World Academy of Science, Engineering and Technology
International Journal of Nuclear and Quantum Engineering
Vol:18, No:3, 2024

Accelerating Quantum Chemistry Calculations: Machine Learning for Efficient Evaluation of Electron-Repulsion Integrals

Nishant Rodrigues, Nicole Spanedda, Chilukuri K. Mohan, Arindam Chakraborty

Abstract—A crucial objective in quantum chemistry is the computation of the energy levels of chemical systems. This task requires electron-repulsion integrals as inputs and the steep computational cost of evaluating these integrals poses a major numerical challenge in efficient implementation of quantum chemical software. This work presents a moment-based machine learning approach for the efficient evaluation of electron-repulsion integrals. These integrals were approximated using linear combinations of a small number of moments. Machine learning algorithms were applied to estimate the coefficients in the linear combination. A random forest approach was used to identify promising features using a recursive feature elimination approach, which performed best for learning the sign of each coefficient, but not the magnitude. A neural network with two hidden layers was then used to learn the coefficient magnitudes, along with an iterative feature masking approach to perform input vector compression, identifying a small subset of orbitals whose coefficients are sufficient for the quantum state energy computation. Finally, a small ensemble of neural networks (with a median rule for decision fusion) was shown to improve results when compared to a single network.

Keywords—Quantum energy calculations, atomic orbitals, electron-repulsion integrals, ensemble machine learning, random forests, neural networks, feature extraction.

I. Introduction

CALCULATION of the energy of a quantum state is essential in quantum chemistry computations, e.g., in designing new materials for photonic quantum computers. 1-particle and 2-particle density matrices are the main components of tensors used to describe the Hamiltonian whose expectation value is the average energy, including information about inter-particle interactions in the chemical system. However, their computations are extremely computationally intensive, a problem addressed here by a two-pronged approach:
(1) An analytical moment-based formulation, along with an approximation as a linear combination of a small number of moments; and
(2) The application of machine learning algorithms to perform feature extraction and estimation of various coefficients in the linear combination.

Both of these are the first research efforts in this direction, and are expected to help achieve breakthroughs in applications such as photonic quantum material design.

Section II provides the quantum chemistry background relevant to this work. Section III describes the finite linear combination approach, data, and metrics used for evaluation. Section IV describes the application of a random forest learning approach to identify a small set of features sufficient for predicting the signs of the coefficients in the linear combination. Section V describes the use of neural networks to identify features and estimate various coefficients (signs and magnitudes), as well as an ensemble of neural networks that provides a better and more stable estimation of coefficients. Finally, Section VI summarizes the work, and discusses directions for future work.

N. Rodrigues and C. K. Mohan are with the Department of Electrical Engineering and Computer Science, Syracuse University, NY 13244, USA (e-mail: nishantrodrigues97@gmail.com).
N. Spanedda and A. Chakraborty are with the Department of Chemistry, Syracuse University, NY 13244.

II. Background

The average energy of a quantum state Ψ is the expectation value of the Hamiltonian H which can be expressed in the form of contractions of 2-index and 4-index tensors:

$$E_{\text{avg}}[\Psi] = \langle \Psi | H | \Psi \rangle \tag{1}$$

$$\langle \Psi | H | \Psi \rangle = \sum_{pq} H^0_{pq}\, {}^1D_{pq} + \frac{1}{2} \sum_{pqrs} V_{pqrs}\, {}^2D_{pqrs} \tag{2}$$

where ${}^1D_{pq}$ and ${}^2D_{pqrs}$ are *1-particle* and *2-particle density matrices* obtained from the quantum state Ψ. The matrices $\mathbf{H}^0$ and $\mathbf{V}$ are obtained from the Hamiltonian operator H and contain information about the inter-particle interactions in the chemical system. The Hamiltonian H plays a central role in both the time-independent (3) and time-dependent (4) Schrödinger equations, which are indispensable for quantum chemical calculations.

