A FUNCTION-CENTRIC GRAPH NEURAL NETWORK APPROACH FOR PREDICTING ELECTRON DENSITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Electronic structure predictions are relevant for a wide range of applications, from drug discovery to materials science. Since the cost of purely quantum mechanical methods can be prohibitive, machine learning surrogates are used to predict the result of these calculations. This work introduces the Basis Overlap Architecture (BOA), an equivariant graph neural network architecture based on a novel message passing scheme that utilizes the overlap matrix of the basis functions used to represent the ground state electron density. BOA is evaluated on QM9 and MD density datasets, surpassing the previous state-of-the-art in predicting accurate electron densities. Excellent generalization to larger molecules of up to nearly 200 atoms is demonstrated using a model trained only on QM9 molecules of up to 9 heavy atoms.

1 INTRODUCTION

Accurate electronic structure predictions are crucial for the development of new catalysts, improved batteries or more specific drugs. Today's gold standard for reasonably sized systems is Kohn-Sham density functional theory (KS-DFT). It accounts for a significant fraction of worldwide supercomputing time, and three of its cornerstone methods are amongst the ten most cited publications of all times and fields (Van Noorden, 2025). Still, its computational cost prohibits routine use on large systems, or in very high throughput scenarios. In response, machine learning surrogates are developed to either circumvent or speed up KS-DFT calculations.

These methods range from property prediction (directly predicting observables from molecular geometry (Batatia et al., 2022; Ko et al., 2023; Kozinsky et al., 2023; Simeon & Fabritiis, 2023; Batatia et al., 2025; Liao et al., 2023; Anstine et al., 2025; Wood et al., 2025)) to incorporating more physical knowledge into the model. Examples for the latter include Hamiltonian prediction (Zhang et al., 2024a; Yuan et al., 2024; Wang et al., 2024; Qian et al., 2025; Luo et al., 2025), mimicking the self-consistent iterations of KS-DFT (Song & Feng, 2024), or predicting the electron density using orbital-free schemes (Zhang et al., 2024b; Remme et al., 2025). Directly predicting the electron density, as done here, lies between these extremes. The ground state electron density is an observable of central interest as it uniquely determines all ground state molecular properties in theory (Hohenberg & Kohn, 1964), and many properties of interest can be derived from the density in practice. Another use for electron densities is to reduce the number of self-consistent field iterations required in KS-DFT (Koker et al., 2024; Sunshine et al., 2023). Even a single KS-DFT diagonalization step can be enough to reach chemically accurate energies relative to a KS self-consistent solution (Jørgensen & Bhowmik, 2022; Li et al., 2025).

Most previous work on directly predicting ground state electron densities can be broadly grouped into two categories based on the representation used: The first class is based on a representation of the density in a typically atom-centered basis (Cuevas-Zuviría & Pacios, 2021; Rackers et al., 2023; del Rio et al., 2023; Elsborg et al., 2025; Qiao et al., 2022; 2020; Fu et al., 2024; Cheng & Peng, 2023; Kim & Ahn, 2024; Mitnikov & Jacobson, 2024; Febrer et al., 2025). A model would then predict the coefficients of an expansion of the ground state electron density in the given basis. This approach has the advantage of being relatively scalable, as the density only needs to be evaluated on a volumetric grid at the end of the model, if at all. The choice of basis is however crucial, and to achieve high accuracy a very large number of basis functions is typically required. This problem was partially addressed by Fu et al. (2024), where virtual nodes are employed, and the basis functions

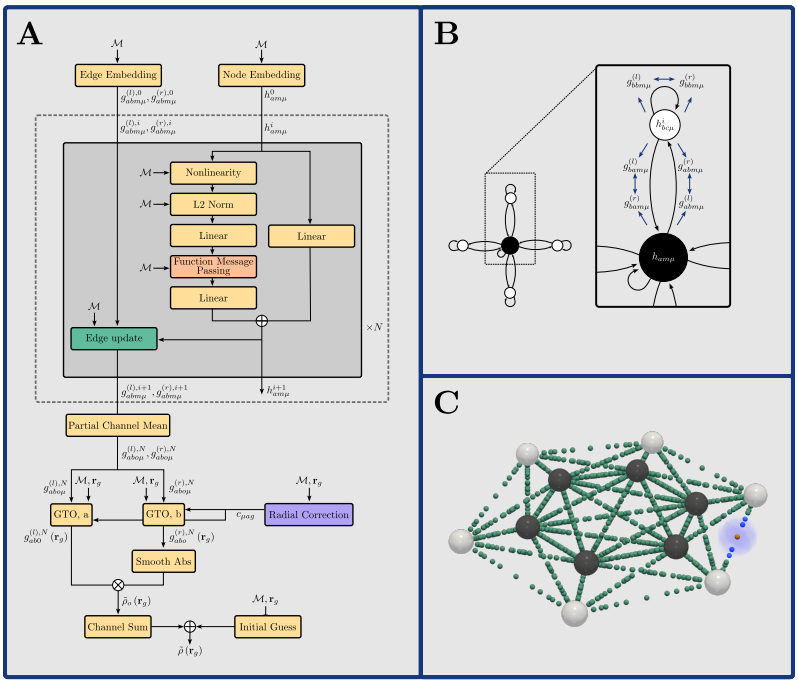

Figure 1: **The Basis Overlap Architecture (BOA).** An overview of BOA is given in **(A)**. The input molecular geometry is fed into the edge and node embedding layers. The node embeddings are updated using BOA blocks, which contain a function message passing step to facilitate communication between nodes. The edge features are modified using the edge update block, which uses the current edge and node features to calculate new edge features. The output of the BOA backbone consists of coefficients used to expand the density in atom-centered Gaussian-type basis functions as described in Section 2.1. In the partial channel mean the number of channels is reduced by taking the mean of groups of channels. $\mathcal{M}$ denotes the molecular geometry and $\mathbf{r}_g$ are the grid positions with $g \in \{1, \ldots, N^g\}$ and N^g being the number of grid points. **(B)** shows the flow of information in the BOA message passing block. While the edge features are updated using the node features, there is no flow of information from the edge features to the node features. Every edge in the graph has two edge features with superscript (l) and (r) respectively, corresponding to the two nodes the edge connects. Self-loops are included in the graph. **(C)** shows a benzene molecule from the MD dataset with carbon atoms (black) and hydrogen atoms (white). Shown as green spheres are the centers of the Gaussian parts of the products of the atom-centered basis functions. The smaller `def2-SVP` (Weigend & Ahlrichs, 2005) basis set is used here for better visualization. A product of $l = 0$ basis functions from two different hydrogen atoms is shown in blue. The center of the product of these two basis functions is shown in yellow, lying in the middle between the two atoms on which the original basis functions are centered. Since the products of basis functions are well distributed in space, a highly accurate representation of the density can be achieved without the need for floating basis functions or virtual nodes.

adapted in a fine-tuning step to mitigate the impact of a suboptimal basis. Virtual nodes are placed in the molecule, e.g. at the midpoint of bonds, and additional localized basis functions are centered at these nodes. Another approach to address this problem was taken by Elsborg et al. (2025), where floating basis functions are employed. These floating orbitals are no longer centered at the atoms of the molecule and instead the basis function positions are predicted per molecule individually, enabling a much more flexible representation of the density.

