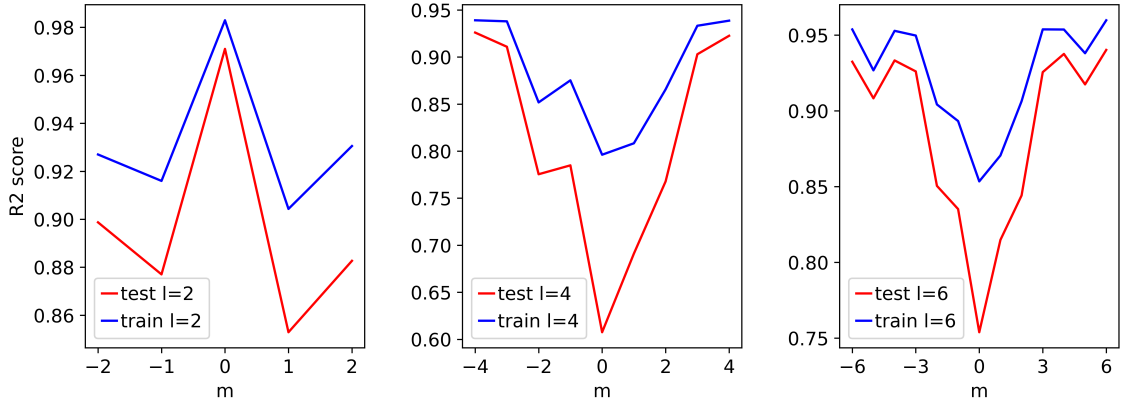


Figure 3.16: **Three body GNN performance analysis.** The R^2 core of the three body GNN model's tesseral prediction. Due to the different orders of magnitude, overall R^2 score fails as metric and instead is evaluated for each oder in l and m individually. It becomes clear that the prediction in higher order $m = 0$ terms leads to the mayor error, due to those terms non-gaussian distribution and basis misalignment.

train one model for each order in l , and combine the results to predict the 27 $l=2,4,6$ components. The error analysis also needs to be performed on an individual l basis, as otherwise the error on the $l=2$ dominates the total. As one can see, the prediction of $l=2$ shows excellent agreement, however $l=6$ has worse R^2 scores and $l=4$ even worse. To understand this, we need to further analyze the source of the discrepancy by analyzing the individual m terms in the following. We show the exact R^2 , RMSE and mean error values in table 3.4 averaged over five training runs, differentiated by order in l , and the $m=0$ ones separately as well. This shows that the higher order $m=0$ terms provide the dominating error, which can also further be seen in figure 3.16, where we show the R^2 score calculated for each l,m pair separately. It becomes clear that the higher order $m=0$ terms behave qualitatively different, and therefore lead to a higher error than the other terms. Additionally, we show the learning with different dataset sizes in figure 3.17 a), which shows high stability in learning with a small standard deviation, however also shows that the performance has not yet converged yet at maximum size in this dataset. However, the total R^2 score is dominated by the $l=2$ contribution, and as such we also show the individual orders' R^2 scores in panel b-d). It once again quickly becomes clear that $l=4$ performs the worst due to the different nature of the $l=4$, $m=0$ distribution. In order to investigate this further, in the supplementary material A we show the distribution of the Hamiltonians tesseral components for each order in l . This time, the relative relation between different m terms is shown, which reveals a slightly non-uniform distribution. Additionally, we also show the one-body tesseral feature vectors of the first coordination shell, which show a highly non-uniform distribution, revealing the source of the bad prediction for $l=4$, $m=0$: the tesseral vectors of all atoms act as a basis the GNN uses and adds up to predict the Hamiltonian tesseral decomposition. In molecular magnetism the first coordination shell is known to have the highest contribution towards magnetic properties, and as such are also most important for the GNN prediction. However the features of those atoms do not cover the space of target vectors, and as such force the GNN to put higher weights on more distant atoms to approximate these regions, leading to bad prediction in $l=4$, $m=0$ terms.

However, observing the error on the order of tesseral components does not provide one with a decent intuition on how well it predicts magnetic properties, since the connection is far too complex to perform any kind of error analysis. Instead, in a second step we reconstruct the spin Hamiltonian based on both the true tesseral components and the ones predicted by the GNN, and compare their eigenvalues (Kramers energies) to each other. This allows us to perform the parity analysis of the Kramers energies in figure 3.18. The analysis shows excellent agreement between

l, m	2, all	2, 0	4, all	4, 0	6, all	6, 0
Train R^2	0.97	0.98	0.89	0.80	0.93	0.85
Test R^2	0.96	0.97	0.81	0.61	0.89	0.75
Train RMSE	0.76	1.07	2.47E-02	5.62E-02	3.02E-04	3.80E-04
Test RMSE	0.96	1.41	3.29E-02	7.75E-02	3.75E-04	4.87E-04
Train mean error	0.52	0.81	1.46E-02	4.01E-02	2.04E-04	2.95E-04
Test mean error	0.64	1.03	1.83E-02	5.46E-02	2.46E-04	3.78E-04

Table 3.4: **Equivariant GNN tesseral results.** The R^2 score, RMSE and mean error for both the test and training set prediction of the equivariant body tesseral components. The value are calculated for each order in l separately due to different orders of magnitude, and also for each orders $m=0$ component individually as this term provides the dominant error.

the two reconstructions, although not quite at the level of the invariant three-body feature model. The final R^2 score reached at 0.949 supports the slightly worse agreement, although it is to be expected due to the much more complex quantity predicted and subsequently boiled down to Kramers energies. We also report the individual Kramers levels R^2 scores and errors in table 3.5 for better comparison with the three-body model, in this case only from a single reconstruction.

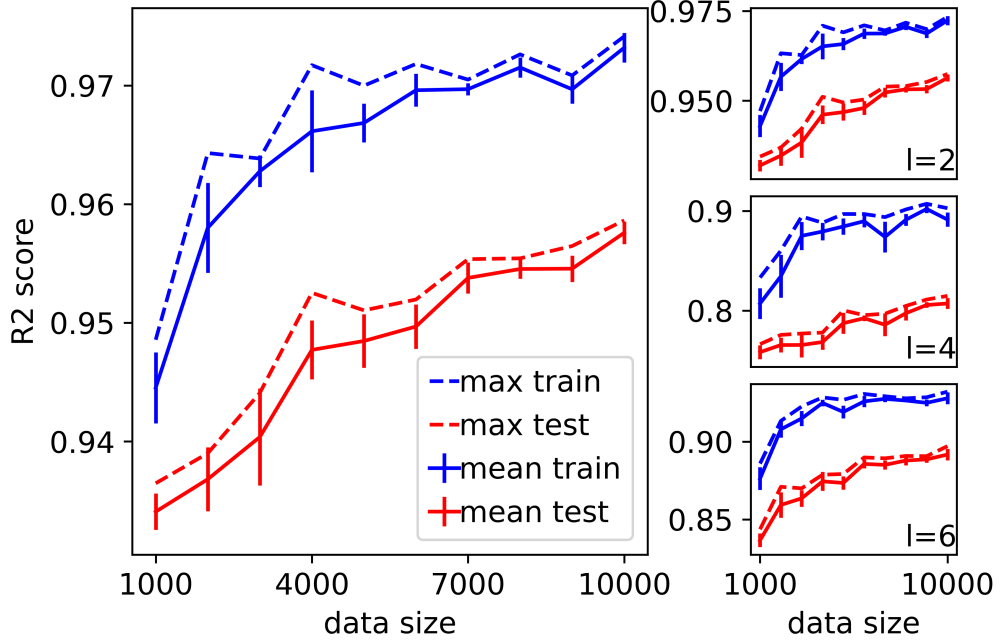


Figure 3.17: **Equivariant GNN Learning curve.** Left: Overall learning curve of the R^2 score of the equivariant models prediction for the tesseral components, averaged over 5 runs with their mean shown in solid lines and the maximum in dashed. The $l=2$ component dominates the R^2 value, and for further analysis the individuals order are shown on the right in panel b-d). The bad performance in the $l=4$ components stems from the $l=4$ $m=0$ component, which has a highly non Gaussian distribution.

