Multi-task learning for electronic structure to predict and explore molecular potential energy surfaces

Zhuoran Qiao
California Institute of Technology
Pasadena, CA 91125
zqiao@caltech.edu

Feizhi Ding
Entos, Inc.
Los Angeles, CA 90027
feizhi@entos.ai

Matthew Welborn
Entos, Inc.
Los Angeles, CA 90027
matt@entos.ai

Peter J. Bygrave
Entos, Inc.
Los Angeles, CA 90027
peter@entos.ai

Daniel G. A. Smith
Entos, Inc.
Los Angeles, CA 90027
daniel@entos.ai

Animashree Anandkumar
California Institute of Technology
Pasadena, CA 91125
NVIDIA
Santa Clara, CA 95051
anima@caltech.edu

Frederick R. Manby
Entos, Inc.
Los Angeles, CA 90027
fred@entos.ai

Thomas F. Miller III
California Institute of Technology
Pasadena, CA 91125
Entos, Inc.
Los Angeles, CA 90027
tfm@caltech.edu, tom@entos.ai

Abstract

We refine the OrbNet model to accurately predict energy, forces, and other response properties for molecules using a graph neural-network architecture based on features from low-cost approximated quantum operators in the symmetry-adapted atomic orbital basis. The model is end-to-end differentiable due to the derivation of analytic gradients for all electronic structure terms, and is shown to be transferable across chemical space due to the use of domain-specific features. The learning efficiency is improved by incorporating physically motivated constraints on the electronic structure through multi-task learning. The model outperforms existing methods on energy prediction tasks for the QM9 dataset and for molecular geometry optimizations on conformer datasets, at a computational cost that is thousand-fold or more reduced compared to conventional quantum-chemistry calculations (such as density functional theory) that offer similar accuracy.

1 Introduction

Quantum chemistry calculations - most commonly those obtained using density functional theory (DFT) - provide a level of accuracy that is important for many chemical applications but at a computational cost that is often prohibitive. As a result, machine-learning efforts have focused on the prediction of molecular potential energy surfaces, using both physically motivated features [1, 2, 3, 4, 5, 6] and neural-network-based representation learning [7, 8, 9, 10, 11, 12, 13]. Despite the success of such methods in predicting energies on various benchmarks, the generalizability of deep

Machine Learning for Molecules Workshop at NeurIPS 2020. https://ml4molecules.github.io

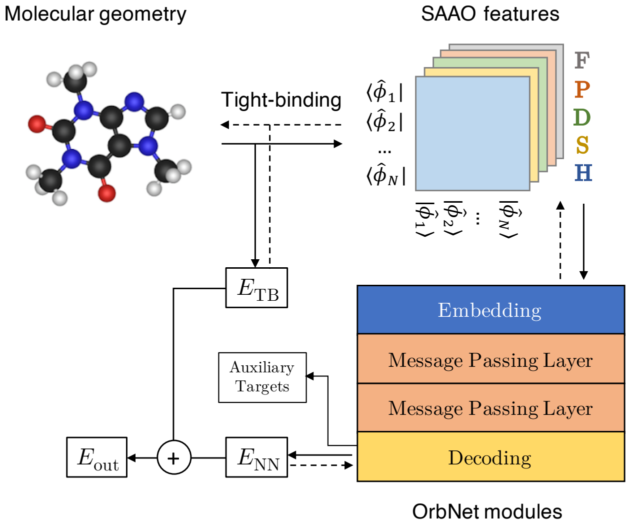


Figure 1: Schematic of the employed OrbNet deep-learning approach. A low-cost extended tight-binding calculation is performed on the molecular system, generating the approximate energy E_{TB} and the SAAO feature matrices. The SAAO feature matrices are passed to the OrbNet deep neural network modules, yielding the machine-learned correction to the total energy, E_{NN}, as well as auxiliary atom-specific target properties. Dashed arrows indicate components that must be analytically differentiated for the gradient calculation. Additional details are provided in Appendix B.

neural network models across chemical space and for out-of-equilibrium geometries is less investigated.

In this work, we demonstrate an approach using features from a low-cost electronic-structure calculation in the basis of symmetry-adapted atomic orbitals (SAAOs) with a deep neural network architecture (OrbNet). The model has previously been shown to predict the molecular energies with DFT accuracy for both chemical and conformational degrees of freedom, even when applied to systems significantly larger than the training molecules [14]. To improve learning efficiency, we introduce a multi-task learning strategy in which OrbNet is trained with respect to both molecular energies and other computed properties of the quantum mechanical wavefunction. Furthermore, we introduce and numerically demonstrate the analytical gradient theory for OrbNet, which is essential for the calculation of inter-atomic forces and other response properties, such as dipoles and linear-response excited states.

2 Method

2.1 OrbNet: Neural message passing on SAAOs with atomic and global attention

In this work, the molecular system is encoded as graph-structured data with features obtained from a low-cost tight-binding calculation, following Qiao et al [14]. We employ features obtained from matrix elements of approximated quantum operators of an extended tight-binding method (GFN-xTB[15]), evaluated in the symmetry-adapted atomic orbital (SAAO) basis. Specifically, the Fock ($\mathbf{F}$), density ($\mathbf{P}$), orbital centroid distances ($\mathbf{D}$), core Hamiltonian ($\mathbf{H}$), and overlap ($\mathbf{S}$) matrices are used as the input features, with node features corresponding to diagonal SAAO matrix elements $X_u = [F_{uu}, P_{uu}, H_{uu}]$ and edge features corresponding to off-diagonal SAAO matrix elements $X^{\text{e}}_{uv} = [F_{uv}, D_{uv}, P_{uv}, S_{uv}, H_{uv}]$. Fig. 1 summarizes the deep-learning approach, and additional details are provided in Appendix B.

The feature embedding and neural message-passing mechanism employed for the node and edge attributes is largely unchanged from Ref. [14]. However, to enable multi-task learning and to improve the learning capacity of the model, we introduce atom-specific attributes, $\mathbf{f}^l_A$, and global molecule-level attributes, $\mathbf{q}^l$, where l is the message passing layer index and A is the atom index. The whole-molecule and atom-specific attributes allow for the prediction of auxiliary targets (Fig. 1) through multi-task learning, thereby providing physically motivated constraints on the electronic structure of the molecule that can be used to refine the representation at the SAAO level.

For the prediction of both the electronic energies and the auxiliary targets, only the final atom-specific attributes, $\mathbf{f}_A^L$, are employed, since they self-consistently incorporate the effect of the whole-molecule and node- and edge-specific attributes. The electronic energy is obtained by combining the approximate energy E_{TB} from the extended tight-binding calculation and the model output E_{NN}, the latter of which is a one-body sum over atomic contributions; the atom-specific auxiliary targets $\mathbf{d}_A$ are predicted from the same attributes.

$$\hat{E}_{\text{out}} = E_{\mathrm{TB}} + E_{\mathrm{NN}} = E_{\mathrm{TB}} + \sum_A \epsilon_A = E_{\mathrm{TB}} + \sum_A [\mathrm{Dec}(\mathbf{f}_A^L) + E_A^{\mathrm{c}}] \tag{1}$$

$$\hat{\mathbf{d}}_A = \mathrm{Dec}^{\mathrm{aux}}(\mathbf{f}_A^L) \tag{2}$$

Here, the energy decoder Dec and the auxiliary-target decoder $\mathrm{Dec}^{\mathrm{aux}}$ are residual neural networks [16] built with fully connected and normalization layers, and E_A^{c} are element-specific, constant shift parameters for the isolated-atom contributions to the total energy.

The GradNorm algorithm [17] is used to adaptively adjust the weight of the auxiliary target loss based on the gradients of the last fully-connected layer before the decoding networks.

2.2 End-to-end differentiability: Analytic gradients

The model is constructed to be end-to-end differentiable by employing input features (i.e., the SAAO matrix elements) that are smooth functions of both atomic coordinates and external fields. We derive the analytic gradients of the total energy E^{out} with respect to the atom coordinates, and we employ local energy minimization with respect to molecular structure as an exemplary task to demonstrate the quality of the learned potential energy surface (Section 3.2).

Using a Lagrangian formalism [18, 19], the analytic gradient of the predicted energy with respect to an atom coordinate x can be expressed in terms of contributions from the tight-binding model, the neural network, and additional constraint terms:

$$\frac{dE_{\text{out}}}{dx} = \frac{dE_{\mathrm{TB}}}{dx} + \sum_{\mathbf{f}\in\{\ \mathbf{F},\mathbf{D},\mathbf{P},\mathbf{S},\mathbf{H}\ \}} \mathrm{Tr}\left[\frac{\partial E_{\mathrm{NN}}}{\partial \mathbf{f}}\frac{\partial \mathbf{f}}{\partial x}\right] + \mathrm{Tr}[\mathbf{W}\frac{\partial \mathbf{S}^{\mathrm{AO}}}{\partial x}] + \mathrm{Tr}[\mathbf{z}\frac{\partial \mathbf{F}^{\mathrm{AO}}}{\partial x}]. \tag{3}$$

Here, the third and fourth terms on the right-hand side are gradient contributions from the orbital orthogonality constraint and the Brillouin condition, respectively, where $\mathbf{F}^{\mathrm{AO}}$ and $\mathbf{S}^{\mathrm{AO}}$ are the Fock matrix and orbital overlap matrix in the atomic orbital (AO) basis. Detailed expressions for $\frac{\partial \mathbf{f}}{\partial x}$, $\mathbf{W}$, and $\mathbf{z}$ are provided in Appendix D. The tight-binding gradient $\frac{dE_{\mathrm{TB}}}{dx}$ for the GFN-xTB model has been previously reported [15], and the neural network gradients with respect to the input features $\frac{\partial E_{\mathrm{NN}}}{\partial \mathbf{f}}$ are obtained using reverse-mode automatic differentiation [20].

2.3 Auxiliary targets from density matrix projection

The utility of graph- and atom-level auxiliary tasks to improve the generalizability of the learned representations for molecules has been highlighted for learning molecular properties in the context of graph pre-training [21, 22] and multi-task learning [23]. Here, we employ multi-task learning with respect to the total molecular energy and atom-specific auxiliary targets. The atom-specific targets that we employ are similar to the features introduced in the DeePHF model [24], obtained by projecting the density matrix into a basis set that does not depend upon the identity of the atomic element,

$$\mathbf{d}_{nl}^A = [\mathrm{EigenVals}_{m,m'}([\ ^{\mathrm{O}}\mathcal{D}_{nl}^A]_{m,m'})||\mathrm{EigenVals}_{m,m'}([\ ^{\mathrm{V}}\mathcal{D}_{nl}^A]_{m,m'})]. \tag{4}$$

Here, the projected density matrix is given by $[\ ^{\mathrm{O}}\mathcal{D}_{nl}^A]_{m,m'} = \sum_{i\in\mathrm{occ}}\langle\alpha_{nlm}^A|\psi_i\rangle\langle\psi_i|\alpha_{nlm'}^A\rangle$, and the projected valence-occupied density matrix is given by $[\ ^{\mathrm{V}}\mathcal{D}_{nl}^A]_{m,m'} = \sum_{j\in\mathrm{valocc}}\langle\alpha_{nlm}^A|\psi_j\rangle\langle\psi_j|\alpha_{nlm'}^A\rangle$, where $|\psi_{\{i,j\}}\rangle$ are molecular orbitals from the reference DFT calculation, $|\alpha_{nlm}^A\rangle$ is a basis function centered at atom A with radial index n and spherical-harmonic degree l and order m. The indices i and j runs over all occupied orbitals and valence-occupied orbital indices, respectively, and $||$ denotes a vector concatenation operation. The auxiliary target vector $\mathbf{d}_A$ for each atom A in the molecule is obtained by concatenating $\mathbf{d}^A_{nl}$ for all n and l. The parameters for the projection basis $|\alpha^A_{nlm}\rangle$ are described in Appendix E. Additional attributes, such as such as partial charges and reactivities, could also be naturally included within this framework.