$$H\Psi_n = E_n \Psi_n \tag{3}$$

$$i\hbar \frac{d\Psi}{dt} = H\Psi \tag{4}$$

The tensor elements V_{pqrs} are known as electron-repulsion integrals (ERI) and are important quantities in the fields of quantum chemistry and many-body condensed matter physics. The principal computational challenge in performing quantum mechanical calculations on large chemical systems is in the construction of the 4-index, $\mathbf{V}$, tensor. [1] Many different strategies such as singular-value decomposition, [1] density-fitting, [2] and Cholesky decomposition [3] have been used to reduce the computation cost for the construction of $\mathbf{V}$ tensors.

A. Moments Approach

This section presents a machine learning accelerated approach for efficient construction of the Hamiltonian by developing a moment-based formulation for construction of the electron-repulsion integral tensor, the details of which are presented below.

A set of N_{MO} ortho-normal molecular orbitals $\{\psi_p\}$ are defined below and

$$\int_{-\infty}^{+\infty} d\mathbf{r}\psi_p(\mathbf{r})\psi_q(\mathbf{r}) = \delta_{pq} \tag{5}$$

$\mathbf{r}$ is the three-dimensional Cartesian vector. The two-electron integral is defined as follows, using $\mathbf{V}$, a 4-index tensor of size N_{MO}^4; note that $\mathbf{r}_{12} = \|\mathbf{r}_1 - \mathbf{r}_2\|$ refers to the distance between the two electrons.

$$V_{pqrs} = \int_{-\infty}^{+\infty} d\mathbf{r}_1 d\mathbf{r}_2\, \psi_p(\mathbf{r}_1)\psi_q(\mathbf{r}_1)\mathbf{r}_{12}^{-1}\psi_r(\mathbf{r}_2)\psi_s(\mathbf{r}_2) \tag{6}$$

In the LCAO-MO representation, each molecular orbital is expressed as a linear combination of atomic orbitals:

$$\psi_p(\mathbf{r}) = \sum_{\mu=1}^{N_{\mathrm{AO}}} C_{\mu p}\phi_p(\mathbf{r}) \tag{7}$$

$$J_{\mu\nu\lambda\sigma} = \int_{-\infty}^{+\infty} d\mathbf{r}_1 d\mathbf{r}_2\, \phi_\mu(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1)\mathbf{r}_{12}^{-1}\phi_\lambda(\mathbf{r}_2)\phi_\sigma(\mathbf{r}_2) \tag{8}$$

$\mathbf{J}$ is a 4-index tensor of size N_{AO}^4. Table I illustrates the large sizes of different chemical systems used in quantum dot materials.

TABLE I
SIZES OF CHEMICAL SYSTEMS

Chemical System	N_{MO}	N_{MO}^4	Quantum dot diameter (nm)
Cd_6S_6	168	7.97×10^8	0.76
$Cd_{24}S_{24}$	672	2.04×10^{11}	1.27
$Cd_{45}S_{45}$	1260	2.52×10^{12}	1.50
Cd_3Se_3	84	4.98×10^7	0.50
$Cd_{24}Se_{24}$	672	2.04×10^{11}	1.32
$Cd_{54}Se_{54}$	1512	5.23×10^{12}	1.79
Pb_4Se_4	64	1.68×10^7	0.68
$Pb_{29}Se_{29}$	464	4.64×10^{10}	1.40
$Pb_{52}Se_{52}$	832	4.79×10^{11}	1.59
Pb_4S_4	64	1.68×10^7	0.51
$Pb_{44}S_{44}$	704	2.46×10^{11}	1.54
$Pb_{140}S_{140}$	2000	1.60×10^{13}	2.28

The atomic orbitals are expressed as a linear combination of Gaussian functions, and the integrals needed for the construction of $\mathbf{J}$ are obtained analytically. After the construction of $\mathbf{J}$, the construction of $\mathbf{V}$ is accomplished using the following tensor contraction.