The second class of methods works directly with a representation of the electron density on a volumetric grid (Jørgensen & Bhowmik, 2022; Li et al., 2025; Koker et al., 2024; Gong et al., 2019; Sunshine et al., 2023). While this approach avoids basis set related inaccuracies, it is typically much more memory intensive, due to the large number of grid points required to accurately represent the density.

Contributions This work falls into the first category of methods, i.e. we represent the density in an atom-centered basis and the model predicts coefficients in this basis. In contrast to previous work, we, however, choose to represent the density in a quadratic expansion of the basis functions, inspired by the internal representation of the density in KS-DFT calculations using a density matrix. A product of two atom-centered Gaussian-type basis functions, as employed in this work, will be centered between the two atoms, avoiding the need for floating orbitals or virtual nodes. For a benzene molecule, this is shown in Figure 1C. This work avoids explicitly predicting coefficients for each pair of basis functions, effectively employing a low-rank representation of blocks of the density matrix, without constructing the full coefficient matrix at any point.

To effectively utilize this representation, we introduce the Basis Overlap Architecture (BOA), a novel equivariant message passing neural network architecture. The full BOA architecture is shown in Figure 1A. The fundamental idea of BOA is to imbue the model with information about the underlying basis by *interpreting*, at appropriate points in the model, the internal features *as functions* represented in the given basis. Most notably the message passing formulation utilizes the overlap between functions represented in atom-centered basis functions at different atoms to facilitate communication between nodes. During message passing each message is transformed from the basis of the sending node to the basis of the receiving node, essentially giving the best fit of the message in the basis of the receiving node. This instills the model with information not only about the underlying basis, but also the geometry of the molecule, as the overlap between basis functions centered at different atoms depends on their relative position. This novel approach to message passing is shown to be very effective for the task of predicting the electron density.

The internal features of BOA are separated into node and edge features, with the bulk of the computation happening in the node features. This separation avoids the high computational costs of basing the full procedure solely on edge features, while still allowing for a rich representation needed to accurately predict the density. The flow of information between node and edge features is shown in Figure 1B.

BOA is evaluated on electron densities generated from the widely used QM9 dataset (Jørgensen & Bhowmik, 2022; Li et al., 2025; Ruddigkeit et al., 2012; Ramakrishnan et al., 2014) and a molecular dynamics (MD) dataset of small organic molecules (Cheng & Peng, 2023; Bogojeski et al., 2020; Brockherde et al., 2017). On all evaluated datasets BOA outperforms previous state-of-the-art methods by a significant margin.

2 METHODS

To achieve the desired inductive bias, internal features are designed to be interpreted as functions represented in an atom-centered basis. At each point the node features are given by $h_{am\mu}$, where $a \in \mathcal{N}$ is the node with atom type Z_a, $\mathcal{N}$ is the set of all nodes, $m \in \{1, \dots, N^c\}$ is the feature channel, and $\mu \in \{1, \dots, N^B_{Z_a}\}$ is the basis index. We use Gaussian-type ("GTO") basis functions, specifically a fully uncontracted version of `def2-QZVPPD` (Weigend & Ahlrichs, 2005). A correction to the radial part is predicted by the model; see Section 2.9 for details. The internal features $h_{am\mu}$ are understood as expansion coefficients of N^c functions of space

$$h_m(\mathbf{r}) = \sum_{a \in \mathcal{N}} h_{am}(\mathbf{r}) = \sum_{a \in \mathcal{N}} \sum_{\mu} h_{am\mu} \omega_\mu^{Z_a}(\mathbf{r} - \mathbf{r}_a), \tag{1}$$

consisting of a sum over functions $h_{am}(\mathbf{r})$ localized at the node positions $\mathbf{r}_a$. Each of these functions in turn is represented in the atom type specific basis $\omega_\mu^{Z_a}(\mathbf{r})$. The same principle holds for the edge features $g^{(l)}_{abm\mu}$ and $g^{(r)}_{abm\mu}$, where $(a, b) \in \mathcal{E}_\text{e}$ is a directed edge in the graph, with $\mathcal{E}_\text{e}$ being the set of all edges. Superscripts l and r denote left and right, respectively, for reasons that are clear from Equation 2. The l edge features are interpreted as functions localized at node a, while the r edge features are interpreted as functions localized at node b.

To guarantee rotational equivariance the choice of basis functions is essential. Choosing basis functions where the angular part is given by spherical harmonics ensures that the basis function coefficients transform according to irreducible representations of SO(3), making it possible to construct equivariant operations. BOA is fully equivariant under rotations and translations by construction, details are shown in Appendix Section A.6.

2.1 Density Representation

Previous work explored different ways to represent the density, either in a basis or directly on the grid. We choose a representation in an atom-centered basis. In contrast to previous work, we however do not expand the density (Cuevas-Zuviría & Pacios, 2021; Rackers et al., 2023; del Rio et al., 2023; Qiao et al., 2022; Fu et al., 2024; Cheng & Peng, 2023) or its square root (Mitnikov & Jacobson, 2024) directly as a linear expansion of the basis functions $\rho(\mathbf{r}) = \sum_{a\in\mathcal{N}}\sum_{\mu} p_{a\mu}\omega_{\mu}^{Z_a}(\mathbf{r}-\mathbf{r}_a)$. Instead, we choose a quadratic expansion, inspired by the natural expansion of the density in squared orbital functions in KS-DFT:

$$\rho(\mathbf{r}) = \sum_{a\in\mathcal{N}} \hat{g}_a^{(l)}(\mathbf{r})\hat{g}_a^{(r)}(\mathbf{r}) + \sum_{(a,b)\in\mathcal{E}_e}\sum_{o}^{N^o} g_{abo}^{(l)}(\mathbf{r})g_{abo}^{(r)}(\mathbf{r}), \tag{2}$$

where N^o is the number of function pairs per edge and $g_{abo}^{(l)}(\mathbf{r})$, $g_{abo}^{(r)}(\mathbf{r})$, $\hat{g}_a^{(l)}(\mathbf{r})$ and $\hat{g}_a^{(r)}(\mathbf{r})$ are expanded in the localized basis functions $\omega_{\mu}^{Z_a}(\mathbf{r})$ and $\omega_{\mu}^{Z_b}(\mathbf{r})$:

$$g_{abo}^{(l)}(\mathbf{r}) = \sum_{\mu} g_{abo\mu}^{(l)}\omega_{\mu}^{Z_a}(\mathbf{r}-\mathbf{r}_a), \quad g_{abo}^{(r)}(\mathbf{r}) = \sum_{\mu} g_{abo\mu}^{(r)}\omega_{\mu}^{Z_b}(\mathbf{r}-\mathbf{r}_b), \tag{3}$$

$$\hat{g}_a^{(l)}(\mathbf{r}) = \sum_{\mu} \hat{g}_{a\mu}^{(l)}\omega_{\mu}^{Z_a}(\mathbf{r}-\mathbf{r}_a), \qquad \hat{g}_a^{(r)}(\mathbf{r}) = \sum_{\mu} \hat{g}_{a\mu}^{(r)}\omega_{\mu}^{Z_a}(\mathbf{r}-\mathbf{r}_a). \tag{4}$$