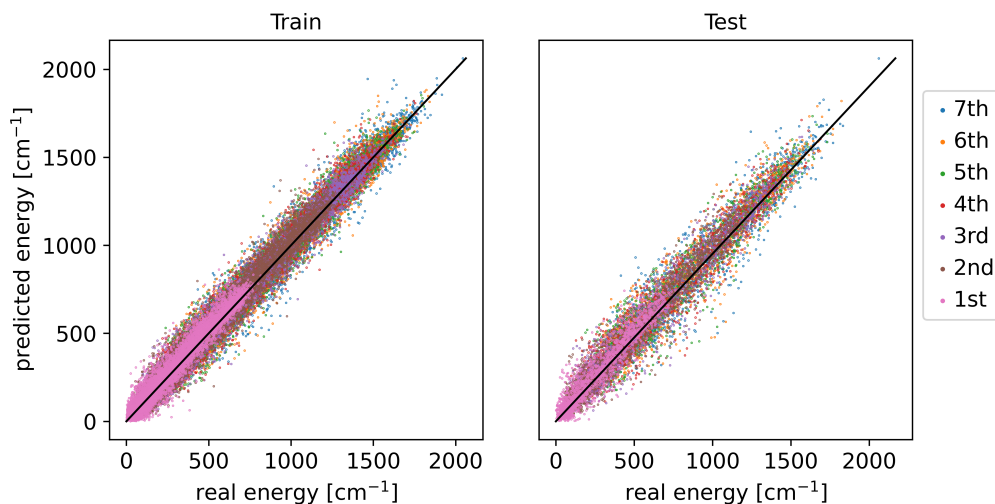


Figure 3.18: **Equivariant GNN Kramers reconstruction.** Training (left) and Test (right) set parity plot between the Kramers energies reconstructed from the tesseral components, separated in color by the Kramers level. The true values (x-axis) we also reconstructed from the true tesseral decomposition, since the limited order reconstruction and read-out precision of ORCA introduces additional errors that are not due to the GNN. The prediction is not as accurate as the direct three-body feature prediction, but still shows excellent agreement with an test R^2 score of 0.949.

3.4 Discussion

In this chapter, 4 different complexity levels of neural network models were implemented and analyzed in their capability of predicting magnetic properties. The first model is a GNN using only one- and two-body features in the shape of element numbers and atomic distances expanded in radial basis functions, and performs the worst, with a final averaged over 5 runs R^2 of 0.847. The benchmarking DNN model, which is trained on refined chemical information, namely molecular planarity scores, axial ligands bonding angles, distances and elements, outperforms this model with an R^2 of 0.905, however, it struggles with highly distorted bi-pyramidal geometries, due to the lower amount of data present and reduced complexity. The introduction of three-body features in the shape of bonding angles expanded in their Legendre polynomial values drastically increased precision: the R^2 score across all Kramers gaps increases to 0.975, a value that allows for almost perfect classification of high-interest compounds. It also features the lowest mean error in the first Kramers gap, at only 27.4 cm^{-1} . The final model is enriched with equivariant tesseral features, adding a whole new level of capabilities due to its capability of predicting equivariant properties. Instead of predicting Kramers energies, it can predict the tesseral decomposition of the spin Hamiltonian with an R^2 of 0.955 across the first three orders. This allows the direct prediction of any kind of magnetic properties

Kramers level	1	2	3	4	5	6	7	all
Test R^2	0.88	0.94	0.95	0.95	0.94	0.93	0.92	0.95
Train R^2	0.94	0.97	0.98	0.98	0.97	0.97	0.96	0.98
Test RMSE [cm^{-1}]	68.20	81.99	82.27	87.16	92.95	96.16	104.98	88.35
Train RMSE [cm^{-1}]	50.39	58.82	58.85	60.78	64.59	65.96	70.91	61.77
Test mean error [cm^{-1}]	48.46	58.96	60.83	64.29	68.88	70.48	77.82	64.24
Train mean error [cm^{-1}]	36.55	43.07	44.00	45.77	49.04	49.84	53.85	46.02

Table 3.5: **Equivariant GNN results Kramers reconstruction.** The R^2 score, RMSE and mean error for both the test and training set prediction of the equivariant body model’s Kramers reconstruction, separated by Kramers level. While the relative prediction shown by the R^2 score is best for the 3rd and 4th level, the absolute error is smallest for the first, as expected.

from the reconstructed Hamiltonian. Specifically, we investigate the reconstruction precision of Kramers energies as these are already used across this work, which this equivariant model performs with an R^2 score of 0.95 and a mean error in the lowest Kramers gap of 48.5 cm^{-1} . While it can not compete with the direct prediction of the three-body model, it still outperforms the other two models and is significantly more universal in its prediction.

4 Conclusion and further development

The development of new molecular magnets with long spin relaxation times is an ongoing problem, and the methods presented in this work propose a possible path to accelerate this challenge using modern machine learning and genetic algorithms. The direct calculation of spin lifetimes of a single compound requires hundreds of thousands of CASSCF calculations, and is limited to a small handful of compounds without extreme resource consumption. Instead, this work focuses on a proxy property, which is cheaper to predict and acts as an indicator of spin lifetime. In the case of Co(II), this is the magnetic anisotropy D , while for Dy(III), the first Kramers gap ΔE_{01} acts as the proxy.

In the absence of a homogeneous training dataset, we first implemented genetic algorithms to more efficiently search the chemical space. While they have been in wide use in general chemistry and have already been used for individual properties of coordination compounds, to our knowledge, this work shows the first application of genetic algorithms to molecular magnets. The two algorithms presented are capable of searching vast chemical subspaces in a more efficient way, reducing the dependency on chemical intuition and potential personal biases. While the benchmarking against an existing dataset revealed a computational speed up of 2-10 times the cost against a random search approach, the final compounds discovered by the static algorithm were not record-breaking. A likely explanation for this is the fact that the first geometry analyzed, tetrahedral Co(II) compounds, was unsuitable for this task, as all high D compounds discovered were predicted to assume a planar or see-saw-like geometry. Nevertheless, the algorithm discovered this need for a different geometry on its own and managed to populate this chemical regime within significantly shorter time than a random search approach, even when limited due to an extremely short genome length. The second application of the dynamic algorithm created Dy (III) compounds which showed a record level for the first Kramers gap, leading to a final energy gap of 615 cm^{-1} . In addition to this and a population of more than 64 highly promising compounds, it also showed a trend in ligands that

maximize the Kramers gap: strongly binding axial ligands, with flexible tails that contain a high charge localization, appear to interact with the hydrogen atoms of the planar water molecules and stabilize their planar location. This information could potentially be used to hand-design new compounds with a high likelihood of being suitable molecular magnets. While the efficient traversal of the chemical space is capable of speeding up the discovery of new compounds, it is difficult to control due to a large number of hyperparameters and randomness in the algorithm itself, and convergence can not be predicted due to the heuristic nature of genetic algorithms. Instead, a speedup in the calculation of magnetic properties itself is required, which brings us to the application of machine learning methods.