$$\mathbf{V} = \mathrm{Tr}[\mathbf{CCCJ}] \tag{9}$$

Therefore, each tensor element V_{pqrs} can be expressed using (10) given below.

$$V_{pqrs} = \sum_{\mu=1}^{N_{\mathrm{AO}}}\sum_{\nu=1}^{N_{\mathrm{AO}}}\sum_{\lambda=1}^{N_{\mathrm{AO}}}\sum_{\sigma=1}^{N_{\mathrm{AO}}} C_{\mu p}C_{\nu q}C_{\lambda r}C_{\sigma s}J_{\mu\nu\lambda\sigma} \tag{10}$$

The use of moments is inspired by the mathematical connection between the ERI and multipole expansion [4] for the electron-electron interaction kernel. In Cartesian coordinates the, multipole expansion of $\mathbf{r}_{12}^{-1}$ results in the following polynomial in $x_1, y_1, z_1, x_2, y_2, z_2$:

$$\mathbf{r}_{12}^{-1} = \sum_{\mathbf{n}=0}^{\infty}\sum_{\mathbf{m}=0}^{\infty} C_{\mathbf{nm}}x_1^{n_x}y_1^{n_y}y_1^{n_z}x_2^{m_x}y_2^{m_y}y_2^{m_z} \tag{11}$$

where $\mathbf{n} = [n_x, n_y, n_z]$ and $\mathbf{m} = [m_x, m_y, m_z]$ are sets of integer powers in the above expansion and $\mathbf{C}$ is the vector of expansion coefficients. Substituting (11) into (6), the exact expression of V_{pqrs} is obtained in terms of the moments:

$$V_{pqrs} = \sum_{\mathbf{n}=0}^{\infty}\sum_{\mathbf{m}=0}^{\infty} C_{\mathbf{nm}}M_{pq}(\mathbf{n})M_{rs}(\mathbf{m}) \tag{12}$$

where, for $n \geq 0$, the moments along each axis are:

$$M_{pq,n,x} = \int_{-\infty}^{+\infty} d\mathbf{r}\, \psi_p(\mathbf{r})x^n\psi_q(\mathbf{r}) \tag{13}$$

$$M_{pq,n,y} = \int_{-\infty}^{+\infty} d\mathbf{r}\, \psi_p(\mathbf{r})y^n\psi_q(\mathbf{r}) \tag{14}$$

$$M_{pq,n,z} = \int_{-\infty}^{+\infty} d\mathbf{r}\, \psi_p(\mathbf{r})z^n\psi_q(\mathbf{r}) \tag{15}$$

III. METHOD

This section describes the proposed finite approximation approach, followed by data and metrics.

A. Finite Approximation Approach

The result in (12) illustrates the exact map between $\mathbf{M}$ and $\mathbf{V}$. To reduce the amount of computation required, two important approximations are proposed here. First, the infinite summation over all moments is approximated by a finite summation to a maximum moment of N_{mom}. Second, the analytically determined $C_{\mathbf{nm}}$ expansion coefficients are replaced by numerically estimated $\hat{C}_{\mathbf{nm}}$ coefficients, obtained as a result of training machine learning models.

$$\hat{V}_{pqrs} = \sum_{\mathbf{n}=0}^{N_{\mathrm{mom}}}\sum_{\mathbf{m}=0}^{N_{\mathrm{mom}}} \hat{C}_{\mathbf{nm}}M_{pq}(\mathbf{n})M_{rs}(\mathbf{m}) \tag{16}$$