This representation can be rewritten by defining $\Gamma_{ab\mu\nu} = \sum_{o}^{N^o} g_{abo\mu}^{(l)} g_{abo\nu}^{(r)} + \delta_{ab}\hat{g}_{a\mu}^{(l)}\hat{g}_{a\nu}^{(r)}$, with $\delta_{ab} = 1$ if $a = b$ and 0 otherwise, resulting in $\rho(\mathbf{r}) = \sum_{(a,b)\in\mathcal{E}_e}\Gamma_{ab\mu\nu}\omega_{\mu}^{Z_a}(\mathbf{r}-\mathbf{r}_a)\omega_{\nu}^{Z_b}(\mathbf{r}-\mathbf{r}_b)$. Each $\Gamma_{ab\mu\nu}$ can be interpreted as a block of the full density matrix $\Gamma_{\mu\nu}$, where the indices μ and ν are understood to run over all basis functions of all atoms. Using the full density matrix, the density is represented by $\rho(\mathbf{r}) = \Gamma_{\mu\nu}\bar{\omega}_{\mu}(\mathbf{r})\bar{\omega}_{\nu}(\mathbf{r})$, where $\bar{\omega}(\mathbf{r})$ is the concatenation of the basis functions of all atoms. This is exactly the representation of the density used in KS-DFT calculations. The full density matrix $\Gamma_{\mu\nu}$ is never explicitly constructed in BOA, instead the functions $g_{abo}^{(l)}(\mathbf{r})$ and $g_{abo}^{(r)}(\mathbf{r})$ are evaluated on the grid and the density is obtained using Equation 2, avoiding the costly evaluation of all pairwise products of basis functions on the grid. Equation 2 amounts to evaluating a density given by a low-rank representation of each block $\Gamma_{ab\mu\nu}$ of the density matrix, where the rank is given by N^o.

The expansion coefficients $g_{abo\mu}^{(l)}$ and $g_{abo\mu}^{(r)}$ are predicted from the molecule geometry $\mathcal{M}$ by BOA as described in the following sections. The self-loop coefficients $\hat{g}_{a\mu}^{(l)}$ and $\hat{g}_{a\mu}^{(r)}$ are used to represent an initial guess for the density, which is added to the density predicted by BOA. Taking the smooth absolute value of one of the functions in each pair, as described in Appendix A.8, proved beneficial.

2.2 Initial Guess of the Electron Density

BOA uses an initial guess of the ground state electron density based on the atom types of the nodes. After pre-training the initial guess, only an offset to that guess is learned. The learned guess amounts to one pair of edge functions as described in Section 2.1 per atom. Consequently, the initial guess for the density is given by $\sum_{a\in\mathcal{N}}\hat{g}_a^{(l)}(\mathbf{r})\hat{g}_a^{(r)}(\mathbf{r})$ and is part of the predicted electron density as shown in Equation 2. The initial guess coefficients are concatenated to the output of the BOA model in the channel dimension before evaluating the density, resulting in the initial guess being added to the density predicted by the BOA model. The initial guess coefficients are pre-trained for 1000 steps, but are not fixed after that, i.e. the initial guess is refined during the full training process.

2.3 Node and Edge Embeddings

The node and edge embeddings are used to set the initial node and edge features respectively. Both are lookup tables, which map the atom type Z_a of the node a and the atom types of the nodes connected by the edge (a, b) to a learned set of features. To keep equivariance under rotations, only the coefficients corresponding to $l = 0$ basis functions are set in the embeddings, while the rest is initialized to zero. The node features are therefore initialized as

$$h_{am\mu} = \begin{cases} W_{Z_a m\mu}^{(\mathrm{n})}, \text{ if } \omega_{\mu}^{Z_a}(\mathbf{r}) \text{ is an } l_{Z_a\mu} = 0 \text{ basis function} \\ 0, \text{ otherwise,} \end{cases} \tag{5}$$

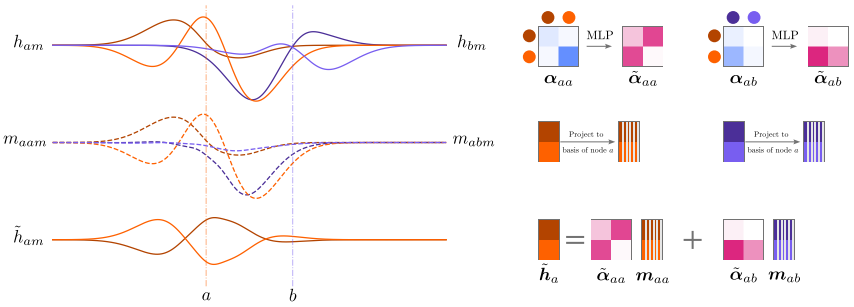

Figure 2: **1D illustration of basis overlap message passing.** Two nodes a and b are shown, each with two feature functions (blue and orange) centered at their respective positions. These functions are expanded in a local basis build from monomials multiplied with Gaussians. The first line shows the original feature functions h_{am} and h_{bm}. The second line shows the messages send from node a and node b to node a. These are the projection of the respective feature functions to the basis of node a. The third line shows the new features $\tilde{h}_{am}$ after message passing which are a weighted sum of all incoming messages.

where $W^{(\mathrm{n})}_{Z_a m\mu}$ is learned during training and $l_{Z_a\mu}$ is labeling the irreducible representation of the basis function $\omega^{Z_a}_{\mu}(\mathbf{r})$. The edge features are initialized similarly as

$$g^{(l)}_{abm\mu} = \begin{cases} W^{(\mathrm{e},l)}_{Z_a Z_b m\mu}, & \text{if } \omega^{Z_a}_{\mu}(\mathbf{r}) \text{ is an } l_{Z_a\mu} = 0 \text{ basis function} \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

$$g^{(r)}_{abm\mu} = \begin{cases} W^{(\mathrm{e},r)}_{Z_a Z_b m\mu}, & \text{if } \omega^{Z_b}_{\mu}(\mathbf{r}) \text{ is an } l_{Z_b\mu} = 0 \text{ basis function} \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where $W^{(\mathrm{e},l)}_{Z_a Z_b m\mu}$ and $W^{(\mathrm{e},r)}_{Z_a Z_b m\mu}$ are again learned.

2.4 Basis Overlap Message Passing

The fundamental principle behind the message passing mechanism introduced in this section is that, each feature channel should be interpretable as a function represented in a given basis. To respect this interpretation, a basis change is employed to transform each message from the basis of the sending node to the basis of the receiving node, a concept similar to frame-to-frame transitions employed in local canonicalization schemes (Lippmann et al., 2025). Inspired by the attention mechanism (Vaswani et al., 2017), the messages are weighted by attention weights, which are calculated from the overlap integrals between the features of the sending and receiving node. The full message passing mechanism is shown in Figure 3B and a one dimensional illustration is given in Figure 2.

BOA operates using two different cutoff radii $r_{\mathrm{e}}, r_{\mathrm{mp}} \in \mathbb{R}$, with r_{e} being a smaller cutoff defining the edges $\mathcal{E}_{\mathrm{e}}$ used for the edge features and in the edge update module defined in 2.8, and r_{mp} being a larger cutoff defining the edges $\mathcal{E}_{\mathrm{mp}}$ used in the message passing step.