3 Results

We present results for molecular energy prediction and geometry optimization tasks. All models are produced using the same set of hyperparameters and the training procedure in Appendix C.

3.1 QM9 formation energy

We begin with a standard benchmark test of predicting molecular energies for the QM9 dataset, which consists of 133,885 organic molecules with up to nine heavy atoms at locally optimized geometries. Table 1 presents results from current work, as well as previously published results using SchNet [9], PhysNet [11], DimeNet [13], DeepMoleNet [23], and OrbNet [14]. The approach proposed in this work significantly outperforms existing methods in terms of both data efficiency and prediction accuracy in this dataset. In particular, it is seen that the use of multi-task learning in the current study leads to significant improvement over the previously published OrbNet results, which already exhibited the smallest errors among published methods.

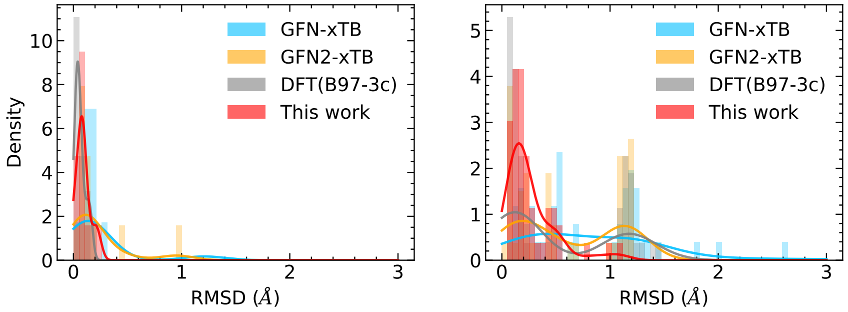
Training size	SchNet	PhysNet	DimeNet	DeepMoleNet	OrbNet	This work
25,000	-	-	-	-	11.6	**8.08**
50,000	15	13	-	-	8.22	**5.89**
110,000	14	8.2	8.02	6.1	5.01	**3.87**

Table 1: MAEs (reported in meV) for predicting the QM9 dataset of total molecular energies. The employed labels are the published values [25] calculated at the B3LYP/6-31G(2df,p) level of theory.

3.2 Molecular geometry optimizations

A practical application of energy gradient (i.e., force) calculations is to optimize molecule structures by locally minimizing the energy. Here, we use this application as a test of the accuracy of the OrbNet potential energy surface in comparison to other widely used methods of comparable and greater computational cost. Test are performed for the ROT34 [26] and MCONF [27] datasets, with initial structures that are locally optimized at the high-quality level of ωB97X-D3/Def2-TZVP DFT with tight convergence parameters. ROT34 includes conformers of 12 small organic molecules with up to 13 heavy atoms; MCONF includes 52 conformers of the melatonin molecule which has 17 heavy atoms. From these initial structures, we performed a local geometry optimization using the various energy methods, including OrbNet from the current work, the GFN semi-empirical methods [15, 28], and the relatively low-cost DFT functional B97-3c [29]. The error in the resulting structure with respect to the reference structures optimized at the ωB97X-D3/Def2-TZVP level was computed as root mean squared distance (RMSD) following optimal molecular alignment. This test investigates whether the potential energy landscape for each method is locally consistent with a high-quality DFT description.

Fig. 2 presents the resulting distribution of errors for the various methods over each dataset, with results summarized in the accompanying table. It is clear that while the GFN semi-empirical methods provide a computational cost that is comparable to OrbNet, the resulting geometry optimizations are substantially less accurate, with a significant (and in some cases very large) fraction of the local geometry optimizations relaxing into structures that are inconsistent with the optimized reference DFT structures (i.e., with RMSD in excess of 0.6 Angstrom). In comparison to DFT using the B97-3c functional, OrbNet provides optimized structures that are of comparable accuracy for ROT34 and that are significantly more accurate for MCONF; this should be viewed in light of the fact that OrbNet is over 100-fold less computationally costly. On the whole, OrbNet is the best approximation to the reference DFT results, at a computational cost that is over 1,000-fold reduced.



Method	Mean RMSD (Å)		Incorrect geometries		Time/step
	ROT34	MCONF	ROT34	MCONF	MCONF
GFN-xTB	0.23	0.90	8%	52%	< 1 s
GFN2-xTB	0.21	0.60	8%	44%	< 1 s
DFT (B97-3c)	**0.06**	0.51	**0%**	37%	> 100 s
This work	0.09	**0.26**	**0%**	**6%**	< 1 s
Ref. DFT (ωB97X-D3)	-	-	-	-	> 1,000 s

Figure 2: The molecular geometry optimization accuracy for the ROT34 (left) and MCONF (right) datasets, reported as the best-alignment root-mean-square-deviation (RMSD) compared to the reference DFT geometries at the ωB97X-D3/Def2-TZVP level. The distribution of errors are plotted as histograms (with overlaying kernel density estimations), and the table reports the mean errors and the percentage of optimized structures that correspond to incorrect geometries (i.e., RMSD $>$ 0.6 Angstrom). Timings correspond to the average cost for a single force evaluation for the MCONF dataset on a single Intel Xeon Gold 6130 @ 2.10GHz CPU core.

4 Conclusions

We extend the OrbNet deep-learning model through the use of multi-task learning and the development of the analytical gradient theory for calculating molecular forces and other response properties. It is shown that multi-task learning leads to improved data efficiency, with OrbNet providing lower errors than previously reported deep-learning methods for the QM9 formation energy prediction task. Moreover, it is shown that geometry optimizations on the OrbNet potentially energy surface provide accuracy that is significantly greater than that available from semi-empirical methods and that even outperform fully quantum mechanical DFT descriptions that are vastly more computationally costly. The method is immediately applicable to other down-stream tasks.

Acknowledgement

Z.Q. acknowledges the graduate research funding from Caltech. T.F.M. and A.A. acknowledge partial support from the Caltech DeLogi fund, and A.A. acknowledges support from a Caltech Bren professorship. The authors gratefully acknowledge NVIDIA, including Abe Stern and Tom Gibbs, for helpful discussions regarding GPU implementations of graph neural networks.

References

[1] Albert P Bartók et al. “Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons”. In: *Physical review letters* 104.13 (2010), p. 136403.

[2] Raghunathan Ramakrishnan et al. “Big data meets quantum chemistry approximations: the Δ-machine learning approach”. In: *J. Chem. Theory Comput.* 11.5 (2015), p. 2087.

[3] Thuong T. Nguyen et al. “Comparison of permutationally invariant polynomials, neural networks, and Gaussian approximation potentials in representing water interactions through many-body expansions”. In: *J. Chem. Phys.* 148.24 (2018), p. 241725.

[4] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. “ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost”. In: *Chemical science* 8.4 (2017), pp. 3192–3203.

[5] Justin S Smith et al. “Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning”. In: *Nat. Commun.* 10.1 (2019), pp. 1–8.

[6] Lixue Cheng et al. “A universal density matrix functional from molecular orbital-based machine learning: Transferability across organic molecules”. In: *J. Chem. Phys.* 150.13 (2019), p. 131103.

[7] Petar Veličković et al. “Graph Attention Networks”. In: *International Conference on Learning Representations*. 2018.

[8] Kevin Yang et al. “Analyzing learned molecular representations for property prediction”. In: *J. Chem. Inf. Model.* 59.8 (2019), pp. 3370–3388.

[9] Kristof Schütt et al. “Schnet: A continuous-filter convolutional neural network for modeling quantum interactions”. In: *Advances in neural information processing systems*. 2017, pp. 991–1001.

[10] Justin Gilmer et al. “Neural message passing for Quantum chemistry”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 2017, pp. 1263–1272.

[11] Oliver T Unke and Markus Meuwly. “PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges”. In: *J. Chem. Theory Comput.* 15.6 (2019), pp. 3678–3693.

[12] Brandon Anderson, Truong Son Hy, and Risi Kondor. “Cormorant: Covariant Molecular Neural Networks”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 14537–14546. URL: `http://papers.nips.cc/paper/9596-cormorant-covariant-molecular-neural-networks.pdf`.

[13] Johannes Klicpera, Janek Groß, and Stephan Günnemann. “Directional Message Passing for Molecular Graphs”. In: *International Conference on Learning Representations*. 2019.

[14] Zhuoran Qiao et al. “OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features”. In: *The Journal of Chemical Physics* 153.12 (2020), p. 124111.

[15] Stefan Grimme, Christoph Bannwarth, and Philip Shushkov. “A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (Z=1–86)”. In: *J. Chem. Theory Comput.* 13.5 (2017), pp. 1989–2009.

[16] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[17] Zhao Chen et al. “Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 794–803.

[18] Sebastian JR Lee et al. “Analytical gradients for projection-based wavefunction-in-DFT embedding”. In: *The Journal of Chemical Physics* 151.6 (2019), p. 064112.

[19] Martin Schütz et al. “Analytical energy gradients for local second-order Møller–Plesset perturbation theory using density fitting approximations”. In: *The Journal of chemical physics* 121.2 (2004), pp. 737–750.

[20] Adam Paszke et al. “Automatic Differentiation in PyTorch”. In: *NIPS 2017 Workshop on Autodiff*. Long Beach, California, USA, 2017. URL: `https://openreview.net/forum?id=BJJsrmfCZ`.

[21] Weihua Hu et al. “Strategies for Pre-training Graph Neural Networks”. In: *International Conference on Learning Representations*. 2019.

[22] Garrett B Goh et al. “Using rule-based labels for weak supervised learning: a ChemNet for transferable chemical property prediction”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 302–310.

[23] Ziteng Liu et al. “Transferable multi-level attention neural network for accurate prediction of quantum chemistry properties via multi-task learning”. In: *ChemRxiv* 12588170 (2020), p. v1.

[24] Yixiao Chen et al. “Ground State Energy Functional with Hartree–Fock Efficiency and Chemical Accuracy”. In: *The Journal of Physical Chemistry A* 124.35 (2020), pp. 7155–7165.

[25] Raghunathan Ramakrishnan et al. “Quantum chemistry structures and properties of 134 kilo molecules”. In: *Sci. Data* 1.1 (2014), pp. 1–7.

[26] Tobias Risthaus, Marc Steinmetz, and Stefan Grimme. “Implementation of nuclear gradients of range-separated hybrid density functionals and benchmarking on rotational constants for organic molecules”. In: *Journal of Computational Chemistry* 35.20 (2014), pp. 1509–1516.

[27] Uma R Fogueri et al. “The melatonin conformer space: Benchmark and assessment of wave function and DFT methods for a paradigmatic biological and pharmacological molecule”. In: *The Journal of Physical Chemistry A* 117.10 (2013), pp. 2269–2277.

[28] Christoph Bannwarth, Sebastian Ehlert, and Stefan Grimme. “GFN2-xTB — An accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions”. In: *J. Chem. Theory Comput.* 15.3 (2019), pp. 1652–1671.

[29] Jan Gerit Brandenburg et al. “B97-3c: A revised low-cost variant of the B97-D density functional method”. In: *The Journal of chemical physics* 148.6 (2018), p. 064104.

Appendices

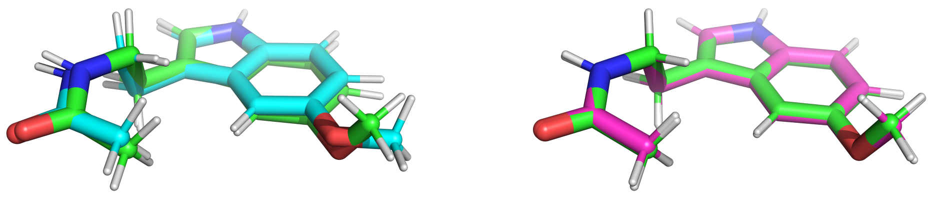
A Dataset and computational details

For results reported in Section 3.1, we employ the QM9 dataset[1] with pre-computed DFT labels. From this dataset, 3054 molecules were excluded as recommended in Ref. [1]; we sample 110000 molecules for training and 10831 molecules for testing. The training sets of 25000 and 50000 molecules are subsampled from the 110000-molecule dataset.