The loss function is defined as,

$$\hat{f}_{pqrs} = [V_{pqrs} - \hat{V}_{pqrs}]^2 \tag{17}$$

to be minimized by training with machine learning algorithms. It is emphasized that the results obtainable can be steadily improved by increasing N_{mom}, i.e., the number of terms in the finite linear combination; the loss function decreases to zero in the limit of taking infinitely many moments, i.e.,

$$\lim_{\mathbf{n}\to\infty} \hat{f}_{pqrs} = 0 \tag{18}$$

B. Datasets

Each input vector (for our machine learning algorithms) was constructed from a set of $\mathbf{M}_{pq}$ and $\mathbf{M}_{rs}$ moments with $N_{\text{mom}} = 2$. This resulted in three sets of moments with $n = 0, 1, 2$ for each Cartesian coordinate (x, y, z), which results in $3^3 = 27$ moments. The moments were generated for both pq and rs pair which results in the total size of the input vector to be equal to $27 \times 2 = 54$. The spatial orbitals were represented by *Gaussian-type orbitals (GTO)* and the training dataset for the machine learning algorithms was generated stochastically, by randomly generating four GTOs and evaluating V_{pqrs}, $\mathbf{M}_{pq}$, and $\mathbf{M}_{rs}$. The GTOs have been used successfully in quantum chemical calculations and is has been demonstrated to yield accurate energies and chemical properties. [5]–[7]

The *GTO integrals API*[1] implements the overlap and moments using 1-electron and 2-electron integrals involving Gaussian-type orbitals. The data generated are stochastic in nature and sampled from a Gaussian distribution. Each row of data contains information about the Gaussian-type orbitals that correspond to 54 features and an interaction term. Using the stochastic API interface, multiple datasets were generated, varying the number of rows (data points), as in Table II.

TABLE II
SAMPLES PER DATASET

Size Label	Number of data points (Rows)
Mini	10,000
Mega	1,000,000
Magna	10,000,000

The *Mini* sets were used for data analysis and developing tree-based approaches, and the *Mega* sets were used to train neural networks. The *Magna* sets were used for testing and validating the trained models.

C. Metrics

The problem is formulated in terms of two separate objectives: predicting the magnitude (strength) of each force, and its sign (attractive vs. repulsive).

To determine the performance of the model in terms of the magnitude of the predicted value, the Mean Absolute Error and the R^2 score are used. The Mean Absolute Error (MAE) is computed as the mean of the absolute difference between the actual and predicted value. The R^2 score is the coefficient of determination that quantifies the predictability of the dependent variable from the independent variable.

Separately, a binary accuracy function compares just the sign of the predicted and actual values.

$$Binary\ Accuracy = \frac{\sum_i sign(y_i \hat{y}_i)}{n} \tag{19}$$

$$\text{where} \quad sign(x) = \begin{cases} 0 & \text{if } x < 0 \\ +1 & \text{if } x \geq 0 \end{cases} \tag{20}$$

[1]This API was developed by the authors, and will be made publicly available, and integrated with JuliaChem libraries.

This binary accuracy value is important since the sign determines the type of interaction. Learning whether the force is attractive or repulsive is considerably more significant than errors in magnitude of the force (emphasized by MAE).

IV. Random Forest Models for Sign Prediction

Random forest models [8] were used to help analyze the complexity of the problem and determine a baseline solution. The model is trained on the mini dataset (10,000 rows); the use of larger data sets did not improve model performance, but increased the training time substantially. A 10-fold cross-validation strategy is implemented to validate the performance of the random forest model.

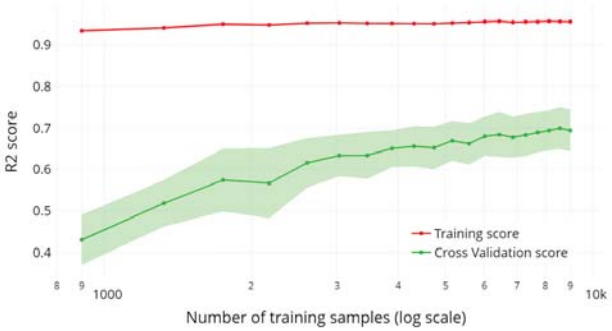

Fig. 1 Training and Cross-Validation R^2 score versus the number of samples in the training set

The learning curve in Fig. 1 shows that the trained model over-fits the data; and that no performance improvement is obtained by using more than 9000 training samples. The validation performance is much worse than the training performance even after training with a large number of samples. The training time increased with the number of training samples, as shown in Fig. 2, without resulting in a performance gain.