The first step in transforming the features of node b into the basis of node a is calculating the overlap integrals between these features and the basis functions of node a. This overlap $o_{abm\mu}$ is given by

$$o_{abm\mu} = \int d\mathbf{r}\, \omega^{Z_a}_{\mu}(\mathbf{r} - \mathbf{r}_a) h_{bm}(\mathbf{r}) = \int d\mathbf{r}\, \omega^{Z_a}_{\mu}(\mathbf{r} - \mathbf{r}_a) \sum_{\nu} \omega^{Z_b}_{\nu}(\mathbf{r} - \mathbf{r}_b) h_{bm\nu} \tag{8}$$

$$= \sum_{\nu} W^{ab}_{\mu\nu} h_{bm\nu}, \tag{9}$$

where $W^{ab}_{\mu\nu} = \int d\mathbf{r}\, \omega^{Z_a}_{\mu}(\mathbf{r} - \mathbf{r}_a)\omega^{Z_b}_{\nu}(\mathbf{r} - \mathbf{r}_b)$ is the overlap matrix between the basis functions of node a and node b. From these overlap integrals, the representation of the features of node b in the basis of node a can be obtained by multiplying with the inverse of the overlap matrix $(W^{aa})^{-1}_{\mu\nu}$ of

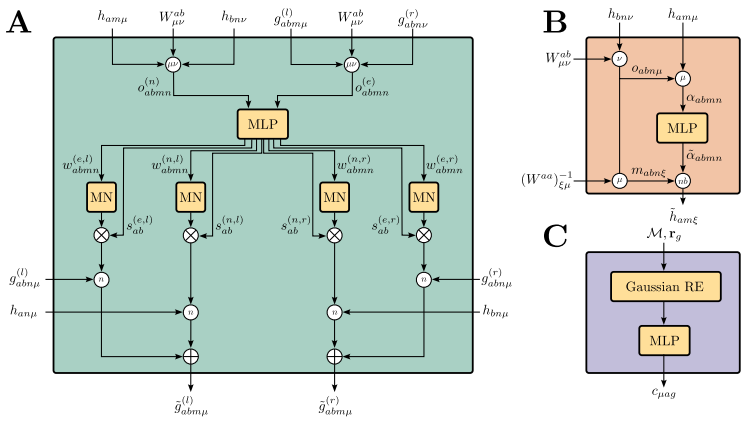

Figure 3: **Subblocks of the BOA architecture.** An encircled index in the graph indicates a contraction over the corresponding dimension, i.e. multiplication of the inputs and summation over the given index. **(A)** The new edge features $\tilde{g}^{(l)}_{abm\mu}$ and $\tilde{g}^{(r)}_{abm\mu}$ are generated as a superposition of the old edge features $g^{(l)}_{abm\mu}$ and $g^{(r)}_{abm\mu}$ and the node features $h_{am\mu}$ and $h_{bn\mu}$. The weights of this superposition are generated by passing the overlap integrals of the edge and node features through an MLP. After the MLP the weights are normalized using the matrix normalization (MN) block and scaled by the predicted factors $s^{(\cdot,\cdot)}_{ab}$. The resulting weight matrices are then used to linearly mix the edge and node features. **(B)** Incoming messages are calculated by transforming the feature functions from the basis of the sending node to the basis of the receiving node. Attention weights $\tilde{\alpha}_{abmn}$ are calculated from the overlap between the feature functions of the two nodes and used to weight the messages. **(C)** A correction to the radial part of the basis functions is learned for each atom type separately by a small MLP. The radius is passed through a Gaussian radial embedding (Passaro & Zitnick, 2023) before being fed into the MLP. The output c_μ of the MLP is then used to scale the radial part of the basis functions by $1 + c_\mu$.

the basis functions of node a, resulting in the message $m_{abm\mu}$ from node b to node a

$$m_{abm\mu} = \sum_{\nu} (W^{aa})^{-1}_{\mu\nu}\, o_{abm\nu}. \tag{10}$$

These messages $m_{abm\mu}$ are the coefficients in the basis of node a that best represent the features of node b in the least squares sense. A derivation of this result can be found in Appendix A.5.

To weight the messages, an attention matrix $\tilde{\alpha}_{abmn}$ is calculated for each edge (a, b). Each entry in this attention matrix describes the weight of the message from channel n of node b to channel m of node a. The attention matrix is calculated from the overlap between the feature functions of the two nodes

$$\alpha_{abmn} = \int d\mathbf{r}\; h_{am}(\mathbf{r}) h_{bn}(\mathbf{r}) = \int d\mathbf{r}\; \sum_{\mu\nu} h_{am\mu}\omega^a_\mu(\mathbf{r} - \mathbf{r}_a) h_{bn\nu}\omega^b_\nu(\mathbf{r} - \mathbf{r}_b) \tag{11}$$

$$= \sum_{\mu\nu} h_{am\mu} W^{ab}_{\mu\nu} h_{bn\nu} = \sum_{\mu} h_{am\mu} o_{abn\mu}. \tag{12}$$

This feature overlap matrix α_{abmn} is processed by a multi-layer perceptron to produce the attention weights $\tilde{\alpha}_{abmn}$, which determine the contribution of messages from neighboring nodes. These weighted messages are then aggregated to update the features of node a

$$\tilde{h}_{am\mu} = \sum_{b \in \mathcal{N}_{\text{mp}}(a)} \sum_{n} \tilde{\alpha}_{abmn} m_{abn\mu}, \tag{13}$$

where $\mathcal{N}_{\text{mp}}(a) = \{b \mid (a, b) \in \mathcal{E}_{\text{mp}}\}$ is the set of neighboring nodes of node a defined by the message passing edges $\mathcal{E}_{\text{mp}}$. Since the edges $\mathcal{E}_{\text{mp}}$ include self-loops, the original features of each node are included in the message passing step, which amounts to a residual connection weighed by the attention weights $\tilde{\alpha}_{aamn}$.

2.5 NONLINEARITY

Searching for a suitable nonlinearity for features that should be interpretable as functions, one faces similar problems as for nonlinearities applied in equivariant neural networks. Here features are grouped together to form tensors with known transformation behavior under rotations, so that the nonlinearity can not be applied independently to each feature. In our case, the features are not only grouped to form tensors, but also form larger groups that each represent a function.

One solution in the case of equivariant networks are gated nonlinearities (Weiler et al., 2018), where first scalar features are computed to which some nonlinear function is applied. The resulting scalar features are then used to scale the tensors, i.e. each group of features forming a tensor is scaled with the same scalar. In a similar approach we first calculate scalar features

$$l_{amn} = \int d\mathbf{r} \int d\mathbf{r}' \, \frac{h_{am}(\mathbf{r}) h_{an}(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|} = \sum_{\mu\nu} h_{am\mu} C^{aa}_{\mu\nu} h_{an\nu}, \tag{14}$$

where $C^{aa}_{\mu\nu} = \int d\mathbf{r} \int d\mathbf{r}' \, \frac{\omega^{Z_a}_{\mu}(\mathbf{r}-\mathbf{r}_a)\omega^{Z_a}_{\nu}(\mathbf{r}'-\mathbf{r}_a)}{\|\mathbf{r}-\mathbf{r}'\|}$ is the Coulomb matrix, which can be generated for Gaussian-type basis functions using the PySCF package (Sun et al., 2020). The resulting scalar features l_{amn} are flattened and passed through an MLP. The result

$$\mathbf{w}_a = \mathrm{MLP}_{Z_a}(\mathbf{l}_a) \tag{15}$$

is reshaped to the original shape of l_{amn} and then used to linearly transform the features of node a

$$\tilde{h}_{am\mu} = \sum_n w_{amn} h_{an\mu}. \tag{16}$$

Z_a denotes the atom type of node a, i.e. separate MLPs are learned for each atom type.