To train the model reported in Section 3.2, we employ the published geometries from Ref. [2], which include optimized and thermalized geometries of molecules up to 30 heavy atoms from the QM7b-T, QM9, GDB13-T, and DrugBank-T datasets. We perform model training using the dataset splits of Model 3 in Ref. [2]. DFT labels are computed using the ωB97X-D3 functional [3] with a Def2-TZVP AO basis set[4] and using density fitting[5] for both the Coulomb and exchange integrals using the Def2-Universal-JKFIT basis set.[6]

For results reported in Section 3.2, we perform geometry optimization for the DFT, OrbNet, and GFN-xTB calculations by minimizing the potential energy using the BFGS algorithm with the Translation-rotation coordinates (TRIC) of Wang and Song[7]; geometry optimizations for GFN2-xTB are performed using the default algorithm in the XTB package [8]. All local geometry optimizations are initialized from pre-optimized structures at the ωB97X-D3/Def2-TZVP level of theory. For the B97-3c method, the mTZVP basis set[9] is employed.

All DFT and GFN-xTB calculations are performed using ENTOS QCORE [10]; GFN2-xTB calculation are performed using XTB package [8].



(a) GFN2-xTB v.s. reference, RMSD=1.2Å

(b) OrbNet v.s. reference, RMSD=0.14Å

Figure 3: Comparison of optimized structure of the conformer MCONF/41 from GFN2-xTB (cyan), OrbNet (magenta) and the reference DFT structure (green).

B Specification of OrbNet embedding, message passing & pooling, and decoding layers

We employ the feature embedding scheme introduced in OrbNet [2] where the SAAO feature matrices are transformed by radial basis functions,

$$\mathbf{h}_u^{\mathrm{RBF}} = [\phi_1^{\mathrm{h}}(\tilde{X}_u), \phi_2^{\mathrm{h}}(\tilde{X}_u), ..., \phi_{n_{\mathrm{r}}}^{\mathrm{h}}(\tilde{X}_u)] \tag{5}$$

$$\mathbf{e}_{uv}^{\mathrm{RBF}} = [\phi_1^{\mathrm{e}}(\tilde{X}_{uv}^{\mathrm{e}}), \phi_2^{\mathrm{e}}(\tilde{X}_{uv}^{\mathrm{e}}), ..., \phi_{m_{\mathrm{r}}}^{\mathrm{e}}(\tilde{X}_{uv}^{\mathrm{e}})], \tag{6}$$

where $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{X}}^{\mathrm{e}}$ are pre-normalized SAAO feature matrices, $\phi_n^{\mathrm{h}}(r) = \sin(\pi n r)$ is a sine function used for node (SAAO) embedding; to improve the smoothess of the potential energy surface, we used the real Morlet wavelet functions for edge embedding:

$$\phi_m^{\mathrm{e}}(r) = \exp(-(\frac{r}{\sigma \cdot c_{\mathbf{X}}})^2) \cdot \sin(\pi m r / c_{\mathbf{X}}) \tag{7}$$

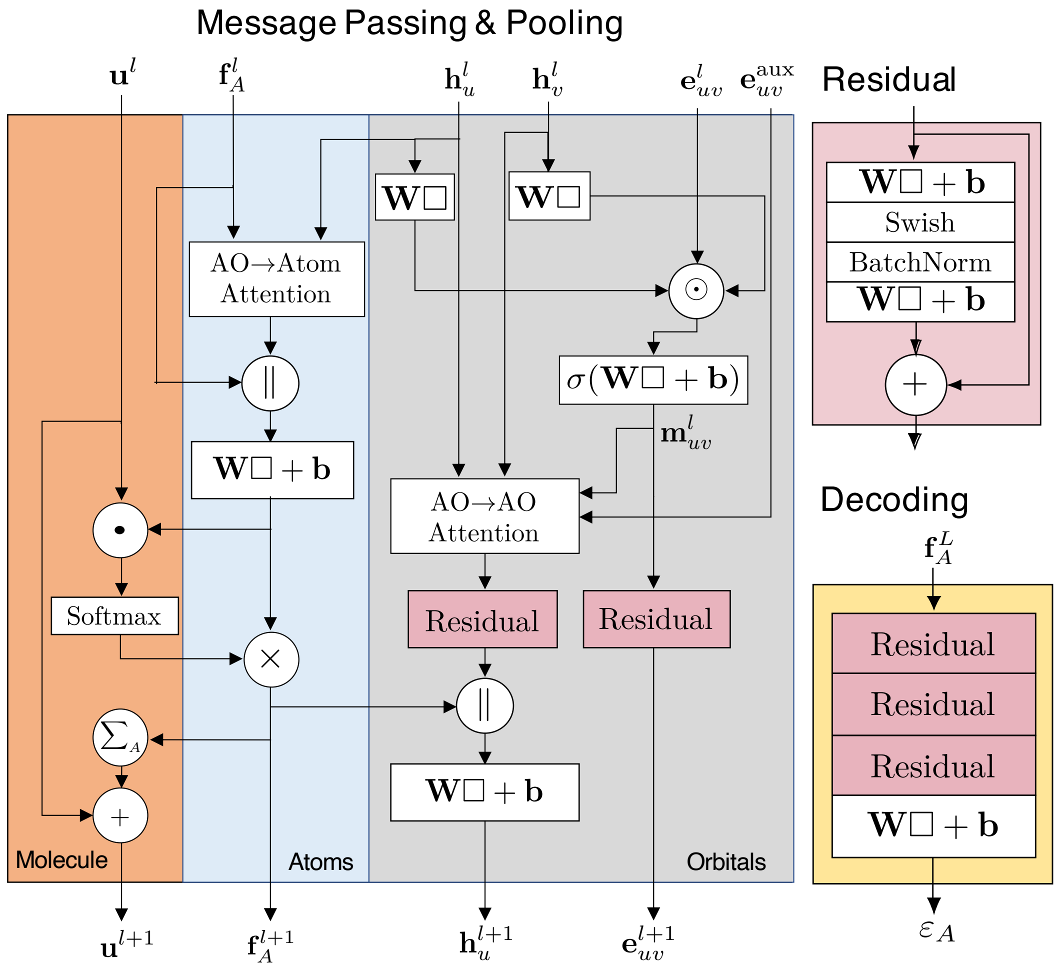


Figure 4: Detail of a single message-passing and pooling layer ("Message Passing Layer" in Fig. 1), and a decoding network ("Decoding" in Fig. 1). At message passing and pooling layer $l+1$, the whole-molecule, atom-specific, node-specific, and edge-specific attributes are updated. The atom-specific attributes $\mathbf{f}_A^l$ are updated with input from node- and edge-specific attributes $\mathbf{h}_u^l$ and $\mathbf{e}_{uv}^l$ and likewise includes the back-propagation from the whole-molecule attributes; finally, the whole-molecule attributes $\mathbf{q}^l$ are updated with input from the atom-specific attributes. The final atom-specific attributes are passed into separate decoding networks to generate the energy prediction and auxiliary target predictions. A decoding network is composed of multiple residual blocks ("Residual") and a linear output layer, as illustrated above.

and $c_{\mathbf{X}}$ ($\mathbf{X} \in \{\mathbf{F}, \mathbf{D}, \mathbf{P}, \mathbf{S}, \mathbf{H}\}$) is the operator-specific upper cutoff value to $\tilde{X}^{\mathrm{e}}_{uv}$. To ensure size-consistency for energy predictions, a mollifier $I_{\mathbf{X}}(r)$ with the auxiliary edge attribute $\mathbf{e}^{\mathrm{aux}}_{uv}$ is introduced:

$$\mathbf{e}^{\mathrm{aux}}_{uv} = \mathbf{W}^{\mathrm{aux}} \cdot I_{\mathbf{X}}(\tilde{X}^{\mathrm{e}}_{uv}), \tag{8}$$

where

$$I_{\mathbf{X}}(r) = \begin{cases} \exp\left(\frac{c_{\mathbf{X}}}{|r| - c_{\mathbf{X}}} + 1\right) \cdot \exp(-(\frac{r}{\sigma \cdot c_{\mathbf{X}}})^2) & \text{if } 0 \leq |r| < c_{\mathbf{X}} \\ 0 \quad \text{if } |r| \geq c_{\mathbf{X}} \end{cases} \tag{9}$$

The radial basis function embeddings of the SAAOs and a one-hot encoding of the chemical element of the atoms ($\mathbf{f}^{\mathrm{onehot}}_A$) are transformed by neural network modules to yield 0-th order SAAO, SAAO-pair, and atom attributes,

$$\mathbf{h}^0_u = \mathrm{Enc}_{\mathrm{h}}(\mathbf{h}^{\mathrm{RBF}}_u),\ \mathbf{e}^0_{uv} = \mathrm{Enc}_{\mathrm{e}}(\mathbf{e}^{\mathrm{RBF}}_{uv}),\ \mathbf{f}^0_A = \mathrm{Enc}_{\mathrm{f}}(\mathbf{f}^{\mathrm{onehot}}_A) \tag{10}$$

where $\mathrm{Enc}_{\mathrm{h}}$ and $\mathrm{Enc}_{\mathrm{e}}$ are residual blocks[11] comprising 3 dense NN layers, and $\mathrm{Enc}_{\mathrm{f}}$ is a single dense NN layer. In contrast to atom-based message passing neural networks, this additional embedding transformation captures the interactions among the physical operators.

The update of the node- and edge-specific attributes (gray block in Fig. 4) is unchanged from Ref. [2], except with the additional information back-propagation from the atom-specific attributes. The node and edge attributes at step $l + 1$ are updated via the following neural message passing mechanism (corresponding to "AO-AO attention" in Fig. 4):

$$\tilde{\mathbf{h}}^{l+1}_u = \mathbf{h}^l_u + \mathbf{W}^l_{\mathrm{h},2} \cdot \mathrm{Swish}\Big(\mathrm{BatchNorm}\big(\mathbf{W}^l_{\mathrm{h},1} \cdot [\bigoplus_i (\sum_{v \in N(u)} w^{l,i}_{uv} \cdot \mathbf{m}^l_{uv})] + \mathbf{b}^l_{\mathrm{h},1}\big)\Big) + \mathbf{b}^l_{\mathrm{h},2} \tag{11a}$$

$$\mathbf{m}^l_{uv} = \mathrm{Swish}(\mathbf{W}^l_{\mathrm{m}} \cdot [\mathbf{h}^l_u \odot \mathbf{h}^l_v \odot \mathbf{e}^l_{uv}] + \mathbf{b}^l_{\mathrm{m}}) \tag{11b}$$

$$w^{l,i}_{uv} = \mathrm{Tanh}(\sum [(\mathbf{W}^{l,i}_{\mathrm{a}} \cdot \mathbf{h}^l_u) \odot (\mathbf{W}^{l,i}_{\mathrm{a}} \cdot \mathbf{h}^l_v) \odot \mathbf{e}^l_{uv} \odot \mathbf{e}^{\mathrm{aux}}_{uv}]/n_{\mathrm{e}}) \tag{11c}$$

$$\mathbf{e}^{l+1}_{uv} = \mathbf{e}^l_{uv} + \mathbf{W}^l_{\mathrm{e},2} \cdot \big(\mathrm{Swish}(\mathbf{W}^l_{\mathrm{e},1} \cdot \mathbf{m}^l_{uv} + \mathbf{b}^l_{\mathrm{e},1})\big) + \mathbf{b}^l_{\mathrm{e},2} \tag{11d}$$

where $\mathbf{m}^l_{uv}$ is the message function on each edge, $w^{l,i}_{uv}$, are multi-head attention scores [12] for the relative importance of SAAO pairs (i indexes attention heads), $\bigoplus$ denotes a vector concatenation operation, $\odot$ denotes the Hadamard product, and $\cdot$ denotes the matrix-vector product.