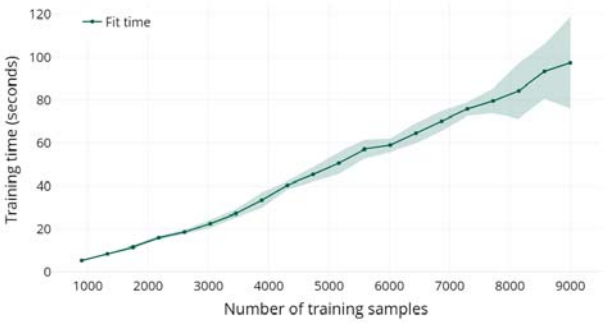

Fig. 2 Random-Forest; time to fit versus the number of samples in the training set

These results imply that the problem is complex and the tree-based models are unable to generate a viable solution for the regression problem. It may also be inferred that the

training feature data are noisy, obscuring the useful "signal" information in the data.

The trained random forest model can also be used to generate a feature importance score that ranks the features based on their impact on decision making. Noisy features have little impact on the predicted value (low feature importance scores) and can be dropped.

Several possible feature selection strategies exist; the simplest one is to just drop the feature with the lowest feature importance score (noisy features). This method is a quick and easy way to reduce model size, but does not account for the non-linear dependencies between the various features. Additionally, it is difficult to determine the number of features to keep/drop as the feature importance score is not comparable to the actual model performance.

Recursive feature elimination [9] overcomes these drawbacks, making it a robust strategy, although it is far more resource-intensive. This approach iteratively trains the model, dropping the least important feature at the end of each iteration. Hence the method trains and scores 53 separate models. The training and validation scores are recorded, and this helps determine the minimum number of features required to maximize model performance.

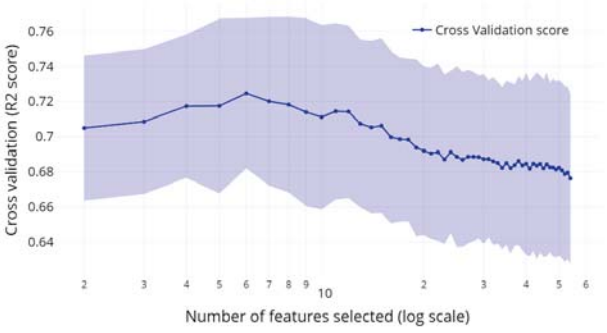

Fig. 3 Cross-Validation score versus the number of features selected using Recursive Feature Elimination

Fig. 3 depicts how model performance varies with the number of features. Only 6 features were required to achieve the maximum R^2 score and additional features had negligible impact on model performance. These 6 features are denoted by indices *(0, 3, 27, 29, 30, 36)*. The results from the feature selection strategies are compared in Table III.

TABLE III
RANDOM-FOREST MODEL PERFORMANCE WITH DIFFERENT FEATURE SELECTION METHODS

Feature set	**Train R^2 score**	**Validation R^2 score**
All features	0.95250 +/- 0.00252	0.67630 +/- 0.04841
Top 20 features	0.95471 +/- 0.00153	0.69168 +/- 0.04969
Top 10 features	0.95734 +/- 0.00196	0.71462 +/- 0.05441
RFE 6 features	0.95992 +/- 0.00095	0.72516 +/- 0.03802

Recursive feature elimination produces the best validation score, but overall the random forest model still overfits. The validation scores are significantly worse than the training results. This suggests that the problem is too complex or nonlinear to be addressed satisfactorily using tree-based models with bounded depth. However, the random forest model performs very well on the binary accuracy metric, as shown in Table IV.