Their requirements on datasets first led to the development of a 9889 compound strong, highly homogeneous dataset based on ligands from the QM9 and QM9* dataset. These compounds of bi-pyramidal Dy(III) complexes with water molecules attached in the plane are expected to show a wide range of different distortions to the geometry, and different sizes of Kramers gaps. This dataset is, to our knowledge, one of the largest datasets available for molecular magnetic properties, and the high homogeneity allows the training of neural network models to predict these properties.

A number of graph neural networks were consequently designed and trained on this dataset, and subsequently compared against a combination of deep neural networks and chemical intuition. They are trained on individual atoms, pairwise and three-body features, with rising precision the more information is included, and benchmarked against a simpler DNN model that is fed information refined by chemical intuition. The graph neural networks delivered highly accurate predictions of both the Kramers gaps and the tesseral decomposition of the spin Hamiltonian itself, allowing for the evaluation of other magnetic properties. With a peak performance on the training set of an R^2 score of 0.975 and a best-case mean error of only 27.4 cm^{-1} on the lowest one, the Kramers energy prediction reached a precision level that allows for almost perfect detection of highly suitable compounds for manual verification later. The direct prediction of the spin Hamiltonian decomposition using an equivariant GNN leads to a final R^2 score between 0.81 and 0.97, depending on the order of l , giving much more detailed insight into the magnetic properties by reconstructing the entire spin Hamiltonian. This test was also performed and used to reconstruct the Kramers energies from this predicted Hamiltonian, instead of predicting it directly in the GNN, giving a final R^2 of 0.95. While it lags behind the invariant three-body GNN’s predictive capabilities, it still outperforms the DNN, and the trade-off of complete access to the Hamiltonian itself is well worth it.

The algorithms presented in this work are far from the end of their capabilities, and numerous different trajectories can be selected from here. We want to group them into different avenues to better show the distinction between them.

The most straightforward development possibility is to improve the developed mechanisms themselves. The number of hyperparameters in a GA is huge, and more complex choices like changing the exact crossover mechanism, the parent selection method, and others can drastically impact its performance. Additionally, applying the GA to different kinds of compounds with more individual ligands, leading to a larger genome length, is a straightforward step, however, requiring resources to evaluate their properties. The algorithm itself was implemented with maximum modularity in mind, as such could also be applied to higher dentate ligands: metal centers surrounded by two carbon rings have been shown to exhibit excellent anisotropy values, with the arms of the rings being modifiable in many different ways on a SELFIES level.

Similar opportunities arise for the GNN: While the number of layers and feature number have been optimized to prevent overfitting, there are other possibilities to improve the performance: The model allows for any feature based one, two or three atoms. Therefor, one could include direct chemical knowledge extracted from DFT or similar calculations, such as local charges or dipole moments, into the features used by the GNN. This might be able to allow more precise predictions, however comes at the drawback of requiring previous calculation outputs, making the model less suitable to be applied to structures extracted from experiments.

We showed the discrepancy between the tesseral vectors the model is trying to predict and the tesseral vectors of the first coordination shell atoms the model uses for this prediction, limiting its precision. Changing the exact nature of equivariant features to increase the basis generated from the first coordination shell has high potential to improve the accuracy.

The second avenue is the combination of the two models: performing another GA run on Dy(III) compounds, however, equipping it with the GNN instead of performing the calculations completely using ORCA, promises a significant speedup, at least in the CASSCF stage. While the prediction on unoptimized compounds is expected to be significantly worse, and remains to be analyzed exactly, the GNN could be used either as a pre-filter instead of the RFC or even as a direct evaluation on the optimized compound instead of CASSCF. Additionally, GNNs have been very successfully used to predict energies and force-fields, and with this model’s excellent

performance on magnetic properties, which are notoriously hard to predict, it is reasonable to expect it to perform well as a force field predictor for the optimization itself, allowing for completely simulated discovery of compounds, which will require computational treatment afterwards for verification.

The most promising opportunity is the investigation of the distortion prediction capabilities of the GNN model. All efforts in this work have been directed towards predicting anisotropy as a proxy value for spin relaxation. While this is an efficient approach, the numerical evaluation of spin phonon relaxation rates is still prohibitively expensive for large-scale analysis, due to the requirement of first and second order derivatives of the spin Hamiltonian with respect to atomic displacements, leading to hundreds of CASSCF calculations per compound needed. If the GNN is capable of predicting the impact of displacements on the spin Hamiltonian, these derivatives can be efficiently approximated for hundreds of compounds, with the most promising ones being selected for *ab initio* investigation.

A Analysis of Tesseral basis

The accuracy of prediction of the equivariant GNN model drops for larger orders in l , specifically for $m=0$ terms. While this could simply be attributed to the bad distribution of $m=0$ values in the dataset, additional analyses have been performed on this, specifically comparing the distribution of tesseral components. The Hamilton tesseral feature vector has a dimension of $(2l+1)$ for each l , and therefore, visualization as a vector is not viable. Instead, we show the distribution of the target values in this chapter, and compare them to the distribution of the tesseral vector computed by the first coordination shell atoms. While technically the entire molecule contributes to the prediction, the first coordination shell has been shown to be the dominant contribution. If the "basis" spanned by the tesseral vectors of those 7 atoms does not overlap well with the target property, the GNN is forced to apply higher than natural weights to further distant atoms, unless one enriches the basis by using a different tesseral computation method. Figures A.1, A.3, A.5 show the first coordination shell distributions, while figures A.2, A.4, A.6 show the real Hamiltonians distribution for different orders in l . For $l=2$, both the real and first coordination shell feature distribution are relatively uniform, and as such lead to good overlap. However, for $l=6$, the shape of the first coordination shell features starts having more specific shapes, while the target distributions are still uniform, leading to a slightly lower precision. However, for $l=4$, both the real target distributions and the first coordination shell features become non-uniform, and the overlap, especially for $m=0$, becomes smaller, leading to the inability of the GNN model to predict those.