The SAAO attributes are accumulated into the atoms on which the corresponding SAAOs are centered, using an attention-based pooling operation ("AO-Atom attention" in Fig. 4) inspired by the set transformer [13] architecture:

$$a^l_{A,u} = \mathrm{Softmax}(\mathbf{f}^l_A \cdot (\mathbf{h}^l_u)^{\mathrm{T}}/\sqrt{n_{\mathrm{h}}}) \tag{12a}$$

$$\tilde{\mathbf{f}}^{l+1}_A = \mathbf{W}^l_{\mathrm{f},1} \cdot [\mathbf{f}^l_A || (\sum_{u \in A} a^l_{A,u} \mathbf{h}^l_u)] + \mathbf{b}^l_{\mathrm{f},1} \tag{12b}$$

where the Softmax operation is taken over all SAAOs u centered on atom A. Then the global attention α^l_A is calculated for all atoms in the molecule to update the molecule-level attribute $\mathbf{q}^{l+1}$:

$$\alpha^{l+1}_A = \mathrm{Softmax}(\mathbf{q}^l \cdot (\tilde{\mathbf{f}}^{l+1}_A)^{\mathrm{T}}/\sqrt{n_{\mathrm{h}}}) \tag{13a}$$

$$\mathbf{q}^{l+1} = \mathbf{q}^l + \sum_A \alpha^{l+1}_A \tilde{\mathbf{f}}^{l+1}_A \tag{13b}$$

where the Softmax is taken over all atoms in the molecule, and the initial global attribute $\mathbf{q}^0$ is a molecule-independent, trainable parameter vector.

Finally, the molecule- and atom-level information is propagated back to the SAAO attributes:

$$\mathbf{f}^{l+1}_A = \alpha^{l+1}_A \tilde{\mathbf{f}}^{l+1}_A \tag{14a}$$

$$\mathbf{h}^{l+1}_u = \mathbf{W}^l_{\mathrm{f},2} \cdot [\mathbf{f}^{l+1}_A || \tilde{\mathbf{h}}^{l+1}_u] + \mathbf{b}^l_{\mathrm{f},2}. \tag{14b}$$

The list of trainable model parameters is: $\mathbf{W}^{\mathrm{aux}}$, $\mathbf{W}^l_{\mathrm{h},1}$, $\mathbf{W}^l_{\mathrm{h},2}$, $\mathbf{b}^l_{\mathrm{h},1}$, $\mathbf{b}^l_{\mathrm{h},2}$, $\mathbf{W}^l_{\mathrm{m}}$, $\mathbf{b}^l_{\mathrm{m}}$, $\mathbf{W}^{l,i}_{\mathrm{a}}$, $\mathbf{W}^l_{\mathrm{e},1}$, $\mathbf{W}^l_{\mathrm{e},2}$, $\mathbf{b}^l_{\mathrm{e},1}$, $\mathbf{b}^l_{\mathrm{e},2}$, $\mathbf{W}^l_{\mathrm{f},1}$, $\mathbf{W}^l_{\mathrm{f},2}$, $\mathbf{b}^l_{\mathrm{f},1}$, $\mathbf{b}^l_{\mathrm{f},2}$, $\mathbf{q}^0$, and the parameters of $\mathrm{Enc}_{\mathrm{h}}$, $\mathrm{Enc}_{\mathrm{e}}$, $\mathrm{Enc}_{\mathrm{f}}$, Dec, and $\mathrm{Dec}^{\mathrm{aux}}$.

Hyperparameter	Meaning	Value
n_{r}	Number of basis functions for node embedding	8
m_{r}	Number of basis functions for edge embedding	8
n_{h}	Dimension of hidden node attributes	256
n_{e}	Dimension of hidden edge attributes	64
n_{a}	Number of attention heads	4
L	Number of message passing & pooling layers	2
L_{enc}	Number of dense layers in $\mathrm{Enc}_{\mathrm{h}}$ and $\mathrm{Enc}_{\mathrm{e}}$	3
L_{dec}	Number of residual blocks in a decoding network	3
n_{d}	Hidden dimension of a decoding network	256
γ	Batch normalization momentum	0.4
$c_{\mathbf{F}}$	Cutoff value for $\tilde{F}_{uv}$	6.0
$c_{\mathbf{D}}$	Cutoff value for $\tilde{D}_{uv}$	9.45
$c_{\mathbf{P}}$	Cutoff value for $\tilde{P}_{uv}$	6.0
$c_{\mathbf{S}}$	Cutoff value for $\tilde{S}_{uv}$	6.0
$c_{\mathbf{H}}$	Cutoff value for $\tilde{H}_{uv}$	6.0
σ	Morlet wavelet RBF scale	1/3

Table 2: Model hyperparameters employed in OrbNet. All cutoff values are in atomic units.

C Model hyperparameters and training details

Table 2 summarizes the hyperparameters employed in this work. We perform a pre-transformation on the input features from $\mathbf{F}$, $\mathbf{D}$, $\mathbf{P}$, $\mathbf{H}$ and $\mathbf{S}$ to obtain $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{X}}^{\mathrm{e}}$: We normalize all diagonal SAAO tensor values X_{uu} to the range $[0, 1)$ for each operator type to obtain $\tilde{X}_u$; for off-diagonal SAAO tensor values, we take $\tilde{X}_{uv} = -\ln(|X_{uv}|)$ for $\mathbf{X} \in \{\, \mathbf{F}, \mathbf{P}, \mathbf{S}, \mathbf{H} \,\}$, and $\tilde{D}_{uv} = D_{uv}$.

Training is performed on a loss function of the form

$$\begin{aligned} \mathcal{L}(\hat{\mathbf{E}}, \mathbf{E}, \hat{\mathbf{d}}, \mathbf{d}) &= (1-\alpha)\sum_i \mathcal{L}_2(\hat{E}_i, E_i) \\ &+ \alpha \sum_i \mathcal{L}_2(\hat{E}_i - \hat{E}_{t(i)}, E_i - E_{t(i)}) \quad (15) \\ &+ \beta \sum_i \sum_{A \in i} \mathcal{L}_2(\hat{\mathbf{d}}_A, \mathbf{d}_A). \quad (16) \end{aligned}$$

$\sum_i$ denotes summation over a minibatch of molecular geometries i. For each geometry i, we randomly sample another conformer of the same molecule $t(i)$ to evaluate the relative conformer loss $\mathcal{L}_2(\hat{E}_i - \hat{E}_{t(i)}, E_i - E_{t(i)})$; $\mathbf{E}$ denotes the ground truth energy values of the minibatch, $\hat{\mathbf{E}}$ denotes the model prediction values of the minibatch; $\hat{\mathbf{d}}_A$ and $\mathbf{d}_A$ denote the predicted and reference auxiliary target vectors for each atom A in molecule i, and $\mathcal{L}_2(\hat{y}, y) = ||\hat{y} - y||_2^2$ denotes the L2 loss function. For the model used in Section 3.1, we choose $\alpha = 0$ as only the optimized geometries are available; for models in Section 3.2, we choose $\alpha = 0.95$. β is adaptively updated using the GradNorm[14] method.

All models are trained on a single Nvidia Tesla V100-SXM2-32GB GPU using the Adam optimizer [15]. For all training runs, we set the minibatch size to 64 and use a cosine annealing with warmup learning rate schedule [16] that performs a linear learning rate increase from 3×10^{-6} to 3×10^{-4} for the initial 100 epochs, and a cosine decay from 3×10^{-4} to 0 for 200 epochs.

D Analytical nuclear gradients for symmetry-adapted atomic-orbital features

The electronic energy in the OrbNet model is given by

$$E_{\text{out}}[\mathbf{f}] = E_{\text{xTB}} + E_{\text{NN}}[\mathbf{f}]. \quad (17)$$

Here, $\mathbf{f}$ denotes the features, which correspond to the matrix elements of the quantum mechanical operators $\{\mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{H}, \mathbf{S}\}$ evaluated in the SAAO basis.

D.1 Generation of SAAOs

We denote $\{\phi^A_{n,l,m}\}$ as the set of atomic basis functions with atom indices A, with principle, angular and magnetic quantum numbers n, l, m, and $\{\psi_i\}$ as the set of canonical molecular orbitals obtained from a low-level electronic structure calculation.

We define the transformation matrix $\mathbf{X}$ between AOs and SAAOs as eigenvectors of the local density matrices (in covariant form):

$$\tilde{\mathbf{P}}^A_{n,l}\mathbf{X}^A_{n,l} = \mathbf{X}^A_{n,l}\Sigma^A_{n,l} \tag{18}$$

where $\tilde{\mathbf{P}}$ is the covariant density matrix in AO basis and is defined as

$$\tilde{\mathbf{P}} = \mathbf{S}\mathbf{P}^{\text{AO}}\mathbf{S} \tag{19}$$

The SAAOs, $\{\hat{\phi}_\kappa\}$, are thus expressed as

$$|\hat{\phi}_\kappa\rangle = \sum_\mu X_{\mu\kappa}|\phi_\mu\rangle \tag{20}$$

D.2 Matrices of operators in the SAAO basis for featurization

- The xTB core-Hamiltonian matrix in the SAAO basis

$$\mathbf{H}^{\text{SAAO}} = \mathbf{X}^\dagger\mathbf{H}^{\text{AO}}\mathbf{X} \tag{21}$$

- Overlap matrix in the SAAO basis

$$\mathbf{S}^{\text{SAAO}} = \mathbf{X}^\dagger\mathbf{S}^{\text{AO}}\mathbf{X} \tag{22}$$

- The xTB Fock matrix in the SAAO basis

$$\mathbf{F}^{\text{SAAO}} = \mathbf{X}^\dagger\mathbf{F}^{\text{AO}}\mathbf{X} \tag{23}$$

- Density matrix in the SAAO basis

$$\mathbf{P}^{\text{SAAO}} = \mathbf{X}^\dagger\mathbf{P}^{\text{AO}}\mathbf{X} \tag{24}$$

- Centroid distance matrix in the SAAO basis

$$D^{\text{SAAO}}_{\kappa\lambda} = ||\langle\hat{\phi}_\kappa|\hat{\mathbf{r}}|\hat{\phi}_\kappa\rangle - \langle\hat{\phi}_\lambda|\hat{\mathbf{r}}|\hat{\phi}_\lambda\rangle|| = \left(\vec{d}_{\kappa\lambda}\cdot\vec{d}_{\kappa\lambda}\right)^{1/2} \tag{25}$$

where $\vec{d}_{\kappa\lambda}$ is defined as

$$\vec{d}_{\kappa\lambda} = \vec{r}^{\text{SAAO}}_{\kappa\kappa} - \vec{r}^{\text{SAAO}}_{\lambda\lambda} = \left(\mathbf{X}^\dagger\mathbf{r}^{\text{AO}}\mathbf{X}\right)_{\kappa\kappa} - \left(\mathbf{X}^\dagger\mathbf{r}^{\text{AO}}\mathbf{X}\right)_{\lambda\lambda} \tag{26}$$

where $\mathbf{r}^{\text{AO}}$ is the AO dipole matrix.

D.3 OrbNet analytical gradient

The Lagrangian for OrbNet is

$$\mathcal{L} = E_{\text{NN}}[\mathbf{f}] + \sum_{pq} W_{pq}\left(\mathbf{C}^\dagger\mathbf{S}\mathbf{C} - \mathbf{I}\right)_{pq} + \sum_{ai} z_{ai}F_{ai} \tag{27}$$

Second term: orbitals orthogonality constraint. Third term: Brillion conditions. Note: i, j are indices for occupied molecular orbitals (MOs), p, q are general indices for MOs.