TABLE IV
RANDOM-FOREST MODEL PERFORMANCE

Set	MAE	R^2 Score	Binary acc.
Train	0.011	0.923	0.997
Validation	0.757	0.757	0.995

V. ESTIMATION USING NEURAL NETWORKS

The capabilities of feedforward neural networks were explored, and compared with the random forest baseline results. After examining the performance of shallow neural networks of various sizes, satisfactory results were obtained for the estimation problem (described in preceding sections) using a neural network with two hidden layers; deep networks were not required for this problem.

A. Single Neural Network Model

The results shown here were obtained with a 54-64-32-1 artificial neural network (Neural Networks are implemented using the PyTorch [10] and Flux [11] open-source libraries), i.e., with:

- an input layer of 54 nodes (corresponding to the 54 features, i.e., coefficients in the finite linear combination expression for atomic orbitals),
- first hidden layer of 64 nodes,
- second hidden layer of 32 nodes, and
- a single node in the output layer.

Each of the input and hidden layers used ReLU activation functions. Each network is trained on a *Mega* dataset (1,000,000 rows) and minimizes the mean absolute error using the Adam optimizer. Using a 10-fold cross validation strategy, the network model was demonstrated to achieve an R^2 score of 0.95 on the training and validation set.

When the network inputs were restricted to using only the set of 6 most important features determined by the recursive feature elimination method from the random forest model (mentioned in the previous section), the selected features improved the binary accuracy score slightly, but had a significant detrimental impact on the mean absolute error and R^2 scores. This necessitated further feature extraction efforts.

Unlike random forest models, neural network training does not directly provide any information about feature importance. The added computational overhead of training neural networks iteratively makes it difficult to determine the impact of features using a recursive elimination approach. Furthermore with a smaller set of input features in each iteration, the size of the hidden layers needs to be optimized. Due to these bottlenecks, it is difficult and computationally intensive to use the Recursive Feature Elimination approach for neural networks.

Instead, we used a simple strategy, masking a single feature during model validation by setting it to 0. This allows us to calculate the impact of each feature by comparing the new metrics to the original values. A feature was considered important for the network if its elimination increases the mean absolute error more than a small threshold, chosen as 0.001 in this work. 11 of the 54 features met this threshold, with feature indices *(0, 27, 36, 9, 1, 28, 3, 30, 33, 45, 18)*. There is a large overlap between this feature set and the set of features from Recursive Feature Selection (from the random forest model), supporting the use of this simpler method for feature selection.

TABLE V
NEURAL NETWORK MODEL PERFORMANCE WITH DIFFERENT FEATURE SETS

Feature set	MAE	R^2 Score	Binary acc.
All features	0.01027	0.950	0.961
RFE 6 features	0.04509	0.349	0.972
Selected 11 features	0.00754	0.968	0.977

These 11 features were considered important for the estimation problem, and were used to train a new neural network. The retrained neural network (using the 11 features identified) outperforms the original network on all metrics, as shown in Table V. The overall improvement in performance is also accompanied by efficiency as the smaller network trains faster and requires fewer computational resources. The network now has an input layer of 11 nodes (11/54 features), hidden layer 1 of 64 nodes, hidden layer 2 of 32 nodes and a single node in the output layer. This change reduces the number of trainable parameters by almost 50%, with a substantial reduction in the first hidden layer. However, further reductions in network size (e.g., by halving the number of nodes in the hidden layers) worsened performance [12].

B. Ensemble of Neural Networks

Our approach is intended to be used as a scientific tool, hence it is important to make sure there is no bias accidentally caused by the randomness in generating data or initial weights of neural network models. This concern can be addressed by using an ensemble of neural network models, rather than a single model. The number of models in the ensemble need not be large, since they increase computational effort without necessarily improving performance.

An ensemble of 5 neural networks was hence used, each trained independently on a different dataset. Utilizing the stochastic property of the data generator, five distinct *Mega* datasets (1,000,000 rows) were generated. Each of these datasets was used to train a different neural network with the same 11-64-32-1 architecture mentioned earlier; results are shown in Fig. 4.