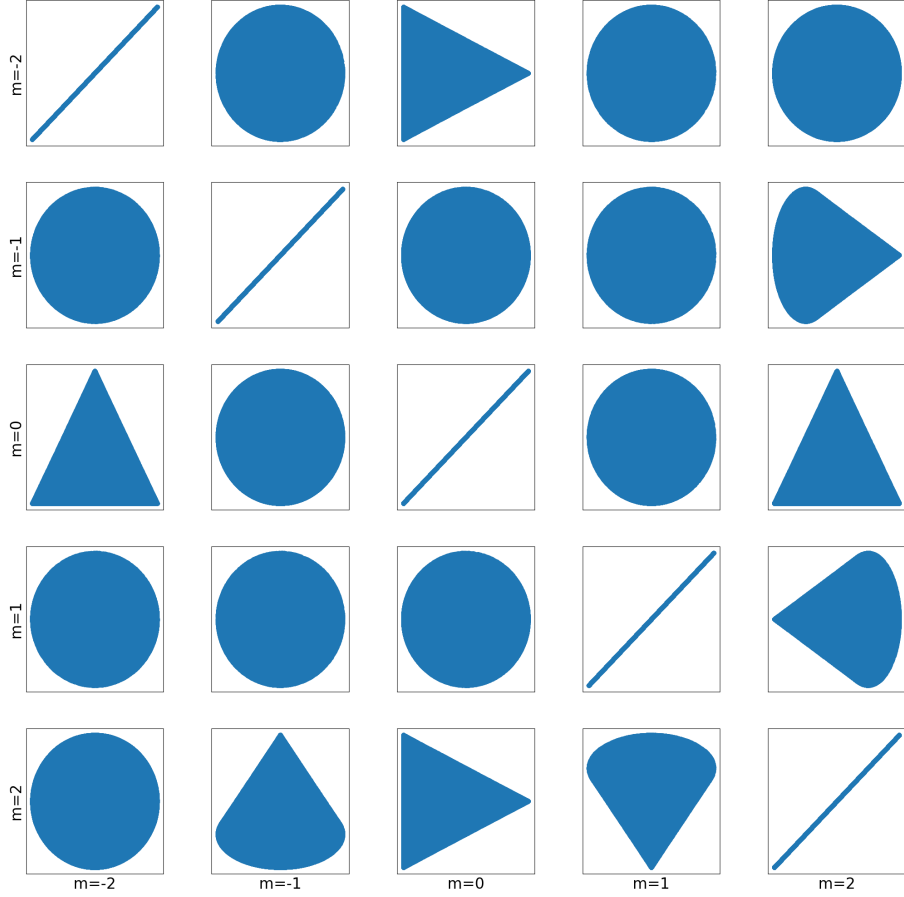


Figure A.1: **FC Tesseral $l=2$** The distribution between different m components of the tesseral components of order $l=2$ for all first coordination shell atoms in the dataset. These individual components create the basis used by the GNN to predict the real tesseral components, and show the non-uniform distribution, which can potentially lead to mismatch between this basis and the target values.

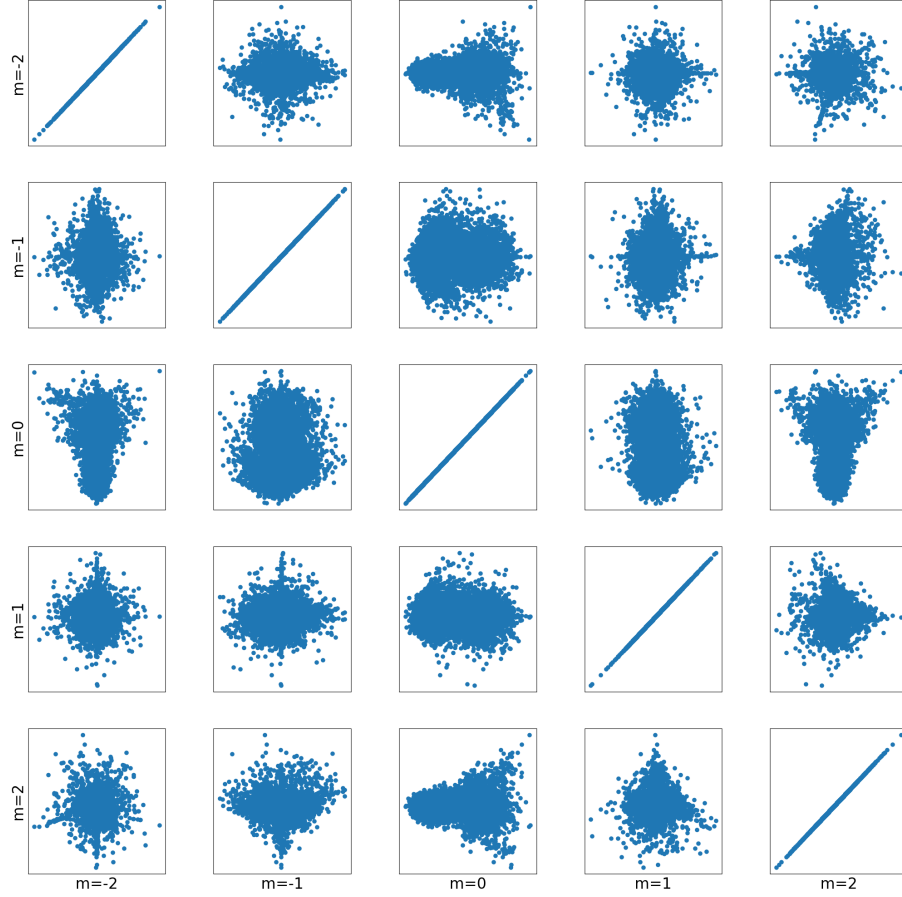


Figure A.2: **True Tesseral $l=2$** The distribution between different m components of the real tesseral components of order $l=2$ extracted from the spin Hamiltonian. These components are the ones to be predicted, and show relatively evenly spread distributions.

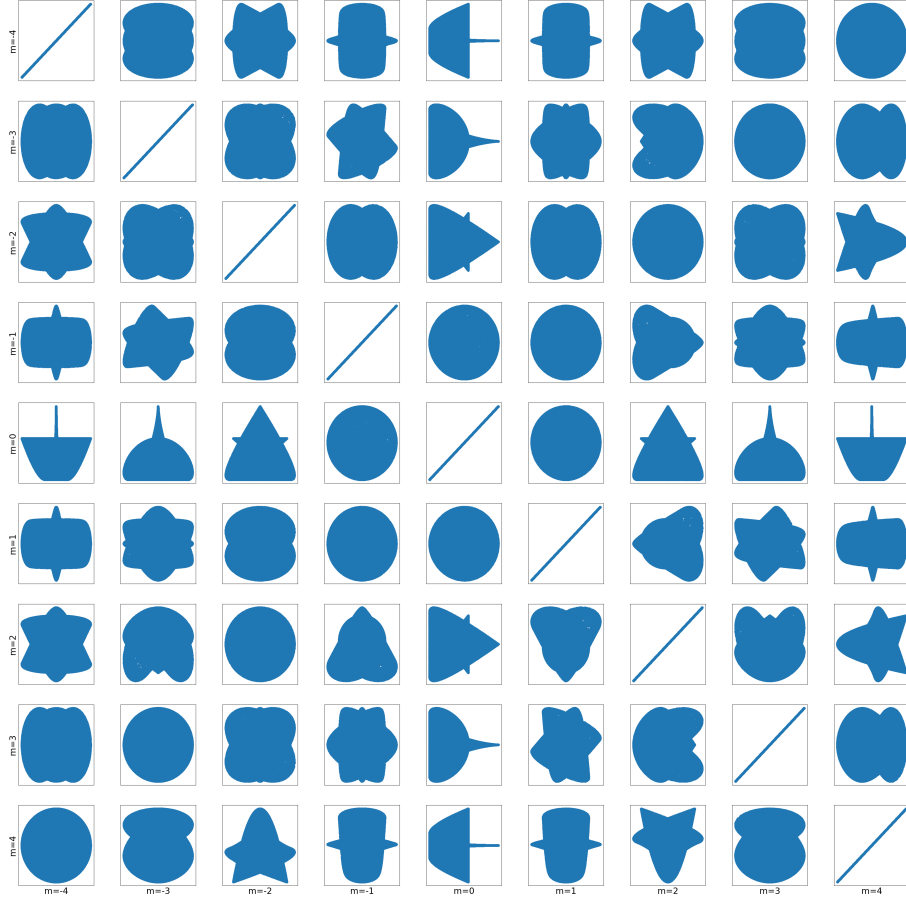


Figure A.3: **FC Tesseral $l=4$** The distribution between different m components of the tesseral components of order $l=4$ for all first coordination shell atoms in the dataset. These individual components create the basis used by the GNN to predict the real tesseral components, and show the non-uniform distribution, which can potentially lead to mismatch between this basis and the target values. Especially the $m=0$ terms have heavily one-sided distributions, leading to an incomplete basis and worse predictions.