D.4 Stationary condition for the Lagrangian with respect to the MOs

The Lagrangian is stationary with respect to variations of the MOs:

$$\frac{\partial \mathcal{L}}{\partial V_{pq}} = 0 \tag{28}$$

where V_{pq} is a variation of the MOs in terms of the orbital rotation between MO pair p and q and is defined as

$$\tilde{\mathbf{C}} = \mathbf{C}(\mathbf{I} + \mathbf{V}) \tag{29}$$

This leads to the following expressions for each term on the right-hand-side of Eq. 27:

$$A_{pq} = \frac{\partial E_{\text{NN}}[\mathbf{f}]}{\partial V_{pq}}\Big|_{\mathbf{V}=0} = \frac{\partial E_{\text{NN}}[\mathbf{f}]}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial V_{pq}}\Big|_{\mathbf{V}=0} \tag{30}$$

$$W_{pq} = \frac{\partial \sum_{pq} W_{pq} \big(\mathbf{C}^{\dagger}\mathbf{S}\mathbf{C} - \mathbf{I}\big)_{pq}}{\partial V_{pq}}\Big|_{\mathbf{V}=0} \tag{31}$$

$$(\mathbf{A}[\mathbf{z}])_{pq} = \frac{\partial \sum_{ai} z_{ai} F_{ai}}{\partial V_{pq}}\Big|_{\mathbf{V}=0} = (\mathbf{F}\mathbf{z})_{pq}\Big|_{q\in\text{occ}} + \big(\mathbf{F}\mathbf{z}^{\dagger}\big)_{pq}\Big|_{q\in\text{vir}} + 2\,(\mathbf{g}[\bar{\mathbf{z}}])_{pq}\Big|_{q\in\text{occ}} \tag{32}$$

In the following sections, we derive the working equations for the above terms.

D.5 SAAO derivatives

As will be shown later, the OrbNet energy gradient involves the derivatives of the SAAO transformation matrix $\mathbf{X}^{A}_{n,l}$ with respect to orbital rotations and nuclear coordinates. The derivatives of SAAOs are a bit involved, since SAAOs are eigenvectors of the local density matrices. We follow reference [17] and show how SAAO derivatives are computed.

Here, we restrict the discussion to the scenario where the eigenvalues of the local density matrices $\tilde{\mathbf{P}}^{A}_{n,l}$ are distinct, such that the eigenvectors (i.e. SAAOs) are uniquely determined up to a constant (real-valued) factor.

For generality, denote the (real, symmetric) matrix for which the eigenvalues/eigenvectors are solved as $\mathbf{A}$, its eigenvalues as $\mathbf{\Lambda}$, and its eigenvectors as $\mathbf{X}$, such that

$$\mathbf{A}\mathbf{X} = \mathbf{X}\mathbf{\Lambda} \tag{33}$$

with the eigenvectors $\mathbf{X}$ being orthonormal to each other,

$$\mathbf{X}^{T}\mathbf{X} = \mathbf{I} \tag{34}$$

Denote the derivative of a martrix with respect to a parameter p by a prime, for example,

$$\frac{d\mathbf{A}}{dp} \equiv \mathbf{A}' \tag{35}$$

The eigenvalue derivatives are computed as

$$\frac{d\lambda_k}{dp} = \mathbf{X}^{\dagger}_{k}\mathbf{A}'\mathbf{X}_{k} \tag{36}$$

Define matrix $\mathbf{T}$ as

$$\mathbf{T} = \mathbf{X}^{-1}\mathbf{X}' \tag{37}$$

For the case where the eigenvalues are distinct, we have

$$T_{kl} = \frac{\mathbf{X}^{\dagger}_{k}\mathbf{A}'\mathbf{X}_{l}}{\lambda_l - \lambda_k} \quad \text{for } k \neq l, \quad T_{kk} = 0 \tag{38}$$

The eigenvector derivative can be determined via Eq. (37), as

$$\mathbf{X}' = \mathbf{X}\mathbf{T} \tag{39}$$

Let's denote a diagonal block of the covariant density matrix on atom A with quantum numbers $\{n, l\}$ as I, such that

$$\tilde{\mathbf{P}}_I \equiv \tilde{\mathbf{P}}^A_{n,l} \tag{40}$$

The SAAO eigenvalue problem for the I-th diagonal block can thus be re-written as

$$\tilde{\mathbf{P}}_I \mathbf{X}_I = \mathbf{X}_I \mathbf{\Sigma}_I \tag{41}$$

The derivatives of $\mathbf{X}_I$ with respect to an arbitrary variable ξ, denoted as $\mathbf{X}^\xi_I$, can be expressed as:

$$\mathbf{X}^\xi_I = \mathbf{X}_I \mathbf{T}^\xi_I \tag{42}$$

where matrix $\mathbf{T}^\xi_I$ is defined according to Eq. (38) as

$$T^\xi_{I,\kappa\lambda} = \frac{\mathbf{X}^\dagger_{I,\kappa} \tilde{\mathbf{P}}^\xi_I \mathbf{X}_{I,\lambda}}{\epsilon_\lambda - \epsilon_\kappa} \quad \text{for } \kappa \neq \lambda, \quad T^\xi_{I,\kappa\kappa} = 0 \tag{43}$$

where $\tilde{\mathbf{P}}^\xi_I$ is the derivative of the covariant density matrix for I-th diagonal block.

The gradients of the OrbNet energy usually involve the term $A^\xi \equiv \text{Tr}[\mathbf{B}\mathbf{T}^\xi]$, which can be re-written as

$$\text{Tr}[\mathbf{B}\mathbf{T}^\xi] = \sum_I \text{Tr}[\mathbf{B}_I \mathbf{T}^\xi_I] = \sum_I \sum_{\kappa \neq \lambda} B_{I,\kappa\lambda} T^\xi_{I,\lambda\kappa} = \sum_I \sum_{\kappa \neq \lambda} B_{I,\kappa\lambda} \frac{\mathbf{X}^\dagger_{I,\lambda} \tilde{\mathbf{P}}^\xi_I \mathbf{X}_{I,\kappa}}{\epsilon_\kappa - \epsilon_\lambda}$$

Define $\bar{\mathbf{B}}_I$ as

$$\bar{B}_{I,\kappa\lambda} = \frac{B_{I,\kappa\lambda}}{\epsilon_\kappa - \epsilon_\lambda} \quad \text{for } \kappa \neq \lambda, \quad \bar{B}_{I,\kappa\kappa} = 0 \tag{44}$$

and $\tilde{\mathbf{B}}_I$ as

$$\tilde{\mathbf{B}}_I = \frac{1}{2} \mathbf{X}_I (\bar{\mathbf{B}}_I + \bar{\mathbf{B}}^\dagger_I) \mathbf{X}^\dagger_I \tag{45}$$

where we have symmetrized $\tilde{\mathbf{B}}_I$. Finally, we have

$$\text{Tr}[\mathbf{B}\mathbf{T}^\xi] = \sum_I \sum_{\kappa\lambda} \bar{B}_{I,\kappa\lambda} \mathbf{X}^\dagger_{I,\lambda} \tilde{\mathbf{P}}^\xi_I \mathbf{X}_{I,\kappa} = \sum_I \text{Tr}[\tilde{\mathbf{B}}_I \tilde{\mathbf{P}}^\xi_I]$$

D.5.1 Derivatives of the SAAO basis with respect to MO variations

The derivatives of the SAAOs, $\mathbf{X}$ with respect to orbital variation V_{pq} can be expressed as:

$$\frac{\partial \mathbf{X}}{\partial V_{pq}} = \mathbf{X}\mathbf{T}^{V_{pq}} \tag{46}$$

where $\mathbf{T}^{V_{pq}}$ is defined according to Eq. (43) as

$$T^{V_{pq}}_{I,\kappa\lambda} = \frac{\mathbf{X}^T_{I,\kappa} \tilde{\mathbf{P}}^{V_{pq}}_I \mathbf{X}_{I,\lambda}}{\epsilon_{I,\lambda} - \epsilon_{I,\kappa}} \quad \text{for } \kappa \neq \lambda, \quad T^{V_{pq}}_{I,\kappa\kappa} = 0 \tag{47}$$

where $\tilde{\mathbf{P}}^{V_{pq}}_I$ is the derivative of the I-th diagonal block of local density matrix with respect to orbital variation V_{pq} and is defined as

$$\begin{aligned} \tilde{P}^{V_{pq}}_{\mu\nu \in I} \equiv \frac{\partial \tilde{P}^I_{\mu\nu}}{\partial V_{pq}} = \frac{\partial (\mathbf{SPS})_{\mu\nu}}{\partial V_{pq}} &= \sum_{\kappa\lambda} S_{\mu\kappa} \frac{\partial P_{\kappa\lambda}}{\partial V_{pq}} S_{\lambda\nu} \\ &= \sum_{\kappa\lambda} S_{\mu\kappa} (C_{\kappa p} n_q C_{\lambda q} + C_{\kappa q} n_q C_{\lambda p}) S_{\lambda\nu} \end{aligned} \tag{48}$$

where n_q is the occupation number of orbital q. For closed-shell systems at zero electronic temperature, n_q is defined as

$$n_q = \begin{cases} 2 & \text{if } q \in \text{ occupied} \\ 0 & \text{otherwise} \end{cases} \tag{49}$$

For other cases, n_q may be fractional numbers.

Define $\mathbf{Y} = \mathbf{SC}$, then

$$\tilde{\mathbf{P}}_I^{V_{pq}} = (\mathbf{Y}_p \mathbf{Y}_q^\dagger + \mathbf{Y}_q \mathbf{Y}_p^\dagger) n_q \tag{50}$$

The orbital derivatives of the OrbNet energy usually involve the term Tr$[\mathbf{BT}^{pq}]$, which can be expressed according to Eq. (46) as

$$\text{Tr}[\mathbf{BT}^{pq}] = \sum_I \text{Tr}[\tilde{\mathbf{B}}_I \tilde{\mathbf{P}}_I^{V_{pq}}] = \sum_I \text{Tr}[\tilde{\mathbf{B}}_I (\mathbf{Y}_p \mathbf{Y}_q^\dagger + \mathbf{Y}_q \mathbf{Y}_p^\dagger) n_q] = M_{pq}$$

where $\tilde{\mathbf{B}}_I$ is defined in Eq. (45); M_{pq} is defined as

$$M_{pq} = 2\mathbf{Y}_p^\dagger \tilde{\mathbf{B}} \mathbf{Y}_q n_q \tag{51}$$

D.5.2 Derivatives of the SAAO basis with respect to nuclear coordinates

The derivatives of $\mathbf{X}$ with respect to nuclear coordinates x can be expressed as

$$\frac{\partial \mathbf{X}}{\partial x} = \mathbf{XT}^x \tag{52}$$

where $\mathbf{T}^x$ is defined according to Eq. (43) as

$$T_{I,\kappa\lambda}^x = \frac{\mathbf{X}_{I,\kappa}^T \tilde{\mathbf{P}}_I^x \mathbf{X}_{I,\lambda}}{\epsilon_{I,\lambda} - \epsilon_{I,\kappa}} \quad \text{for } \kappa \neq \lambda, \quad T_{I,\kappa\kappa}^x = 0 \tag{53}$$

where $\tilde{\mathbf{P}}_I^x$ is defined as

$$\begin{aligned} \tilde{P}_{\mu\nu\in I}^x \equiv \frac{\partial \tilde{P}_{\mu\nu}^I}{\partial x} &= \frac{\partial (\mathbf{SPS})_{\mu\nu}}{\partial x} \\ &= \sum_{\kappa\lambda} \frac{\partial S_{\mu\kappa}}{\partial x} P_{\kappa\lambda} S_{\lambda\nu} + S_{\mu\kappa} P_{\kappa\lambda} \frac{\partial S_{\lambda\nu}}{\partial x} \\ &= \sum_{\kappa\lambda} S_{\mu\kappa}^x P_{\kappa\lambda} S_{\lambda\nu} + S_{\mu\kappa} P_{\kappa\lambda} S_{\lambda\nu}^x \end{aligned} \tag{54}$$

Define $N = \mathbf{PS}$, then

$$\tilde{\mathbf{P}}_I^x = \left[\mathbf{S}^x \mathbf{N} + \mathbf{N}^\dagger \mathbf{S}^x\right]_I \tag{55}$$

The nuclear derivatives of the OrbNet energy usually involve the term Tr$[\mathbf{BT}^x]$, which can be rewritten according to Eq. 46 as

$$\text{Tr}[\mathbf{BT}^x] = \sum_I \text{Tr}[\tilde{\mathbf{B}}_I \tilde{\mathbf{P}}_I^x] = \sum_I \text{Tr}\left(\tilde{\mathbf{B}}_I \left[\mathbf{S}^x \mathbf{N} + \mathbf{N}^\dagger \mathbf{S}^x\right]_I\right) = \text{Tr}[\overline{\mathbf{W}} \mathbf{S}^x]$$

where $\tilde{\mathbf{B}}$ defined in Eq. 45; $\overline{\mathbf{W}}$ is defined as

$$\left.\overline{W}_{\mu\nu}^I\right|_{\nu\in I} = 2\left(\mathbf{N}\tilde{\mathbf{B}}_I\right)_{\mu\nu} \tag{56}$$

D.6 Derivatives of OrbNet energy with respect to the MOs

Define the derivatives of the OrbNet energy with respect to feature $\mathbf{f}$ as:

$$\mathbf{Q}^f = \frac{\partial E_{\text{NN}}[\mathbf{f}]}{\partial \mathbf{f}} \tag{57}$$

where $\mathbf{f} \in \{\mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{H}, \mathbf{S}\}$.