When predicting a value, predictions from all the 5 neural networks are aggregated, to provide a more robust estimate of the actual value. To understand the performance of the different aggregation strategies, we make use of the ”Magna” dataset (10,000,000 rows). The mean, median, and standard deviation were computed for the predictions from the 5 neural network models.

TABLE VI
VALIDATING PERFORMANCE OF MODEL ENSEMBLE; USING MEAN AGGREGATION

Validation set	MAE	R^2 Score	Binary acc.
Magna 1	0.00691	0.961	0.981
Magna 2	0.00689	0.964	0.977

TABLE VII
VALIDATING PERFORMANCE OF MODEL ENSEMBLE; USING MEDIAN AGGREGATION

Validation set	MAE	R^2 Score	Binary acc.
Magna 1	0.00436	0.978	0.986
Magna 2	0.00436	0.979	0.986

Tables VI and VII show that using the median (for aggregating results from the five neural networks in the ensemble) was a little better than using the mean, with a slight improvement over the single model performance. Thus the ensemble approach is more robust than using a single neural network model, and results in improved performance. The ensemble (using the median) achieves a mean absolute error of 0.004, R^2 score of 0.978, and a binary accuracy of 0.986, over the two ”Magna” datasets.

Using the ensemble approach enables the use of a user-defined threshold that can be configured, analogous to other scientific tools. If the 5 neural networks predict almost identical values, we have higher confidence in their accuracy,

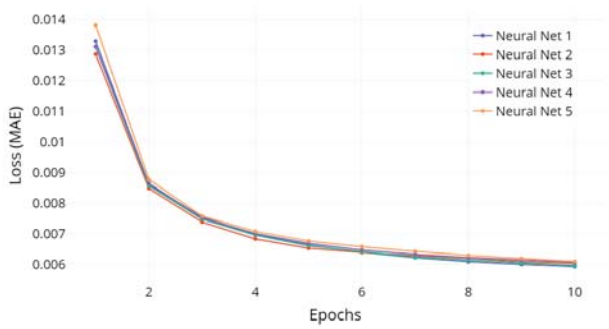

Fig. 4 Training loss curves for an ensemble of 5 neural networks

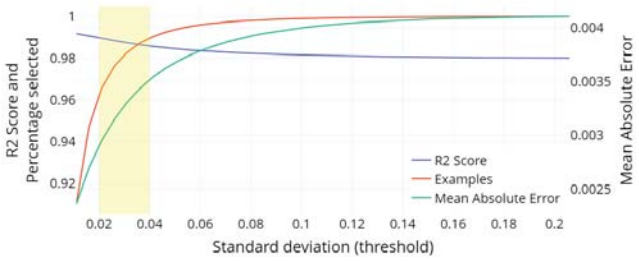

Fig. 5 Threshold analysis using the ensemble of 5 neural networks

when compared to ensemble predictions with a larger standard deviation.

Fig. 5 suggests that a threshold value in the range (0.02, 0.04] covers a large portion of the examples while maintaining a low mean absolute error and a high R^2 score. Furthermore, the threshold can be adjusted based on the type and sensitivity of the experiment. The samples that fall outside the threshold can be addressed using the mathematical approach with the machine learning-based approximation. Using this approach will be more efficient than the approach currently used, with considerable savings in computational resources and time required, despite the simplicity of the machine learning models proposed in this work.

VI. Conclusion

Summarizing, this work has proposed a moments-based approach for the calculation of electron-repulsion integrals, with an approximation using a finite linear combination of moments. A random forest approach successfully reduced the feature set size significantly (from 54 to 6), and performed very well for predicting the signs of coefficients (indicating whether the forces are attractive or repulsive). A neural network approach resulted in further improvement in estimating coefficient magnitudes using 2-hidden layer feedforward models, and a simple feature masking approach reduced the feature set size from 54 to 11, and considerable overlap with the 6 features obtained from the random forest approach was observed. Finally, the best results were obtained using an ensemble of 5 independently trained neural networks whose results were combined using a median rule.