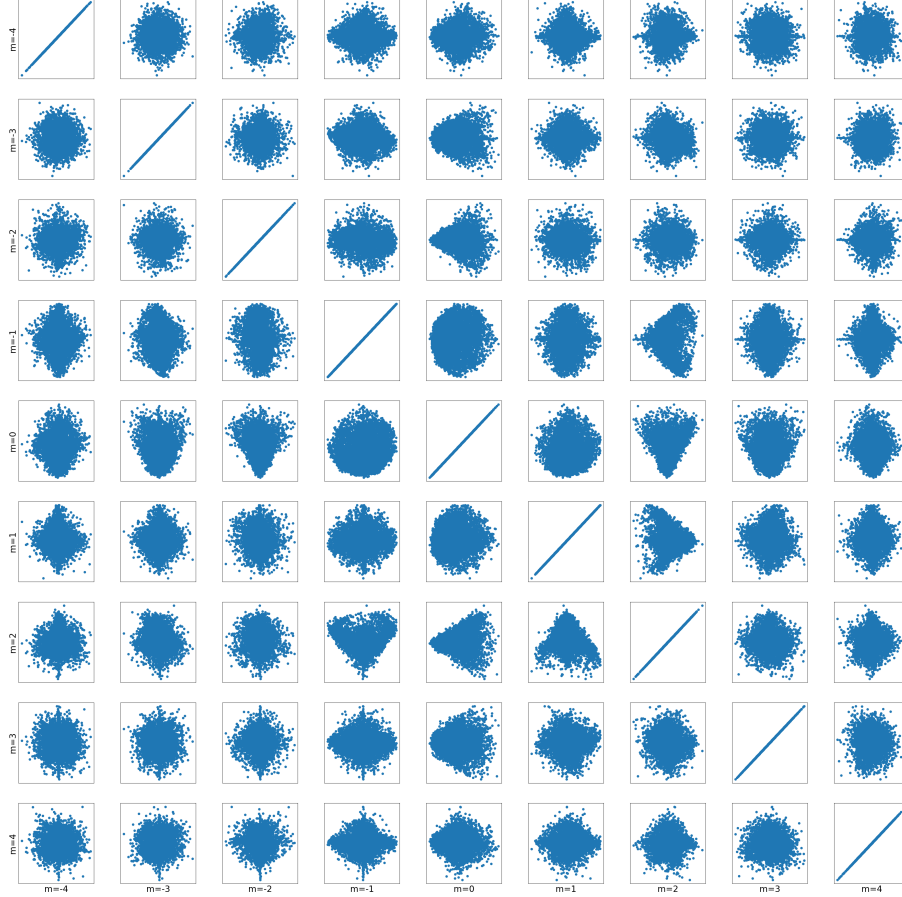


Figure A.4: **True Tesseral $l=4$** The distribution between different m components of the real tesseral components of order $l=4$ extracted from the spin Hamiltonian. These components are the ones to be predicted, and show relatively evenly spread distributions, except for the $m=-1,0,1$ parts, which favor triangular distributions. This mismatch with the basis can lead to reduced prediction accuracy.

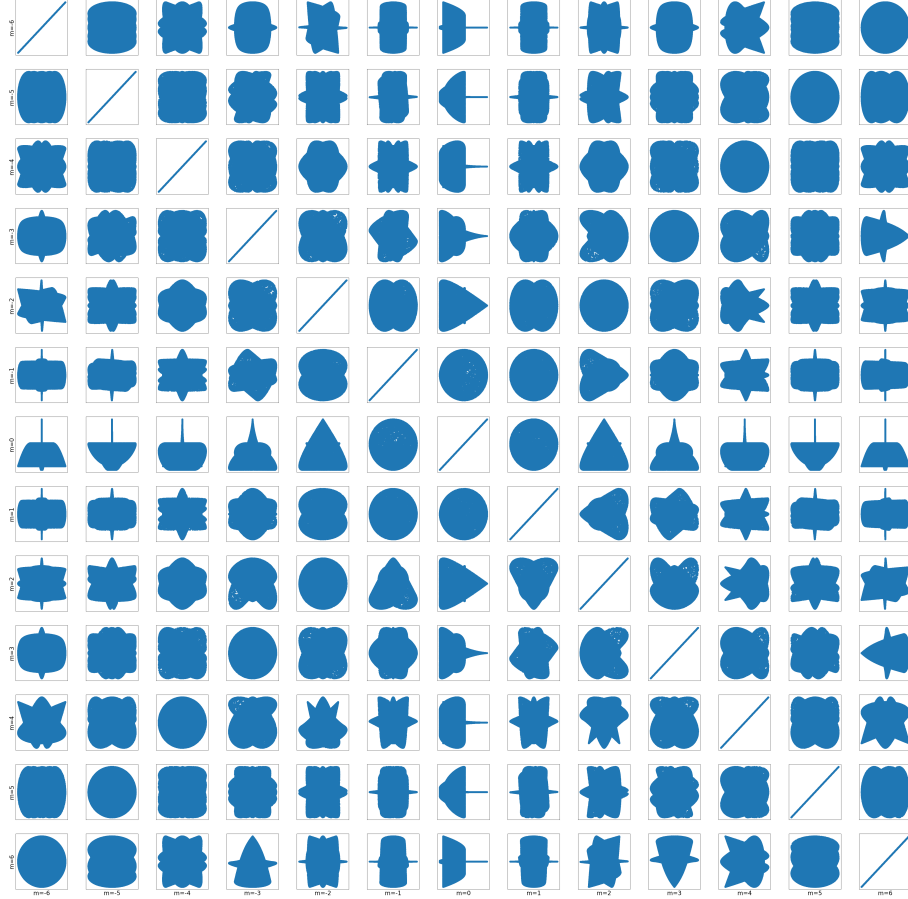


Figure A.5: **FC Tesseral $l=6$** The distribution between different m components of the tesseral components of order $l=6$ for all first coordination shell atoms in the dataset. These individual components create the basis used by the GNN to predict the real tesseral components, and show the non-uniform distribution, which can potentially lead to mismatch between this basis and the target values. Especially the $m=0$ terms have heavily one-sided distributions, leading to an incomplete basis and worse predictions.

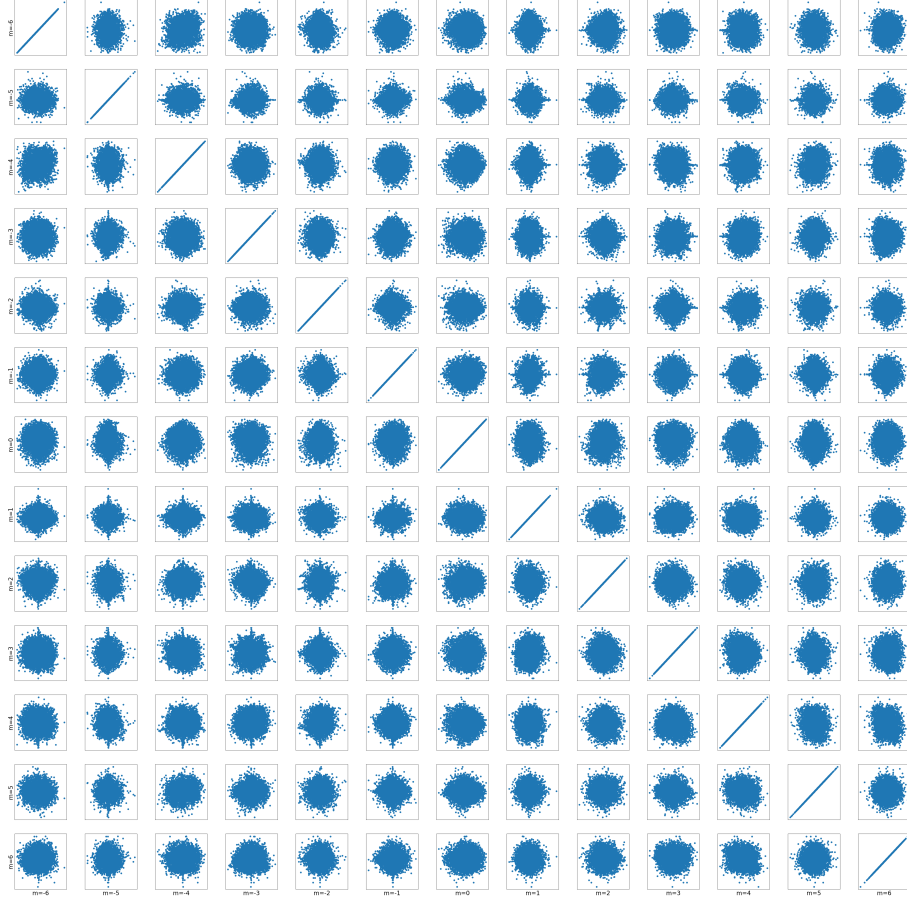


Figure A.6: **True Tesseral $l=4$** The distribution between different m components of the real tesseral components of order $l=4$ extracted from the spin Hamiltonian. These components are the ones to be predicted, and show relatively evenly spread distributions.