2.6 L2 NORMALIZATION

The normalization also aims to respect the function nature of the features, so the L2 norm of each of the per-atom channel functions is normalized. The norm is calculated as $n_{am} = \sqrt{\int d\mathbf{r} \; (h_{am}(\mathbf{r}))^2} = \sqrt{\sum_{\mu\nu} h_{am\mu} W^{aa}_{\mu\nu} h_{am\nu}}$ and the features are normalized by $\tilde{h}_{am\mu} = \frac{h_{am\mu}}{n_{am}+\epsilon}$ where ϵ is chosen as 10^{-6} to avoid division by zero.

2.7 LINEAR LAYERS

Standard equivariant linear layers, as implemented in the e3nn package (Geiger & Smidt, 2022), are applied to the features of each channel and node. This means that only tensors of the same type are mixed and biases are only applied to scalars to preserve equivariance. The linear layers used depend on the atom type of each node, i.e. separate parameters are learned for each atom type. After the application of these equivariant linear layers, the channels are mixed by a weight matrix, which again depends on the atom type of the node. This results in

$$\tilde{h}^{Z_a}_{am\mu} = \sum_{\mu} \sum_{\nu} W^{Z_a}_{mn} W^{Z_a,(\mathrm{eq})}_{\mu\nu} h_{an\nu} \tag{17}$$

where $W^{Z_a}_{mn}$ is the weight matrix for the mixing of the channels and $W^{Z_a,(\mathrm{eq})}_{\mu\nu}$ is the weight matrix for the equivariant linear layer. While the linear layer used here is fully equivariant, it arguably does not fully respect the function interpretation of the features. Initial experiments with linear layers that respect the function interpretation, i.e. by applying an integration $\int d\mathbf{r} \; w_{mn}(\mathbf{r}) h_n(\mathbf{r})$ where $w_{mn}(\mathbf{r})$ are learned functions represented in some basis, showed strong instabilities in training. Since the linear layers shown here worked well in practice, we decided to use them in the final architecture, opting for stability over additional inductive bias.

2.8 EDGE UPDATE

The edge functions $g^{(l)}_{abm}(\mathbf{r}), g^{(r)}_{abm}(\mathbf{r})$ are updated after each BOA block. The flow of information is illustrated in Figure 1B. The node features $h_{am}(\mathbf{r})$ are used together with the old edge features

$g^{(l)}_{abm}(\mathbf{r}), g^{(r)}_{abm}(\mathbf{r})$ to generate new edge features $\tilde{g}^{(l)}_{abm}(\mathbf{r}), \tilde{g}^{(r)}_{abm}(\mathbf{r})$. There is however no flow of information from the edge features to the node features. The full edge update procedure is shown in Figure 3A. Intermediate features are generated using the overlap integrals of edge and node features

$$o^{(n)}_{abmn} = \int d\mathbf{r}\; h_{am}(\mathbf{r}) h_{bn}(\mathbf{r}) = \sum_{\mu\nu} h_{am\mu} W^{ab}_{\mu\nu} h_{bn\nu}, \tag{18}$$

$$o^{(e)}_{abmn} = \int d\mathbf{r}\; g^{(l)}_{abm}(\mathbf{r}) g^{(r)}_{abn}(\mathbf{r}) = \sum_{\mu\nu} g^{(l)}_{abm\mu} W^{ab}_{\mu\nu} g^{(r)}_{abn\nu}, \tag{19}$$

These intermediate features are flattened and passed through an MLP to obtain weights $w^{(n,l)}_{abmn}$, $w^{(n,r)}_{abmn}$, $w^{(e,l)}_{abmn}$, $w^{(e,r)}_{abmn}$, $s^{(n,l)}_{ab}$, $s^{(n,r)}_{ab}$, $s^{(e,l)}_{ab}$, $s^{(e,r)}_{ab}$. The weight matrices $\mathbf{w}^{(n,l)}_{ab}$, $\mathbf{w}^{(n,r)}_{ab}$, $\mathbf{w}^{(e,l)}_{ab}$, $\mathbf{w}^{(e,r)}_{ab}$ are normalized by

$$\tilde{\mathbf{w}}^{(\cdot,\cdot)}_{ab} = \frac{\mathbf{w}^{(\cdot,\cdot)}_{ab}}{\left\| \mathbf{w}^{(\cdot,\cdot)}_{ab} \right\|_f + \epsilon} \gamma^{(\cdot,\cdot)}, \tag{20}$$

where $(\cdot,\cdot)$ denotes either (n,l), (n,r), (e,l) or (e,r) and $\gamma^{(\cdot,\cdot)}$ is a learned scalar factor. The normalization is computed in the Frobenius norm $\|\cdot\|_f$. Using these weights the new edge features

$$\tilde{g}^{(\cdot)}_{abm\mu} = \sum_n s^{(e,\cdot)}_{ab} \tilde{w}^{(e,\cdot)}_{abmn} g^{(\cdot)}_{abn\mu} + \sum_n s^{(n,\cdot)}_{ab} \tilde{w}^{(n,l)}_{abmn} h_{\star n\mu} \tag{21}$$

are generated, where $\star = a$ for the l features and $\star = b$ for the r features.

2.9 Radial Correction Factor

While internally all features are represented in a basis consisting of Gaussian-type basis functions, for the final density representation an additional correction factor to the radial part of the basis functions is learned. For each atom type Z separately a small MLP is used to predict a correction factor c_μ based on the radius $r = \|\mathbf{r}\|$. As shown in Figure 3C the radius is passed through a Gaussian radial embedding (Passaro & Zitnick, 2023) before being fed into the MLP. The corrected basis functions $\tilde{\omega}^Z_\mu(\mathbf{r})$ are then given by $\tilde{\omega}^Z_\mu(\mathbf{r}) = \omega^Z_\mu(\mathbf{r})\,(1 + c_\mu(r))$. The impact of this radial correction factor is evaluated in Appendix Section A.1.1.

3 Experiments

The performance of BOA is evaluated on two electron density datasets based on QM9 (Jørgensen & Bhowmik, 2022; Li et al., 2025; Ruddigkeit et al., 2012; Ramakrishnan et al., 2014) and on a dataset based on MD trajectories (Cheng & Peng, 2023; Bogojeski et al., 2020; Brockherde et al., 2017). These datasets provide ground state electron densities for given geometries on a volumetric grid. Two versions of BOA are evaluated, a small version and a large version, differing only in the number of grid points used to evaluate the loss during training, and the batch size. The small version uses 5000 grid points and a batch size of 12 while the large version uses 6000 grid points and a batch size of 24. Using these settings, the small version fits on a single A100 GPU with ∼40GB of memory while the large version requires an H100 GPU with ∼94GB of memory or some setup with similar memory. All hyperparameters and training details are listed in Appendix A.7.