The results from this approach will result in a reduction in the overall computational cost of performing quantum chemical calculations. In future studies, the new approach will be used to investigation optical and electronic properties of large nanoparticles which are currently inaccessible because of prohibitively high computational cost.

References

[1] B. Peng and K. Kowalski, "Highly efficient and scalable compound decomposition of two-electron integral tensor and its application in coupled cluster calculations," *Journal of chemical theory and computation*, vol. 13, no. 9, pp. 4179–4192, 2017.

[2] F. Weigend, "A fully direct ri-hf algorithm: Implementation, optimised auxiliary basis sets, demonstration of accuracy and efficiency," *Physical Chemistry Chemical Physics*, vol. 4, no. 18, pp. 4285–4291, 2002.

[3] H. Koch, A. Sánchez de Merás, and T. B. Pedersen, "Reduced scaling in electronic structure calculations using cholesky decompositions," *The Journal of chemical physics*, vol. 118, no. 21, pp. 9481–9484, 2003.

[4] J. Jackson, *Classical Electrodynamics*. Wiley, 2021. [Online]. Available: https://books.google.com/books?id=6VV-EAAAQBAJ

[5] B. Nagy and F. Jensen, "Basis sets in quantum chemistry," *Reviews in Computational Chemistry*, vol. 30, pp. 93–149, 2017.

[6] A. Szabo and N. S. Ostlund, *Modern quantum chemistry: introduction to advanced electronic structure theory*. Courier Corporation, 2012.

[7] T. Helgaker, P. Jorgensen, and J. Olsen, *Molecular electronic-structure theory*. John Wiley & Sons, 2013.

[8] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[9] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1–3, p. 389–422, mar 2002. [Online]. Available: https://doi.org/10.1023/A:1012487302797

[10] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[11] M. Innes, E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, T. Karmali, A. Pal, and V. Shah, "Fashionable modelling with flux," *CoRR*, vol. abs/1811.01457, 2018. [Online]. Available: https://arxiv.org/abs/1811.01457

[12] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *IEEE Potentials*, vol. 13, no. 4, pp. 27–31, 1994.

Nishant Rodrgues has completed his MS in Computer Science at Syracuse University (May 2023), and is currently employed as a Data Scientist. He had previously obtained a bachelor's degree from Vellore Institute of Technology, India. His research interests include artificial intelligence, machine learning and optimization algorithms.

Nicole Spanedda is a PhD candidate in Arindam Chakraborty's research group in the Chemistry department at Syracuse University. She has completed her MS in Chemistry at Syracuse University and completed her Bachelor of Science in Chemistry at Fairleigh Dickinson University in New Jersey.

Chilukuri Mohan holds a Ph.D. in Computer Science from SUNY at Stony Brook. He has been teaching at Syracuse University since 1988, where he is a Professor in the Department of Electrical Engineering and Computer Science, and has served as Department Chair; he has also served as Interim Dean of the College of Engineering and Computer Science. He has authored/co-authored over 230 papers and three books: Elements of Artificial Neural Networks (MIT Press, 1997), Frontiers of Expert Systems: Reasoning with limited knowledge (Kluwer, 2000), and Anomaly Detection Principles and Algorithms (Springer, 2017). His research has included the areas of automated reasoning, neural networks, evolutionary algorithms, data mining, social network dynamics, and bioinformatics. He serves on the editorial boards of three journals. In 2019, he received the IEEE Region 1 Technological Innovation (Academic) Award, for development of novel algorithms in computational intelligence.

Arindam Chakraborty is an Professor of Chemistry at Syracuse University. He completed his Ph. D. from University of Minnesota and did his postdoctoral research at Pennsylvania State University. He joined the Chemistry Department at Syracuse University in 2009 and his research interests is in theoretical and computational chemistry with emphasis on development of novel quantum mechanical technique for investigating optical and electronic properties of nanomaterials.