Bibliography

- [1] N. F. Chilton, “Molecular magnetism,” *Annual Review of Materials Research*, vol. 52, no. Volume 52, 2022, pp. 79–101, 2022.
- [2] M. Novak, “Magnetic bistability in a metal-ion cluster,” *Nature*, vol. 365, no. 6442, pp. 141–143, 1993.
- [3] G. Gubbiotti, A. Barman, S. Ladak, C. Bran, D. Grundler, M. Huth, H. Plank, G. Schmidt, S. van Dijken, R. Streubel, O. Dobrovolskiy, V. Scagnoli, L. Heyderman, C. Donnelly, O. Hellwig, L. Fallarino, M. B. Jungfleisch, A. Farhan, N. Maccaferri, P. Vavassori, P. Fischer, R. Tomasello, G. Finocchio, R. Clérac, R. Sessoli, D. Makarov, D. D. Sheka, M. Krawczyk, R. Gallardo, P. Landeros, M. d’Aquino, R. Hertel, P. Pirro, F. Ciubotaru, M. Becherer, J. Gartside, T. Ono, P. Bortolotti, and A. Fernández-Pacheco, “2025 roadmap on 3d nanomagnetism,” *Journal of Physics: Condensed Matter*, vol. 37, no. 14, p. 143502, Feb. 2025.
- [4] L. Bogani and W. Wernsdorfer, “Molecular spintronics using single-molecule magnets,” *Nature materials*, vol. 7, no. 3, pp. 179–186, 2008.
- [5] M. Atzori and R. Sessoli, “The second quantum revolution: role and challenges of molecular chemistry,” *Journal of the American Chemical Society*, vol. 141, no. 29, pp. 11 339–11 352, 2019.
- [6] Y. Rechkemmer, F. D. Breitgoff, M. van der Meer, M. Atanasov, M. Hakl, M. Orlita, P. Neugebauer, F. Neese, B. Sarkar, and J. van Slageren, “A four-coordinate cobalt(ii) single-ion magnet with coercivity and a very high energy barrier,” *Nature Communications*, vol. 7, no. 1, p. 10467, Feb 2016.
- [7] P. C. Bunting, M. Atanasov, E. Damgaard-Møller, M. Perfetti, I. Crassee, M. Orlita, J. Overgaard, J. van Slageren, F. Neese, and J. R. Long, “A linear cobalt(ii) complex with maximal orbital angular momentum from a non-aufbau ground state,” *Science*, vol. 362, no. 6421, p. eaat7319, 2018.

- [8] D. Hunger, J. Netz, S. Suhr, K. Thirunavukkuarasu, H. Engelkamp, B. Fåk, U. Albold, J. Beerhues, W. Frey, I. Hartenbach, M. Schulze, W. Wernsdorfer, B. Sarkar, A. Köhn, and J. van Slageren, “Electronic structure of mononuclear and radical-bridged dinuclear cobalt(ii) single-molecule magnets,” *Nature Communications*, vol. 16, no. 1, p. 2157, Mar 2025.
- [9] F.-S. Guo, B. M. Day, Y.-C. Chen, M.-L. Tong, A. Mansikkamäki, and R. A. Layfield, “Magnetic hysteresis up to 80 kelvin in a dysprosium metallocene single-molecule magnet,” *Science*, vol. 362, no. 6421, pp. 1400–1403, 2018.
- [10] D. Piwowska, P. Gnutek, and C. Rudowicz, “Origin of the ground kramers doublets for $\text{Co}^{2+}(\text{3d}^7)$ ions with the effective spin $3/2$ versus the fictitious ‘spin’ $\frac{1}{2}$,” *Applied Magnetic Resonance*, vol. 50, no. 6, pp. 797–808, Jun 2019.
- [11] A. Lunghi, “Toward exact predictions of spin-phonon relaxation times: An ab initio implementation of open quantum systems theory,” *Science Advances*, vol. 8, no. 31, p. eabn7880, Aug. 2022.
- [12] A. Lunghi and S. Sanvito, “Computational design of magnetic molecules and their environment using quantum chemistry, machine learning and multiscale simulations,” *Nature Reviews Chemistry*, vol. 6, no. 11, pp. 761–781, 2022.
- [13] A. Lunghi, *Spin-Phonon Relaxation in Magnetic Molecules: Theory, Predictions and Insights*. Cham: Springer International Publishing, 2023, pp. 219–289.
- [14] Y.-S. Ding, N. F. Chilton, R. E. P. Winpenny, and Y.-Z. Zheng, “On approaching the limit of molecular magnetic anisotropy: A near-perfect pentagonal bipyramidal dysprosium(iii) single-molecule magnet,” *Angewandte Chemie International Edition*, vol. 55, no. 52, pp. 16 071–16 074, 2016.
- [15] P. Hohenberg and W. Kohn, “Inhomogeneous electron gas,” *Phys. Rev.*, vol. 136, pp. B864–B871, Nov 1964.
- [16] X.-N. Yao, J.-Z. Du, Y.-Q. Zhang, X.-B. Leng, M.-W. Yang, S.-D. Jiang, Z.-X. Wang, Z.-W. Ouyang, L. Deng, B.-W. Wang, and S. Gao, “Two-coordinate $\text{Co}(\text{II})$ imido complexes as outstanding single-molecule magnets,” *Journal of the American Chemical Society*, vol. 139, no. 1, pp. 373–380, 2017, pMID: 27936686.
- [17] P. C. Bunting, M. Atanasov, E. Damgaard-Møller, M. Perfetti, I. Crassee, M. Orlita, J. Overgaard, J. van Slageren, F. Neese, and J. R. Long, “A linear cobalt(ii) complex with maximal orbital angular momentum from a non-aufbau ground state,” *Science*, vol. 362, no. 6421, p. eaat7319, 2018.