Note that $\mathbf{Q}^f$ has the same dimension as $\mathbf{f}$, and is symmetrized.

The derivatives of OrbNet energy with respect to the MO variations, Eq. 30, can be rewritten as

$$A_{pq} = \frac{\partial E_{\text{NN}}[\mathbf{f}]}{\partial V_{pq}}\Big|_{\mathbf{V}=0} = \frac{\partial E_{\text{NN}}[\mathbf{f}]}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial V_{pq}}\Big|_{\mathbf{V}=0} = \sum_f \left[\mathbf{Q}^f \cdot \frac{\partial \mathbf{f}}{\partial V_{pq}}\right] \tag{58}$$

Define

$$A^f_{pq} = \mathbf{Q}^f \cdot \frac{\partial \mathbf{f}}{\partial V_{pq}} \tag{59}$$

which corresponds to the contribution to OrbNet energy derivatives with respect to MOs from a specific feature $\mathbf{f}$. We then derive the expression of A^f_{pq} for each individual feature, as described below.

D.6.1 Core Hamiltonian

$$\begin{aligned} A^H_{pq} &= \mathbf{Q}^H \cdot \frac{\partial \mathbf{H}^{\text{SAAO}}}{\partial V_{pq}} = \mathbf{Q}^H \cdot \left[\mathbf{X}^\dagger \mathbf{H}^{\text{AO}} \frac{\partial \mathbf{X}}{\partial V_{pq}} + \text{transpose}\right] \\ &= \mathbf{Q}^H \cdot \left[\mathbf{X}^\dagger \mathbf{H}^{\text{AO}} \mathbf{X} \mathbf{T}^{pq} + \text{transpose}\right] \\ &= 2\text{Tr}\left[\mathbf{Q}^H \mathbf{H}^{\text{SAAO}} \mathbf{T}^{pq}\right] \\ &= 2\text{Tr}\left[\mathbf{B}^H \mathbf{T}^{pq}\right] \\ &= 2M^H_{pq} \end{aligned} \tag{60}$$

where M^H_{pq} is defined according to Eq. 51:

$$M^H_{pq} = \mathbf{Y}^\dagger_p (\tilde{\mathbf{B}}^H + \tilde{\mathbf{B}}^{H,\dagger}) \mathbf{Y}_q n_q \tag{61}$$

and $\mathbf{B}^H$ is

$$\mathbf{B}^H = \mathbf{Q}^H \mathbf{H}^{\text{SAAO}} \tag{62}$$

D.6.2 Overlap matrix

$$\begin{aligned} A^S_{pq} &= \mathbf{Q}^S \cdot \frac{\partial \mathbf{S}^{\text{SAAO}}}{\partial V_{pq}} = \mathbf{Q}^S \cdot \left[\mathbf{X}^\dagger \mathbf{S}^{\text{AO}} \frac{\partial \mathbf{X}}{\partial V_{pq}} + \text{transpose}\right] \\ &= 2\text{Tr}\left[\mathbf{Q}^S \mathbf{S}^{\text{SAAO}} \mathbf{T}^{pq}\right] \\ &= 2\text{Tr}\left[\mathbf{B}^S \mathbf{T}^{pq}\right] \\ &= 2M^S_{pq} \end{aligned} \tag{63}$$

with M^S_{pq} defined in a similar way as to Eq. 61.

D.6.3 Fock matrix

$$A^F_{pq} = \mathbf{Q}^F \cdot \frac{\partial \mathbf{F}^{\text{SAAO}}}{\partial V_{pq}} \tag{64}$$

with

$$\frac{\partial \mathbf{F}^{\text{SAAO}}}{\partial V_{pq}} = \left(\mathbf{X}^\dagger \mathbf{F}^{\text{AO}} \frac{\partial \mathbf{X}}{\partial V_{pq}} + \text{transpose}\right) + \mathbf{X}^\dagger \frac{\partial \mathbf{F}^{\text{AO}}}{\partial V_{pq}} \mathbf{X} \tag{65}$$

Therefore,

$$\begin{aligned}
A^F_{pq} &= 2\mathrm{Tr}\left[\mathbf{Q}^F\mathbf{F}^{\mathrm{SAAO}}\mathbf{T}^{pq}\right] + \mathrm{Tr}\left[\mathbf{Q}^F\mathbf{X}^\dagger\frac{\partial\mathbf{F}^{\mathrm{AO}}}{\partial V_{pq}}\mathbf{X}\right] \\
&= 2\mathrm{Tr}\left[\mathbf{B}^F\mathbf{T}^{pq}\right] + \mathrm{Tr}\left[\mathbf{Q}^{F,\mathrm{AO}}\frac{\partial\mathbf{F}^{\mathrm{AO}}}{\partial V_{pq}}\right] \\
&= 2\mathrm{Tr}\left[\mathbf{B}^F\mathbf{T}^{pq}\right] + \mathrm{Tr}\left[\mathbf{Q}^{F,\mathrm{AO}}\frac{\partial\mathbf{F}^{\mathrm{AO}}}{\partial\mathbf{D}^{\mathrm{AO}}}\frac{\partial\mathbf{D}^{\mathrm{AO}}}{\partial V_{pq}}\right] \\
&= 2\mathrm{Tr}\left[\mathbf{B}^F\mathbf{T}^{pq}\right] + 2\left(\mathbf{g}[2\mathbf{Q}^{F,\mathrm{AO}}]\right)_{pq} \\
&= 2\mathrm{Tr}\left[\mathbf{B}^F\mathbf{T}^{pq}\right] + 2\left(\mathbf{g}[2\mathbf{Q}^{F,\mathrm{AO}}]\right)_{pq} \\
&= 2M^F_{pq} + 2\left(\mathbf{g}[2\mathbf{Q}^{F,\mathrm{AO}}]\right)_{pq}
\end{aligned} \tag{66}$$

where g is the generalized xTB Fock matrix and $\mathbf{Q}^{F,\mathrm{AO}}$ is defined as

$$\mathbf{Q}^{F,\mathrm{AO}} = \mathbf{X}\mathbf{Q}^F\mathbf{X}^\dagger \tag{67}$$

D.6.4 Density matrix

$$A^P_{pq} = \mathbf{Q}^P \cdot \frac{\partial\mathbf{P}^{\mathrm{SAAO}}}{\partial V_{pq}} \tag{68}$$

where

$$\frac{\partial\mathbf{P}^{\mathrm{SAAO}}}{\partial V_{pq}} = \left(\mathbf{X}^\dagger\mathbf{P}^{\mathrm{AO}}\frac{\partial\mathbf{X}}{\partial V_{pq}} + \text{transpose}\right) + \mathbf{X}^\dagger\frac{\partial\mathbf{P}^{\mathrm{AO}}}{\partial V_{pq}}\mathbf{X} \tag{69}$$

Therefore,

$$\begin{aligned}
A^P_{pq} &= 2\mathrm{Tr}\left[\mathbf{Q}^P\mathbf{P}^{\mathrm{SAAO}}\mathbf{T}^{pq}\right] + 2\left(\mathbf{C}^\dagger\mathbf{Q}^{P,\mathrm{AO}}\mathbf{C}\right)_{pq} n_q \\
&= 2\mathrm{Tr}\left[\mathbf{B}^P T^{pq}\right] + 2\left(\mathbf{C}^\dagger\mathbf{Q}^{P,\mathrm{AO}}\mathbf{C}\right)_{pq} n_q \\
&= 2M^P_{pq} + 2\left(\mathbf{C}^\dagger\mathbf{Q}^{P,\mathrm{AO}}\mathbf{C}\right)_{pq} n_q
\end{aligned} \tag{70}$$

D.6.5 Centroid distance matrix

$$A^D_{pq} = \mathbf{Q}^D \cdot \frac{\partial\mathbf{D}^{\mathrm{SAAO}}}{\partial V_{pq}} \tag{71}$$

with

$$\frac{\partial D^{\mathrm{SAAO}}_{\kappa\lambda}}{\partial V_{pq}} = \frac{1}{D^{\mathrm{SAAO}}_{\kappa\lambda}}\vec{d}_{\kappa\lambda} \cdot \frac{\partial\vec{d}_{\kappa\lambda}}{\partial V_{pq}} \tag{72}$$

where $\frac{\partial\vec{d}_{\kappa\lambda}}{\partial V_{pq}}$ is defined as

$$\begin{aligned}
\frac{\partial\vec{d}_{\kappa\lambda}}{\partial V_{pq}} &= \left(\mathbf{X}^\dagger_\kappa\mathbf{r}^{\mathrm{AO}}\frac{\partial\mathbf{X}_\kappa}{\partial V_{pq}} - \mathbf{X}^\dagger_\lambda\mathbf{r}^{\mathrm{AO}}\frac{\partial\mathbf{X}_\lambda}{\partial V_{pq}}\right) + \text{transpose} \\
&= 2\left[(\mathbf{r}^{\mathrm{SAAO}}\mathbf{T}^{pq})_{\kappa\kappa} - (\mathbf{r}^{\mathrm{SAAO}}\mathbf{T}^{pq})_{\lambda\lambda}\right]
\end{aligned} \tag{73}$$

Then

$$\begin{aligned}
A^D_{pq} &= \mathbf{Q}^D \cdot \frac{\partial\mathbf{D}^{\mathrm{SAAO}}}{\partial V_{pq}} \\
&= \sum_{\kappa\lambda}\frac{Q^D_{\kappa\lambda}}{D^{\mathrm{SAAO}}_{\kappa\lambda}}\vec{d}_{\kappa\lambda} \cdot \frac{\partial\vec{d}_{\kappa\lambda}}{\partial V_{pq}}
\end{aligned} \tag{74}$$

Define

$$\vec{R}_{\kappa\lambda} = \frac{Q^D_{\kappa\lambda}}{D^{\text{SAAO}}_{\kappa\lambda}} \vec{d}_{\kappa\lambda} \tag{75}$$

Then

$$\begin{aligned} A^D_{pq} &= \sum_{\kappa\lambda} \vec{R}_{\kappa\lambda} \cdot \frac{\partial \vec{d}_{\kappa\lambda}}{\partial V_{pq}} \\ &= 2\sum_{\kappa\lambda} \vec{R}_{\kappa\lambda} \cdot \left[(\mathbf{r}^{\text{SAAO}} \mathbf{T}^{pq})_{\kappa\kappa} - (\mathbf{r}^{\text{SAAO}} \mathbf{T}^{pq})_{\lambda\lambda} \right] \\ &= 4\sum_{\kappa\lambda} \vec{R}_{\kappa\lambda} \cdot (\mathbf{r}^{\text{SAAO}} \mathbf{T}^{pq})_{\kappa\kappa} \end{aligned} \tag{76}$$

Define

$$\vec{R}^L_\kappa = \sum_\lambda \vec{R}_{\kappa\lambda} \tag{77}$$

$$B^D_{\kappa\lambda} = \vec{R}^L_\kappa \cdot \vec{r}^{\text{SAAO}}_{\kappa\lambda} \tag{78}$$

Then

$$\begin{aligned} A^D_{pq} &= 4\text{Tr}\left[\mathbf{B}^D \mathbf{T}^{pq} \right] \\ &= 4M^S_{pq} \end{aligned} \tag{79}$$

where M^D_{pq} defined in a similar way as to Eq. 61.