Additionally, the generalization capabilities of BOA are evaluated on larger molecules with up to almost 200 atoms from the QMugs dataset (Isert et al., 2022) not seen during training. Ground-truth ground state electron densities are generated using the same method as used for the QM9 PySCF dataset (Li et al., 2025), the dataset on which the evaluated models were trained.

3.1 Evaluation on Small Molecules

As in previous works, the predicted electron density $\tilde{\rho}(\mathbf{r})$ is compared to the reference electron density $\rho(\mathbf{r})$ using the normalized mean absolute error $\mathrm{NMAE}(\tilde{\rho}, \rho) = \frac{\int d\mathbf{r}\; |\tilde{\rho}(\mathbf{r}) - \rho(\mathbf{r})|}{\int d\mathbf{r}\; |\rho(\mathbf{r})|}$. The integration is approximated on the full grid by summing over all grid points.

The two QM9 based datasets differ in the way the reference electron density was calculated, either using VASP (Jørgensen & Bhowmik, 2022) or PySCF (Li et al., 2025). BOA surpasses the previous state-of-the-art on both QM9 datasets, as shown in Table 1. The split between training, validation and test set is taken from Fu et al. (2024) for the QM9 VASP dataset and from Li et al. (2025) for the QM9 PySCF dataset.

Table 1: **Comparison of BOA with previous best methods on the QM9 charge density datasets.** Two datasets based on QM9 are evaluated, differing in the reference electron density calculation method. Errors are reported as NMAE [%]. For BOA the mean and standard error over three runs are reported for the small models. For the large models the mean and standard error over five runs are reported. Errors of eqDeepDFT, InfGCN, ChargE3Net, and SCDP are reproduced from Fu et al. (2024). The ResNet results are taken from Li et al. (2025).

NMAE [%]	VASP ground truth	PySCF ground truth
eqDeepDFT (Jørgensen & Bhowmik, 2022)	0.284	n/a
InfGCN (Cheng & Peng, 2023)	0.869	n/a
ChargE3Net (Koker et al., 2024)	0.196	n/a
SCDP (Fu et al., 2024)	0.178	n/a
ELECTRA (Elsborg et al., 2025)	0.177	n/a
ResNet (Li et al., 2025)	n/a	0.14
BOA small	0.1381 ± 0.0003	0.13 ± 0.01
BOA large	$\mathbf{0.1339 \pm 0.0005}$	$\mathbf{0.116 \pm 0.006}$

Additionally, the performance of BOA is evaluated on the MD dataset. As described in Cheng & Peng (2023), the MD dataset is curated from two sources. The ethanol, benzene, phenol and resorcinol data is taken from Bogojeski et al. (2020), while the ethane and malonaldehyde data is taken from Brockherde et al. (2017). BOA outperforms all previous methods on all molecules but one, where it matches the state-of-the-art, as shown in Table 2. Especially promising is the fact that no additional hyperparameter tuning was needed to achieve these results. The model is exactly the same as in the QM9 experiments, with the only change being a reduction in training steps to 200,000. Since only a training and test set is provided for the MD dataset, 10% of the training set is randomly sampled and used as a validation set to choose the best model during training.

Table 2: **Comparison of BOA with other methods on the MD charge density dataset.** Errors are reported as NMAE [%]. For BOA the mean and standard error of the mean over three runs are reported. Errors of the other models (InfGCN (Cheng & Peng, 2023), GPWNO (Kim & Ahn, 2024), SCDP (Fu et al., 2024), ELECTRA (Elsborg et al., 2025)) are reproduced from Elsborg et al. (2025).

NMAE [%]	ethanol	benzene	phenol	resorcinol	ethane	malonaldehyde
InfGCN	8.43	5.11	5.51	5.95	7.01	10.34
GPWNO	4.00	2.45	2.68	2.73	3.67	5.32
SCDP	2.34	1.13	1.29	1.35	2.05	2.71
ELECTRA	1.02	0.45	**0.56**	0.62	0.91	0.80
BOA small	$\mathbf{0.710 \pm 0.004}$	$\mathbf{0.361 \pm 0.003}$	$\mathbf{0.56 \pm 0.03}$	$\mathbf{0.371 \pm 0.004}$	$\mathbf{0.772 \pm 0.002}$	$\mathbf{0.61 \pm 0.01}$

3.2 Generalization to Larger Molecules

To evaluate the generalization performance of BOA to larger molecules, the QMugs dataset (Isert et al., 2022) is used. Ground-truth ground state electron densities for molecules with up to almost 200 atoms are calculated using PySCF (Sun et al., 2020) with the same settings as used for the QM9 PySCF dataset (Li et al., 2025). Two BOA models trained on only the QM9 PySCF dataset are evaluated on this new test set. One of the models uses a smaller message passing cutoff of $r_{\text{mp}} = 3$Å instead of $r_{\text{mp}} = 6$Å and a smaller edge feature cutoff of $r_{\text{e}} = 2$Å instead of $r_{\text{e}} = 3$Å. Additionally, we evaluate the ResNet model (Li et al., 2025) on this dataset for comparison. The time-scaling

behavior and the NMAE of all three models are shown in Figure 4. While the BOA model with the standard cutoffs performs worse than the ResNet model, the error of the BOA model with smaller cutoffs stays roughly constant over the evaluated molecule sizes and outperforms the ResNet model. This shows that to achieve good generalization from small to large molecules, it is beneficial to limit the field of view of the model. This may be because the contents of a large field of view differ significantly between small and large molecules, introducing a significant distribution shift, making it difficult for the model to generalize. Using a smaller field of view enables BOA to generalize well to large molecules despite being trained on only small molecules of up to 9 heavy atoms.

The time scaling behavior shown in Figure 4A also shows a second advantage of using smaller cutoffs. The BOA model with smaller cutoffs is significantly faster than both the ResNet model and the BOA model with larger cutoffs when evaluating larger molecules.

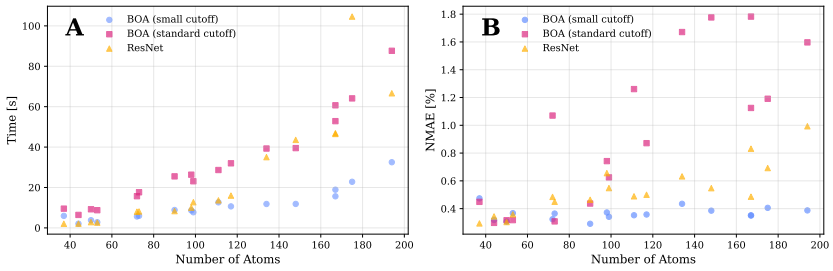

Figure 4: **Generalization of BOA to larger molecules.** Two BOA models trained on the QM9 PySCF data are evaluated on larger molecules from the QMugs dataset (Isert et al., 2022). One of the models uses a smaller message passing cutoff of $r_{\text{mp}} = 3$Å instead of $r_{\text{mp}} = 6$Å and a smaller edge feature cutoff of $r_{\text{e}} = 2$Å instead of $r_{\text{e}} = 3$Å. **(A)** shows the time scaling behavior of both models compared to the ResNet model (Li et al., 2025). **(B)** shows the NMAE $[\%]$ of both models on the QMugs dataset over the number of atoms in the molecules. While the BOA model with the larger cutoffs performs worse than the ResNet model, the BOA model with smaller cutoffs outperforms the ResNet model with accuracy staying roughly constant over the evaluated molecule sizes.