- [18] P. Kumar Sahu, R. Kharel, S. Shome, S. Goswami, and S. Konar, “Understanding the unceasing evolution of co(ii) based single-ion magnets,” *Coordination Chemistry Reviews*, vol. 475, p. 214871, 2023.
- [19] Y. Rechkemmer, F. D. Breitgoff, M. van der Meer, M. Atanasov, M. Hakl, M. Orlita, P. Neugebauer, F. Neese, B. Sarkar, and J. van Slageren, “A four-coordinate cobalt(ii) single-ion magnet with coercivity and a very high energy barrier,” *Nature Communications*, vol. 7, no. 1, p. 10467, Feb 2016.
- [20] P. C. Bunting, M. Atanasov, E. Damgaard-Møller, M. Perfetti, I. Crassee, M. Orlita, J. Overgaard, J. van Slageren, F. Neese, and J. R. Long, “A linear cobalt(II) complex with maximal orbital angular momentum from a non-aufbau ground state,” *Science*, vol. 362, no. 6421, p. eaat7319, Dec. 2018.
- [21] L. A. Mariano, V. H. A. Nguyen, V. Briganti, and A. Lunghi, “Charting regions of cobalt’s chemical space with maximally large magnetic anisotropy: A computational high-throughput study,” *Journal of the American Chemical Society*, vol. 146, no. 49, pp. 34 158–34 166, 2024.
- [22] J. Liu, Y.-C. Chen, J.-L. Liu, V. Vieru, L. Ungur, J.-H. Jia, L. F. Chibotaru, Y. Lan, W. Wernsdorfer, S. Gao, X.-M. Chen, and M.-L. Tong, “A stable pentagonal bipyramidal dy(iii) single-ion magnet with a record magnetization reversal barrier over 1000 k,” *Journal of the American Chemical Society*, vol. 138, no. 16, pp. 5441–5450, 2016, pMID: 27054904.
- [23] Q.-C. Luo, L. A. Mariano, W.-J. Xu, Z.-H. Li, K.-X. Yu, Y. Peng, A. Lunghi, and Y.-Z. Zheng, “Supramolecular interactions modulate raman relaxation in d5h symmetric Dy(III) single-molecule magnets opening the magnetic hysteresis up to 50 K,” Jul. 2025.
- [24] F.-S. Guo, M. He, G.-Z. Huang, S. R. Giblin, D. Billington, F. W. Heinemann, M.-L. Tong, A. Mansikkamäki, and R. A. Layfield, “Discovery of a dysprosium metallocene single-molecule magnet with two high-temperature orbach processes,” *Inorganic Chemistry*, vol. 61, no. 16, pp. 6017–6025, 2022, pMID: 35420419.
- [25] V. H. A. Nguyen and A. Lunghi, “Predicting tensorial molecular properties with equivariant machine learning models,” *Phys. Rev. B*, vol. 105, p. 165131, Apr 2022.
- [26] V. Briganti and A. Lunghi, “A machine-learning framework for accelerating spin-lattice relaxation simulations,” *npj Computational Materials*, vol. 11, no. 1, p. 62, Mar 2025.

- [27] C. R. Groom, I. J. Bruno, M. P. Lightfoot, and S. C. Ward, "The cambridge structural database," *Acta Crystallogr B Struct Sci Cryst Eng Mater*, vol. 72, no. Pt 2, pp. 171–179, Apr. 2016.
- [28] R. Ramakrishnan, P. Dral, M. Rupp, and O. Anatole von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," Jul 2014.
- [29] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, "Automatic chemical design using a data-driven continuous representation of molecules," *ACS Central Science*, vol. 4, no. 2, pp. 268–276, 2018, pMID: 29532027.
- [30] R.-R. Griffiths and J. M. Hernández-Lobato, "Constrained bayesian optimization for automatic chemical design using variational autoencoders," *Chem. Sci.*, vol. 11, pp. 577–586, 2020.
- [31] Y. Dan, Y. Zhao, X. Li, S. Li, M. Hu, and J. Hu, "Generative adversarial networks (gan) based efficient sampling of chemical composition space for inverse design of inorganic materials," *npj Computational Materials*, vol. 6, no. 1, p. 84, Jun 2020.
- [32] Q. Vanhaelen, Y.-C. Lin, and A. Zhavoronkov, "The advent of generative chemistry," *ACS Medicinal Chemistry Letters*, vol. 11, no. 8, p. 1496–1505, Jul 2020.
- [33] P.-Y. Kao, Y.-C. Yang, W.-Y. Chiang, J.-Y. Hsiao, Y. Cao, A. Aliper, F. Ren, A. Aspuru-Guzik, A. Zhavoronkov, M.-H. Hsieh, and Y.-C. Lin, "Exploring the advantages of quantum generative adversarial networks in generative chemistry," *Journal of Chemical Information and Modeling*, vol. 63, no. 11, pp. 3307–3318, 2023, pMID: 37171372.
- [34] U. Rodemerck, M. Baerns, M. Holena, and D. Wolf, "Application of a genetic algorithm and a neural network for the discovery and optimization of new solid catalytic materials," *Applied Surface Science*, vol. 223, pp. 168–174, 2004.
- [35] K. Huang, X.-L. Zhan, F.-Q. Chen, and D.-W. Lü, "Catalyst design for methane oxidative coupling by using artificial neural network and hybrid genetic algorithm," *Chemical Engineering Science*, vol. 58, no. 1, pp. 81–87, 2003.
- [36] M. Strandgaard, J. Seumer, and J. H. Jensen, "Discovery of molybdenum based nitrogen fixation catalysts with genetic algorithms," *Chemical Science*, vol. 15, pp. 10 638–10 650, 2024.
- [37] L. Terfloth and J. Gasteiger, "Neural networks and genetic algorithms in drug design," *Drug Discovery Today*, vol. 6, pp. 102–108, 2001.

- [38] J. Spiegel and J. Durrant, "Autogrow4: An open-source genetic algorithm for de novo drug design and lead optimization," *Journal of Cheminformatics*, vol. 12, p. 25, 04 2020.
- [39] D. Douguet, E. Thoreau, and G. Grassy, "A genetic algorithm for the automated generation of small organic molecules: Drug design using an evolutionary algorithm," *Journal of Computer-Aided Molecular Design*, vol. 14, no. 5, pp. 449–466, Jul 2000.
- [40] L. Frangoulis, Z. Khatibi, L. A. Mariano, and A. Lunghi, "Generating new coordination compounds via multireference simulations, genetic algorithms, and machine learning: The case of co(ii) and dy(iii) molecular magnets," *JACS Au*, vol. 5, no. 8, pp. 3808–3821, 2025.
- [41] D. Weininger, "SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of Chemical Information and Computer Sciences*, vol. 28, no. 1, pp. 31–36, 1988.
- [42] N. M. O’Boyle, "Towards a universal SMILES representation - a standard method to generate canonical SMILES based on the InChI," *J Cheminform*, vol. 4, no. 1, p. 22, Sep. 2012.
- [43] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, "Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation," *Machine Learning: Science and Technology*, vol. 1, p. 045024, dec 2020.
- [44] A. H. Cheng, A. Cai, S. Miret, G. Malkomes, M. Phielipp, and A. Aspuru-Guzik, "Group selfies: a robust fragment-based molecular string representation," *Digital Discovery*, vol. 2, pp. 748–758, 2023.
- [45] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–73, 1992.
- [46] J. Arifovic, "Genetic algorithms and inflationary economies," *Journal of Monetary Economics*, vol. 36, no. 1, pp. 219–243, 1995.
- [47] X. Xiao, M. Yan, S. Basodi, C. Ji, and Y. Pan, "Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm," 2020.
- [48] K. Man, K. Tang, and S. Kwong, "Genetic algorithms: concepts and applications [in engineering design]," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 5, pp. 519–534, 1996.