D.7 Derivatives of OrbNet energy with respect to nuclear coordinates

The derivatives of OrbNet energy with respect to nuclear coordinates can be written as

$$\frac{\partial E_{\text{NN}}}{\partial x} = \frac{\partial E_{\text{NN}}[\mathbf{f}]}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial x} = \sum_f \left[\mathbf{Q}^f \frac{\partial \mathbf{f}}{\partial x} \right] \tag{80}$$

Define:

$$A^f_x = \mathbf{Q}^f \cdot \frac{\partial \mathbf{f}}{\partial x} \tag{81}$$

which corresponds to the contribution to OrbNet energy derivatives with respect to MOs from a specific feature $\mathbf{f}$. x Now let's derive the expression of A^f_x for each individual feature:

D.7.1 Core Hamiltonian

$$\begin{aligned} A^H_x &= \mathbf{Q}^H \cdot \frac{\partial \mathbf{H}^{\text{SAAO}}}{\partial x} = \mathbf{Q}^H \cdot \left[\mathbf{X}^\dagger \mathbf{H}^{\text{AO}} \frac{\partial \mathbf{X}}{\partial x} + \text{transpose} \right] \\ &= \mathbf{Q}^H \cdot \left[\mathbf{X}^\dagger \mathbf{H}^{\text{AO}} \mathbf{X} \mathbf{T}^x + \text{transpose} \right] \\ &= 2\text{Tr}\left[\mathbf{Q}^H \mathbf{H}^{\text{SAAO}} \mathbf{T}^x \right] + \text{Tr}\left[\mathbf{Q}^H \mathbf{X}^\dagger \frac{\partial \mathbf{H}^{\text{AO}}}{\partial x} \mathbf{X} \right] \\ &= 2\text{Tr}\left[\mathbf{Q}^H \mathbf{H}^{\text{SAAO}} \mathbf{T}^x \right] + \text{Tr}\left[\mathbf{Q}^{H,\text{AO}} \frac{\partial \mathbf{H}^{\text{AO}}}{\partial x} \right] \\ &= 2\text{Tr}\left[\mathbf{B}^H \mathbf{T}^x \right] + \text{Tr}\left[\mathbf{Q}^{H,\text{AO}} \frac{\partial \mathbf{H}^{\text{AO}}}{\partial x} \right] \\ &= 2\text{Tr}\left[\overline{\mathbf{W}}^H \mathbf{S}^x \right] + \text{Tr}\left[\mathbf{Q}^{H,\text{AO}} \frac{\partial \mathbf{H}^{\text{AO}}}{\partial x} \right] \end{aligned} \tag{82}$$

where $\overline{\mathbf{W}}^H$ is defined according to Eq. 56.

D.7.2 Overlap matrix

$$
\begin{aligned}
A_x^S = \mathbf{Q}^S \cdot \frac{\partial \mathbf{S}^{\text{SAAO}}}{\partial x} &= \mathbf{Q}^S \cdot \left[\mathbf{X}^\dagger \mathbf{S}^{\text{AO}} \frac{\partial \mathbf{X}}{\partial x} + \text{transpose} \right] \\
&= 2\text{Tr}\left[\mathbf{Q}^S \mathbf{S}^{\text{SAAO}} \mathbf{T}^x \right] + \text{Tr}\left[\mathbf{Q}^S \mathbf{X}^\dagger \frac{\partial \mathbf{S}^{\text{AO}}}{\partial x} \mathbf{X} \right] \\
&= 2\text{Tr}\left[\mathbf{Q}^S \mathbf{S}^{\text{SAAO}} \mathbf{T}^x \right] + \text{Tr}\left[\mathbf{Q}^{S,\text{AO}} \frac{\partial \mathbf{S}^{\text{AO}}}{\partial x} \right] \\
&= 2\text{Tr}\left[\mathbf{B}^S \mathbf{T}^x \right] + \text{Tr}\left[\mathbf{Q}^{S,\text{AO}} \frac{\partial \mathbf{S}^{\text{AO}}}{\partial x} \right] \\
&= 2\text{Tr}\left[\overline{\mathbf{W}}^S \mathbf{S}^x \right] + \text{Tr}\left[\mathbf{Q}^{S,\text{AO}} \mathbf{S}^x \right] \\
&= \text{Tr}\left[(\overline{\mathbf{W}}^S + \mathbf{Q}^{S,\text{AO}}) \mathbf{S}^x \right]
\end{aligned} \tag{83}
$$

where $\overline{\mathbf{W}}^S$ is defined according to Eq. 56.

D.7.3 Fock matrix

$$
A_x^F = \mathbf{Q}^F \cdot \frac{\partial \mathbf{F}^{\text{SAAO}}}{\partial x} \tag{84}
$$

with

$$
\frac{\partial \mathbf{F}^{\text{SAAO}}}{\partial x} = \left(\mathbf{X}^\dagger \mathbf{F}^{\text{AO}} \frac{\partial \mathbf{X}}{\partial x} + \text{transpose} \right) + \mathbf{X}^\dagger \frac{\partial \mathbf{F}^{\text{AO}}}{\partial x} \mathbf{X} \tag{85}
$$

Therefore,

$$
\begin{aligned}
A_x^F &= 2\text{Tr}\left[\mathbf{Q}^F \mathbf{F}^{\text{SAAO}} \mathbf{T}^x \right] + \text{Tr}\left[\mathbf{Q}^F \mathbf{X}^\dagger \frac{\partial \mathbf{F}^{\text{AO}}}{\partial x} \mathbf{X} \right] \\
&= 2\text{Tr}\left[\mathbf{Q}^F \mathbf{F}^{\text{SAAO}} \mathbf{T}^x \right] + \text{Tr}\left[\mathbf{Q}^{F,\text{AO}} \frac{\partial \mathbf{F}^{\text{AO}}}{\partial x} \right] \\
&= 2\text{Tr}\left[\mathbf{B}^F \mathbf{T}^x \right] + \text{Tr}\left[\mathbf{Q}^{F,\text{AO}} \frac{\partial \mathbf{F}^{\text{AO}}}{\partial x} \right] \\
&= 2\text{Tr}\left[\overline{\mathbf{W}}^F \mathbf{S}^x \right] + \text{Tr}\left[\mathbf{Q}^{F,\text{AO}} \frac{\partial \mathbf{F}^{\text{AO}}}{\partial x} \right]
\end{aligned} \tag{86}
$$

where $\overline{\mathbf{W}}^F$ is defined according to Eq. 56.

D.7.4 Density matrix

$$
A_x^P = \mathbf{Q}^P \cdot \frac{\partial \mathbf{P}^{\text{SAAO}}}{\partial x} \tag{87}
$$

where

$$
\frac{\partial \mathbf{P}^{\text{SAAO}}}{\partial x} = \mathbf{X}^\dagger \mathbf{P}^{\text{AO}} \frac{\partial \mathbf{X}}{\partial x} + \text{transpose} \tag{88}
$$

Therefore,

$$
\begin{aligned}
A_x^P &= 2\text{Tr}\left[\mathbf{Q}^P \mathbf{P}^{\text{SAAO}} \mathbf{T}^x \right] \\
&= 2\text{Tr}\left[\mathbf{B}^P \mathbf{T}^x \right] \\
&= 2\text{Tr}\left[\overline{\mathbf{W}}^P \mathbf{S}^x \right]
\end{aligned} \tag{89}
$$

where $\overline{\mathbf{W}}^P$ is defined according to Eq. 56.

D.7.5 Centroid distance matrix

$$A_x^D = \mathbf{Q}^D \cdot \frac{\partial \mathbf{D}^{\mathrm{SAAO}}}{\partial x} \tag{90}$$

with

$$\frac{\partial D_{\kappa\lambda}^{\mathrm{SAAO}}}{\partial x} = \frac{1}{D_{\kappa\lambda}^{\mathrm{SAAO}}} \vec{d}_{\kappa\lambda} \cdot \frac{\partial \vec{d}_{\kappa\lambda}}{\partial x} \tag{91}$$

where $\frac{\partial \vec{d}_{\kappa\lambda}}{\partial x}$ is defined as

$$\begin{aligned}\frac{\partial \vec{d}_{\kappa\lambda}}{\partial x} &= \left[\left(\mathbf{X}_\kappa^\dagger \mathbf{r}^{\mathrm{AO}} \frac{\partial \mathbf{X}_\kappa}{\partial x} - \mathbf{X}_\lambda^\dagger \mathbf{r}^{\mathrm{AO}} \frac{\partial \mathbf{X}_\lambda}{\partial x}\right) + \text{transpose}\right] + (\mathbf{X}^\dagger \frac{\partial \mathbf{r}^{\mathrm{AO}}}{\partial x} \mathbf{X})_{\kappa\kappa} - (\mathbf{X}^\dagger \frac{\partial \mathbf{r}^{\mathrm{AO}}}{\partial x} \mathbf{X})_{\lambda\lambda} \\ &= 2\left[(\mathbf{r}^{\mathrm{SAAO}}\mathbf{T}^x)_{\kappa\kappa} - (\mathbf{r}^{\mathrm{SAAO}}\mathbf{T}^x)_{\lambda\lambda}\right] + (\mathbf{X}^\dagger \frac{\partial \mathbf{r}^{\mathrm{AO}}}{\partial x} \mathbf{X})_{\kappa\kappa} - (\mathbf{X}^\dagger \frac{\partial \mathbf{r}^{\mathrm{AO}}}{\partial x} \mathbf{X})_{\lambda\lambda}\end{aligned} \tag{92}$$

This leads to

$$\begin{aligned}A_x^D &= \mathbf{Q}^D \cdot \frac{\partial \mathbf{D}^{\mathrm{SAAO}}}{\partial x} = \sum_{\kappa\lambda} Q_{\kappa\lambda}^D \frac{\partial D_{\kappa\lambda}^{\mathrm{SAAO}}}{\partial x} \\ &= \sum_{\kappa\lambda} \frac{Q_{\kappa\lambda}^D}{D_{\kappa\lambda}^{\mathrm{SAAO}}} \vec{d}_{\kappa\lambda} \cdot \frac{\partial \vec{d}_{\kappa\lambda}}{\partial x} = \sum_{\kappa\lambda} \vec{R}_{\kappa\lambda} \cdot \frac{\partial \vec{d}_{\kappa\lambda}}{\partial x} \\ &= \sum_{\kappa\lambda} \vec{R}_{\kappa\lambda} \cdot \left(2\left[(\mathbf{r}^{\mathrm{SAAO}}\mathbf{T}^x)_{\kappa\kappa} - (\mathbf{r}^{\mathrm{SAAO}}\mathbf{T}^x)_{\lambda\lambda}\right] + (\mathbf{X}^\dagger \frac{\partial \mathbf{r}^{\mathrm{AO}}}{\partial x} \mathbf{X})_{\kappa\kappa} - (\mathbf{X}^\dagger \frac{\partial \mathbf{r}^{\mathrm{AO}}}{\partial x} \mathbf{X})_{\lambda\lambda}\right) \\ &= 2\sum_{\kappa\lambda} \vec{R}_{\kappa\lambda} \cdot \left(2(\mathbf{r}^{\mathrm{SAAO}}\mathbf{T}^x)_{\kappa\kappa} + (\mathbf{X}^\dagger \frac{\partial \mathbf{r}^{\mathrm{AO}}}{\partial x} \mathbf{X})_{\kappa\kappa}\right)\end{aligned} \tag{93}$$

where $\vec{R}$ is defined in Eq. (75).

Define $\bar{\mathbf{d}}^L$as

$$\bar{\mathbf{d}}_{\mu\nu}^L = \sum_\kappa X_{\mu\kappa} X_{\nu\kappa} \vec{R}_\kappa^L \tag{94}$$

where $\vec{R}^L$ is defined in Eq. (77).