4 Discussion

Electronic structure calculations plays a fundamental role in computational chemistry, with a myriad of practical applications. Accelerating these calculations has the potential of enabling new applications which are currently not possible, and of cutting costs for what is already feasible. BOA takes a step towards this goal by introducing a novel architecture, permitting predictions of the electron density with unprecedented accuracy. These results are enabled by a novel message passing mechanism treating the internal features as functions additional to a representation of the density in a quadratic basis expansion. The excellent generalization capabilities of BOA to larger molecules enables its application to problems were full electronic structure calculations are especially costly.

While BOA already achieves state-of-the-art on the evaluated datasets, there are several avenues for future work. Both the QM9 and the MD dataset contain only relatively small organic molecules. Training on larger more diverse datasets will be needed to enable applicability to a wider range of practical applications. Generalization over a large part of the periodic table would be desirable, which might require changes to the architecture. Currently, BOA uses separate parameters for each atom type, which could become infeasible for more diverse datasets. A unified basis set for all atom types could be employed in the future to mitigate this problem.

An efficient representation of the density is crucial for basis-set-based models like BOA. Another avenue for future work is therefore to improve the basis set used. BOA currently uses a fixed basis set of uncontracted Gaussian-type basis functions as a base and learns a radial correction factor. To achieve more flexibility, also the exponents of the Gaussian basis functions could be learned, as done in (Fu et al., 2024). In BOA, the overlap and Coulomb matrices used internally could be adapted during training, which could be achieved using differentiable quantum chemistry packages like PySCFAD (Zhang & Chan, 2022).

REFERENCES

Dylan M. Anstine, Roman Zubatyuk, and Olexandr Isayev. AIMNet2: a neural network potential to meet your neutral, charged, organic, and elemental-organic needs. *Chemical Science*, 16(23): 10228–10244, June 2025. ISSN 2041-6539. doi: 10.1039/D4SC08572H. URL https://pubs.rsc.org/en/content/articlelanding/2025/sc/d4sc08572h. Publisher: The Royal Society of Chemistry.

Ilyes Batatia, David P. Kovacs, Gregor Simm, Christoph Ortner, and Gabor Csanyi. MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields. *Advances in Neural Information Processing Systems*, 35:11423–11436, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/4a36c3c51af11ed9f34615b81edb5bbc-Abstract-Conference.html.

Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M. Elena, Dávid P. Kovács, Janosh Riebesell, Xavier R. Advincula, Mark Asta, Matthew Avaylon, William J. Baldwin, Fabian Berger, Noam Bernstein, Arghya Bhowmik, Filippo Bigi, Samuel M. Blau, Vlad Cărare, Michele Ceriotti, Sanggyu Chong, James P. Darby, Sandip De, Flaviano Della Pia, Volker L. Deringer, Rokas Elijošius, Zakariya El-Machachi, Fabio Falcioni, Edvin Fako, Andrea C. Ferrari, John L. A. Gardner, Mikolaj J. Gawkowski, Annalena Genreith-Schriever, Janine George, Rhys E. A. Goodall, Jonas Grandel, Clare P. Grey, Petr Grigorev, Shuang Han, Will Handley, Hendrik H. Heenen, Kersti Hermansson, Christian Holm, Cheuk Hin Ho, Stephan Hofmann, Jad Jaafar, Konstantin S. Jakob, Hyunwook Jung, Venkat Kapil, Aaron D. Kaplan, Nima Karimitari, James R. Kermode, Panagiotis Kourtis, Namu Kroupa, Jolla Kullgren, Matthew C. Kuner, Domantas Kuryla, Guoda Liepuoniute, Chen Lin, Johannes T. Margraf, Ioan-Bogdan Magdău, Angelos Michaelides, J. Harry Moore, Aakash A. Naik, Samuel P. Niblett, Sam Walton Norwood, Niamh O'Neill, Christoph Ortner, Kristin A. Persson, Karsten Reuter, Andrew S. Rosen, Louise A. M. Rosset, Lars L. Schaaf, Christoph Schran, Benjamin X. Shi, Eric Sivonxay, Tamás K. Stenczel, Viktor Svahn, Christopher Sutton, Thomas D. Swinburne, Jules Tilly, Cas van der Oord, Santiago Vargas, Eszter Varga-Umbrich, Tejs Vegge, Martin Vondrák, Yangshuai Wang, William C. Witt, Thomas Wolf, Fabian Zills, and Gábor Csányi. A foundation model for atomistic materials chemistry, September 2025. URL http://arxiv.org/abs/2401.00096. arXiv:2401.00096 [physics].

Mihail Bogojeski, Leslie Vogt-Maranto, Mark E. Tuckerman, Klaus-Robert Müller, and Kieron Burke. Quantum chemical accuracy from density functional approximations via machine learning. *Nature Communications*, 11(1):5223, October 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-19093-1. URL https://www.nature.com/articles/s41467-020-19093-1. Publisher: Nature Publishing Group.

Felix Brockherde, Leslie Vogt, Li Li, Mark E. Tuckerman, Kieron Burke, and Klaus-Robert Müller. Bypassing the Kohn-Sham equations with machine learning. *Nature Communications*, 8(1):872, October 2017. ISSN 2041-1723. doi: 10.1038/s41467-017-00839-3. URL https://www.nature.com/articles/s41467-017-00839-3. Publisher: Nature Publishing Group.

Chaoran Cheng and Jian Peng. Equivariant Neural Operator Learning with Graphon Convolution. November 2023. URL https://openreview.net/forum?id=EjiA3uWpnc&noteId=fqtWXxtsdg.

Bruno Cuevas-Zuviría and Luis F. Pacios. Machine Learning of Analytical Electron Density in Large Molecules Through Message-Passing. *Journal of Chemical Information and Modeling*, 61 (6):2658–2666, June 2021. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c00227. URL https://doi.org/10.1021/acs.jcim.1c00227. Publisher: American Chemical Society.

Beatriz G. del Rio, Brandon Phan, and Rampi Ramprasad. A deep learning framework to emulate density functional theory. *npj Computational Materials*, 9(1):158, August 2023. ISSN 2057-3960. doi: 10.1038/s41524-023-01115-3. URL https://www.nature.com/articles/s41524-023-01115-3. Publisher: Nature Publishing Group.

Jonas Elsborg, Luca Thiede, Alán Aspuru-Guzik, Tejs Vegge, and Arghya Bhowmik. ELECTRA: A Cartesian Network for 3D Charge Density Prediction with Floating Orbitals, May 2025. URL http://arxiv.org/abs/2503.08305. arXiv:2503.08305 [cs].

Pol Febrer, Peter Bjørn Jørgensen, Miguel Pruneda, Alberto García, Pablo Ordejón, and Arghya Bhowmik. Graph2Mat: universal graph to matrix conversion for electron density prediction. *Machine Learning: Science and Technology*, 6(2):025013, April 2025. ISSN 2632-2153. doi: 10.1088/2632-2153/adc871. URL https://dx.doi.org/10.1088/2632-2153/adc871. Publisher: IOP Publishing.