- [49] M. T. Bhoskar, M. O. K. Kulkarni, M. N. K. Kulkarni, M. S. L. Patekar, G. Kakandikar, and V. Nandedkar, “Genetic algorithm and its applications to mechanical engineering: A review,” *Materials Today: Proceedings*, vol. 2, no. 4, pp. 2624–2630, 2015, 4th International Conference on Materials Processing and Characterization.
- [50] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future,” *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, Feb 2021.
- [51] B. M. Kariuki, H. Serrano-González, R. L. Johnston, and K. D. Harris, “The application of a genetic algorithm for solving crystal structures from powder diffraction data,” *Chemical Physics Letters*, vol. 280, no. 3, pp. 189–195, 1997.
- [52] N. L. Abraham and M. I. J. Probert, “A periodic genetic algorithm with real-space representation for crystal structure and polymorph prediction,” *Physical Review B*, vol. 73, p. 224104, Jun 2006.
- [53] S. Q. Wu, M. Ji, C. Z. Wang, M. C. Nguyen, X. Zhao, K. Umemoto, R. M. Wentzcovitch, and K. M. Ho, “An adaptive genetic algorithm for crystal structure prediction,” *Journal of Physics: Condensed Matter*, vol. 26, no. 3, p. 035402, dec 2013.
- [54] Z. Falls, P. Avery, X. Wang, K. P. Hilleke, and E. Zurek, “The xtalopt evolutionary algorithm for crystal structure prediction,” *The Journal of Physical Chemistry C*, vol. 125, no. 3, pp. 1601–1620, 2021.
- [55] T. Strassner, M. Busold, and W. A. Herrmann, “Mm3 parametrization of four- and five-coordinated rhenium complexes by a genetic algorithm—which factors influence the optimization performance?” *Journal of Computational Chemistry*, vol. 23, no. 2, pp. 282–290, 2002.
- [56] J. E. Kreutz, A. Shukhaev, W. Du, S. Druskin, O. Daugulis, and R. F. Ismagilov, “Evolution of catalysts directed by genetic algorithms in a plug-based microfluidic device tested with oxidation of methane by oxygen,” *Journal of the American Chemical Society*, vol. 132, no. 9, pp. 3128–3132, 2010.
- [57] Y. Chu, W. Heyndrickx, G. Occhipinti, V. R. Jensen, and B. K. Alsberg, “An evolutionary algorithm for de novo optimization of functional transition metal compounds,” *Journal of the American Chemical Society*, vol. 134, no. 21, pp. 8885–8895, 2012.
- [58] G. Renner and A. Ekárt, “Genetic algorithms in computer aided design,” *Computer-Aided Design*, vol. 35, no. 8, pp. 709–726, 2003, genetic Algorithms.

- [59] D. Hibbert, “Genetic algorithms in chemistry,” *Chemometrics and Intelligent Laboratory Systems*, vol. 19, no. 3, pp. 277–293, 1993.
- [60] S. Kazarlis and V. Petridis, “Varying fitness functions in genetic algorithms: Studying the rate of increase of the dynamic penalty terms,” in *Parallel Problem Solving from Nature — PPSN V*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 211–220.
- [61] M. Safe, J. Carballido, I. Ponzoni, and N. Brignole, “On stopping criteria for genetic algorithms,” in *Advances in Artificial Intelligence – SBIA 2004*, A. L. C. Bazzan and S. Labidi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 405–413.
- [62] B. L. Greenstein, D. C. Elsey, and G. R. Hutchison, “Determining best practices for using genetic algorithms in molecular discovery,” *J Chem Phys*, vol. 159, no. 9, sep 2023.
- [63] M. P. Chetan Chudasama, S. M. Shah, “Comparison of parents selection methods of genetic algorithm for tsp,” *International Conference on Computer Communication and Networks CSI-COMNET-2011*, vol. COMNET, no. 1, pp. 102–105, Decmber 2011.
- [64] A. P. Bartók, R. Kondor, and G. Csányi, “On representing chemical environments,” *Physical Review B*, vol. 87, no. 18, p. 184115, 2013.
- [65] E. I. Ioannidis, T. Z. H. Gani, and H. J. Kulik, “molSimplify: A toolkit for automating discovery in inorganic chemistry,” *Journal of Computational Chemistry*, vol. 37, no. 22, pp. 2106–2117, 2016.
- [66] “RDKit: Open-source cheminformatics,” <https://www.rdkit.org>, 2025.
- [67] F. Neese, F. Wennmohs, U. Becker, and C. Riplinger, “The orca quantum chemistry program package,” *The Journal of chemical physics*, vol. 152, no. 22, p. 224108, 2020.
- [68] D. Rogers and M. Hahn, “Extended-connectivity fingerprints,” *Journal of Chemical Information and Modeling*, vol. 50, no. 5, pp. 742–754, 2010.
- [69] J. Liu, Y.-C. Chen, J.-L. Liu, V. Vieru, L. Ungur, J.-H. Jia, L. F. Chibotaru, Y. Lan, W. Wernsdorfer, S. Gao, X.-M. Chen, and M.-L. Tong, “A stable pentagonal bipyramidal dy(iii) single-ion magnet with a record magnetization reversal barrier over 1000 k,” *Journal of the American Chemical Society*, vol. 138, no. 16, pp. 5441–5450, Apr 2016.

- [70] S. K. Gupta, T. Rajeshkumar, G. Rajaraman, and R. Murugavel, “An air-stable dy(iii) single-ion magnet with high anisotropy barrier and blocking temperature,” *Chem. Sci.*, vol. 7, pp. 5181–5191, 2016.
- [71] L.-L. Li, H.-D. Su, S. Liu, Y.-C. Xu, and W.-Z. Wang, “A new air- and moisture-stable pentagonal-bipyramidal dy(iii) single-ion magnet based on the hmpa ligand,” *Dalton Trans.*, vol. 48, pp. 2213–2219, 2019.
- [72] K. Sharma, Y.-C. Lee, S. Nambi, A. Salian, S. Shah, S.-W. Kim, and S. Kumar, “A survey of graph neural networks for social recommender systems,” *ACM Computing Surveys*, vol. 56, no. 10, p. 1–34, Jun. 2024.
- [73] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, “SchNet – a deep learning architecture for molecules and materials,” *The Journal of Chemical Physics*, vol. 148, no. 24, p. 241722, 03 2018.
- [74] K. T. Schütt, O. T. Unke, and M. Gastegger, “Equivariant message passing for the prediction of tensorial properties and molecular spectra,” 2021.
- [75] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, “E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials,” *Nature Communications*, vol. 13, no. 1, p. 2453, May 2022.
- [76] J. Gasteiger, J. Groß, and S. Günnemann, “Directional message passing for molecular graphs,” 2022.
- [77] C. Ben Mahmoud, L. A. M. Rosset, J. R. Yates, and V. L. Deringer, “Graph-neural-network predictions of solid-state nmr parameters in silica from spherical tensor decomposition,” *The Journal of Chemical Physics*, vol. 163, no. 2, p. 024118, 07 2025.
- [78] A. Nippani, D. Li, H. Ju, H. N. Koutsopoulos, and H. R. Zhang, “Graph neural networks for road safety modeling: Datasets and evaluations for accident analysis,” 2024.
- [79] W. Ma, Z. Chu, H. Chen, and M. Li, “Spatio-temporal envolutional graph neural network for traffic flow prediction in uav-based urban traffic monitoring system,” *Scientific Reports*, vol. 14, no. 1, p. 26800, Nov 2024.
- [80] S. F. Ahmed, S. A. Kuldeep, S. J. Rafa, J. Fazal, M. Hoque, G. Liu, and A. H. Gandomi, “Enhancement of traffic forecasting through graph neural network-based information fusion techniques,” *Information Fusion*, vol. 110, p. 102466, 2024.

- [81] L. Wu, Y. Chen, K. Shen, X. Guo, H. Gao, S. Li, J. Pei, and B. Long, “Graph neural networks for natural language processing: A survey,” 2022.
- [82] M.-J. Tang, T.-C. Zhu, S.-Q. Zhang, and X. Hong, “Qm9star, two million dft-computed equilibrium structures for ions and radicals with atomic information,” *Scientific Data*, vol. 11, no. 1, p. 1158, Oct 2024.