Then

$$\begin{aligned}A_x^D &= 4\mathrm{Tr}[\mathbf{B}^D\mathbf{T}^x] + 2\mathrm{Tr}\left[\bar{\mathbf{d}}^L \cdot \frac{\partial \mathbf{r}^{\mathrm{AO}}}{\partial x}\right] \\ &= 4\mathrm{Tr}\left[\overline{\mathbf{W}}^D \mathbf{S}^x\right] + 2\mathrm{Tr}\left[\bar{\mathbf{d}}^L \cdot \frac{\partial \mathbf{r}^{\mathrm{AO}}}{\partial x}\right].\end{aligned} \tag{95}$$

D.8 xTB generalized Fock matrix

The xTB generalized Fock matrix is defined as

$$(\mathbf{g}[\mathbf{Y}])_{\mu\nu} = \sum_{\kappa\lambda} \frac{\partial F_{\mu\nu}}{\partial P_{\kappa\lambda}} Y_{\kappa\lambda} \tag{96}$$

where $\mathbf{Y}$ is an arbitrary symmetric matrix with the same dimension as the AO density matrix $\mathbf{P}$.

The xTB Fock matrix is defined as

$$F_{\mu\nu} = H_{\mu\nu} + \frac{1}{2} S_{\mu\nu} \sum_{C,l''} (\gamma_{AC,ll''} + \gamma_{BC,l'l''}) p_{l''}^C + \frac{1}{2} S_{\mu\nu} (q_A^2 \Gamma_A + q_B^2 \Gamma_B) \quad (\mu \in A, l; \nu \in B, l') \tag{97}$$

which is a functional of the shell-resolved charges, i.e. $\mathbf{F}[p^C_{l''}]$.

With the above expression, the xTB generalized Fock matrix can be computed as

$$(\mathbf{g}[\mathbf{Y}])_{\mu\nu} = \sum_{\kappa\lambda} \frac{\partial F_{\mu\nu}}{\partial P_{\kappa\lambda}} Y_{\kappa\lambda} = \sum_{C,l''} \sum_{\kappa\lambda} \frac{\partial F_{\mu\nu}}{\partial p^C_{l''}} \frac{\partial p^C_{l''}}{\partial P_{\kappa\lambda}} Y_{\kappa\lambda} \tag{98}$$

The shell-resolved charges $p^C_{l''}$ are defined as

$$p^C_{l''} = {p^C_{l''}}^0 - \sum_{\kappa \in C, l''} \sum_{\lambda} S_{\kappa\lambda} P_{\kappa\lambda} \tag{99}$$

Define

$$\tilde{p}^C_{l''} \equiv \sum_{\kappa\lambda} \frac{\partial p^C_{l''}}{\partial P_{\kappa\lambda}} Y_{\kappa\lambda} = - \sum_{\kappa \in C, l''} \sum_{\lambda} S_{\kappa\lambda} Y_{\kappa\lambda} \tag{100}$$

The final expression for the xTB generalized Fock matrix is

$$\begin{aligned}(\mathbf{g}[\mathbf{Y}])_{\mu\nu} &= \sum_{C,l''} \sum_{\kappa\lambda} \frac{\partial F_{\mu\nu}}{\partial p^C_{l''}} \frac{\partial p^C_{l''}}{\partial P_{\kappa\lambda}} Y_{\kappa\lambda} = \sum_{C,l''} \frac{\partial F_{\mu\nu}}{\partial p^C_{l''}} \tilde{p}^C_{l''} \\ &= \frac{1}{2} S_{\mu\nu} \sum_{C,l''} (\gamma_{AC,ll''} + \gamma_{BC,l'l''}) \tilde{p}^C_{l''} + S_{\mu\nu} (q_A \tilde{q}_A \Gamma_A + q_B \tilde{q}_B \Gamma_B)\end{aligned} \tag{101}$$

where $\tilde{q}_A = \sum_l \tilde{p}^A_l$.

D.9 Coupled-perturbed z-vector equation for xTB

Combining the stationary condition of the Lagrangian, Eq. 28 and the condition $\mathbf{x} = \mathbf{x}^\dagger$ leads to the coupled-perturbed z-vector equation for xTB:

$$(\varepsilon_a - \varepsilon_i) z_{ai} + 2[\mathbf{g}(\bar{\mathbf{z}})]_{ai} = -(A_{ai} - A_{ia}) \tag{102}$$

where $\varepsilon_a, \varepsilon_i$ are the xTB orbital energies, $\mathbf{z}$ is the Lagrange multiplier defined in Eq. 27. $\bar{\mathbf{z}} = \mathbf{z} + \mathbf{z}^\dagger$. $\mathbf{g}(\bar{\mathbf{z}})$ is the generalized xTB Fock matrix and is defined in Eq. 101.

D.10 Expression for W

The stationary condition of the Lagrangian, Eq. 28 also leads to the expression for the weight matrix $\mathbf{W}$:

$$W_{pq} = -\frac{1}{4}(1 + \hat{P}_{pq})[\mathbf{A} + \mathbf{A}(\mathbf{z})]_{pq} \tag{103}$$

where $\hat{P}_{pq}$ is the permutation operator that permutes indices p and q.

D.11 Final gradient expression

With all intermediate quantities obtained in the previous sections, we can now write the expression for the OrbNet energy gradient:

$$\frac{dE_{\text{out}}}{dx} = \frac{\partial E_{\text{out}}}{\partial x} + \text{Tr}[\mathbf{W}\mathbf{S}^x] + \text{Tr}[\mathbf{z}\mathbf{F}^{(x)}] \tag{104}$$

where the first term on the right-hand-side can be computed as

$$\frac{\partial E_{\text{out}}}{\partial x} = \frac{dE_{\text{xTB}}}{dx} + \sum_f \left[\mathbf{Q}^f \frac{\partial \mathbf{f}}{\partial x}\right] \tag{105}$$

$$\begin{aligned}\frac{dE_{\text{out}}}{dx} &= \frac{dE_{\text{xTB}}}{dx} + \sum_f \left[\mathbf{Q}^f \frac{\partial \mathbf{f}}{\partial x}\right] + \text{Tr}[\mathbf{W}\mathbf{S}^x] + \text{Tr}[\mathbf{z}\mathbf{F}^{(x)}] \\ &= \frac{dE_{\text{xTB}}}{dx} + \sum_f \left[\mathbf{Q}^f \frac{\partial \mathbf{f}}{\partial x}\right] + \text{Tr}[\mathbf{W}\mathbf{S}^x] + \text{Tr}[\mathbf{z}^{\text{AO}} \frac{\partial \mathbf{F}^{\text{AO}}}{\partial x}] \\ &= \frac{dE_{\text{xTB}}}{dx} + \text{Tr}[\mathbf{W}\mathbf{S}^x] + \text{Tr}[\mathbf{z}^{\text{AO}}\mathbf{F}^x] \\ &\quad + 2\text{Tr}\left[\overline{\mathbf{W}}^H \mathbf{S}^x\right] + \text{Tr}\left[\mathbf{Q}^{H,\text{AO}}\mathbf{H}^x\right] \\ &\quad + 2\text{Tr}\left[\overline{\mathbf{W}}^S \mathbf{S}^x\right] + \text{Tr}\left[\mathbf{Q}^{S,\text{AO}}\mathbf{S}^x\right] \\ &\quad + 2\text{Tr}\left[\overline{\mathbf{W}}^F \mathbf{S}^x\right] + \text{Tr}\left[\mathbf{Q}^{F,\text{AO}}\mathbf{F}^x\right] \\ &\quad + 2\text{Tr}\left[\overline{\mathbf{W}}^P \mathbf{S}^x\right] \quad + 4\text{Tr}\left[\overline{\mathbf{W}}^D \mathbf{S}^x\right] + 2\text{Tr}\left[\bar{\mathbf{d}}^L \cdot \mathbf{r}^x\right]\end{aligned} \tag{106}$$

The GFN-xTB gradient is written as [18]

$$\frac{dE_{\text{xTB}}}{dx} = \text{Tr}[\mathbf{P}\mathbf{H}^x] + E^x_{\text{h2}} + E^x_{\text{h3}} \tag{107}$$

E Auxiliary basis set for density matrix projection

The basis set file used to produced the projected density matrix auxiliary targets, reported in the NWChem format:

```
#BASIS SET: (30,30p,30d) -> [30,30p,30d]
HCONFSCl SPD
2.560000000000e+02 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1.280000000000e+02 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
6.400000000000e+01 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
3.200000000000e+01 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1.600000000000e+01 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
8.000000000000e+00 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
4.000000000000e+00 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
2.000000000000e+00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1.333333333333e+00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1.000000000000e+00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
8.000000000000e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
6.666666666667e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
5.714285714286e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
5.000000000000e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
4.444444444444e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
4.000000000000e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
3.636363636364e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
3.333333333333e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
3.076923076923e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
2.857142857143e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
2.666666666667e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
2.500000000000e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
2.352941176471e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0
2.222222222222e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0 0.0
2.105263157895e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0 0.0
2.000000000000e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0 0.0
1.904761904762e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0 0.0
1.818181818182e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0 0.0
```

```
1.739130434783e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -1.0
1.666666666667e-01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0
```

Appendix References

[1] Raghunathan Ramakrishnan et al. “Quantum chemistry structures and properties of 134 kilo molecules”. In: *Sci. Data* 1.1 (2014), pp. 1–7.

[2] Zhuoran Qiao et al. “OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features”. In: *The Journal of Chemical Physics* 153.12 (2020), p. 124111.

[3] You-Sheng Lin et al. “Long-range corrected hybrid density functionals with improved dispersion corrections”. In: *Journal of Chemical Theory and Computation* 9.1 (2013), pp. 263–272.

[4] Florian Weigend and Reinhart Ahlrichs. “Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy”. In: *Phys. Chem. Chem. Phys.* 7 (2005).

[5] Robert Polly et al. “Fast Hartree-Fock theory using local density fitting approximations”. In: *Mol. Phys.* 102.21-22 (Nov. 2004), pp. 2311–2321. ISSN: 0026-8976. DOI: `10.1080/0026897042000274801`. URL: `http://www.tandfonline.com/doi/abs/10.1080/0026897042000274801`.

[6] Florian Weigend. “Hartree-Fock exchange fitting basis sets for H to Rn”. In: *J. Comput. Chem.* 29 (2008), pp. 167–175.

[7] Lee-Ping Wang and Chenchen Song. “Geometry optimization made simple with translation and rotation coordinates”. In: *The Journal of chemical physics* 144.21 (2016), p. 214108.

[8] *Semiempirical Extended Tight-Binding Program Package*. `https://github.com/grimme-lab/xtb`. 2020, accessed July 14, 2020.

[9] Jan Gerit Brandenburg et al. “B97-3c: A revised low-cost variant of the B97-D density functional method”. In: *The Journal of chemical physics* 148.6 (2018), p. 064104.

[10] Frederick Manby et al. “entos: A Quantum Molecular Simulation Package”. In: *ChemRxiv preprint 10.26434/chemrxiv.7762646.v2* (2019). URL: `https://chemrxiv.org/articles/entos_A_Quantum_Molecular_Simulation_Package/7762646`.

[11] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[12] Petar Veličković et al. “Graph Attention Networks”. In: *International Conference on Learning Representations*. 2018.

[13] Juho Lee et al. “Set transformer: A framework for attention-based permutation-invariant neural networks”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3744–3753.

[14] Zhao Chen et al. “Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 794–803.

[15] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[16] Priya Goyal et al. “Accurate, large minibatch sgd: Training imagenet in 1 hour”. In: *arXiv preprint arXiv:1706.02677* (2017).

[17] Jan R. Magnus. “On Differentiating Eigenvalues and Eigenvectors”. In: *Econometric Theory* 1.2 (1985), pp. 179–191. DOI: `10.1017/S0266466600011129`.

[18] Stefan Grimme, Christoph Bannwarth, and Philip Shushkov. “A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (Z=1–86)”. In: *J. Chem. Theory Comput.* 13.5 (2017), pp. 1989–2009.