Xiang Fu, Andrew Scott Rosen, Kyle Bystrom, Rui Wang, Albert Musaelian, Boris Kozinsky, Tess Smidt, and Tommi Jaakkola. A Recipe for Charge Density Prediction. November 2024. URL https://openreview.net/forum?id=b7REKaNUTv.

Mario Geiger and Tess Smidt. e3nn: Euclidean Neural Networks, July 2022. URL http://arxiv.org/abs/2207.09453. arXiv:2207.09453 [cs].

Sheng Gong, Tian Xie, Taishan Zhu, Shuo Wang, Eric R. Fadel, Yawei Li, and Jeffrey C. Grossman. Predicting charge density distribution of materials using a local-environment-based graph convolutional network. *Physical Review B*, 100(18):184103, November 2019. doi: 10.1103/PhysRevB.100.184103. URL https://link.aps.org/doi/10.1103/PhysRevB.100.184103. Publisher: American Physical Society.

P. Hohenberg and W. Kohn. Inhomogeneous Electron Gas. *Physical Review*, 136(3B):B864–B871, November 1964. doi: 10.1103/PhysRev.136.B864. URL https://link.aps.org/doi/10.1103/PhysRev.136.B864. Publisher: American Physical Society.

Clemens Isert, Kenneth Atz, José Jiménez-Luna, and Gisbert Schneider. QMugs, quantum mechanical properties of drug-like molecules. *Scientific Data*, 9(1):273, June 2022. ISSN 2052-4463. doi: 10.1038/s41597-022-01390-7. URL https://www.nature.com/articles/s41597-022-01390-7. Publisher: Nature Publishing Group.

Peter Bjørn Jørgensen and Arghya Bhowmik. Equivariant graph neural networks for fast electron density estimation of molecules, liquids, and solids. *npj Computational Materials*, 8(1):183, August 2022. ISSN 2057-3960. doi: 10.1038/s41524-022-00863-y. URL https://www.nature.com/articles/s41524-022-00863-y. Publisher: Nature Publishing Group.

Seongsu Kim and Sungsoo Ahn. Gaussian Plane-Wave Neural Operator for Electron Density Estimation. June 2024. URL https://openreview.net/forum?id=TvoG41N1Y3.

Tsz Wai Ko, Jonas A. Finkler, Stefan Goedecker, and Jörg Behler. Accurate Fourth-Generation Machine Learning Potentials by Electrostatic Embedding. *Journal of Chemical Theory and Computation*, 19(12):3567–3579, June 2023. ISSN 1549-9618. doi: 10.1021/acs.jctc.2c01146. URL https://doi.org/10.1021/acs.jctc.2c01146. Publisher: American Chemical Society.

Teddy Koker, Keegan Quigley, Eric Taw, Kevin Tibbetts, and Lin Li. Higher-order equivariant neural networks for charge density prediction in materials. *npj Computational Materials*, 10(1):161, July 2024. ISSN 2057-3960. doi: 10.1038/s41524-024-01343-1. URL https://www.nature.com/articles/s41524-024-01343-1. Publisher: Nature Publishing Group.

Boris Kozinsky, Albert Musaelian, Anders Johansson, and Simon Batzner. Scaling the Leading Accuracy of Deep Equivariant Models to Biomolecular Simulations of Realistic Size. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '23, pp. 1–12, New York, NY, USA, November 2023. Association for Computing Machinery. ISBN 979-8-4007-0109-2. doi: 10.1145/3581784.3627041. URL https://dl.acm.org/doi/10.1145/3581784.3627041.

Chenghan Li, Or Sharir, Shunyue Yuan, and Garnet Kin-Lic Chan. Image super-resolution inspired electron density prediction. *Nature Communications*, 16(1):4811, May 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-60095-8. URL https://www.nature.com/articles/s41467-025-60095-8. Publisher: Nature Publishing Group.

Yi-Lun Liao, Brandon M. Wood, Abhishek Das, and Tess Smidt. EquiformerV2: Improved Equivariant Transformer for Scaling to Higher-Degree Representations. October 2023. URL https://openreview.net/forum?id=mCOBKZmrzD.

Peter Lippmann, Gerrit Gerhartz, Roman Remme, and Fred A Hamprecht. Beyond Canonicalization: How Tensorial Messages Improve Equivariant Message Passing. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Representation Learning*, volume 2025, pp. 88067–88087, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/db7534a06ace69f4ec95bc89e91d5dbb-Paper-Conference.pdf.

Erpai Luo, Xinran Wei, Lin Huang, Yunyang Li, Han Yang, Zaishuo Xia, Zun Wang, Chang Liu, Bin Shao, and Jia Zhang. Efficient and Scalable Density Functional Theory Hamiltonian Prediction through Adaptive Sparsity. June 2025. URL https://openreview.net/forum?id=K3lykWhXON&noteId=k94UW6cbpQ.

Ilan Mitnikov and Joseph Jacobson. E3STO: Orbital Inspired SE(3)-Equivariant Molecular Representation for Electron Density Prediction, October 2024. URL http://arxiv.org/abs/2410.06119. arXiv:2410.06119 [physics].

Saro Passaro and C. Lawrence Zitnick. Reducing SO(3) Convolutions to SO(2) for Efficient Equivariant GNNs. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 27420–27438. PMLR, July 2023. URL https://proceedings.mlr.press/v202/passaro23a.html. ISSN: 2640-3498.

Chen Qian, Valdas Vitartas, James Kermode, and Reinhard J. Maurer. Equivariant Electronic Hamiltonian Prediction with Many-Body Message Passing, August 2025. URL http://arxiv.org/abs/2508.15108. arXiv:2508.15108 [cond-mat].

Zhuoran Qiao, Matthew Welborn, Animashree Anandkumar, Frederick R. Manby, and Thomas F. Miller, III. OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *The Journal of Chemical Physics*, 153(12):124111, September 2020. ISSN 0021-9606. doi: 10.1063/5.0021955. URL https://doi.org/10.1063/5.0021955.

Zhuoran Qiao, Anders S. Christensen, Matthew Welborn, Frederick R. Manby, Anima Anandkumar, and Thomas F. Miller. Informing geometric deep learning with electronic interactions to accelerate quantum chemistry. *Proceedings of the National Academy of Sciences*, 119(31):e2205221119, 2022. doi: 10.1073/pnas.2205221119. URL https://www.pnas.org/doi/abs/10.1073/pnas.2205221119. _eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.2205221119.

Joshua A Rackers, Lucas Tecot, Mario Geiger, and Tess E Smidt. A recipe for cracking the quantum scaling limit with machine learned electron densities. *Machine Learning: Science and Technology*, 4(1):015027, February 2023. ISSN 2632-2153. doi: 10.1088/2632-2153/acb314. URL https://dx.doi.org/10.1088/2632-2153/acb314. Publisher: IOP Publishing.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):140022, August 2014. ISSN 2052-4463. doi: 10.1038/sdata.2014.22. URL https://www.nature.com/articles/sdata201422. Publisher: Nature Publishing Group.

Roman Remme, Tobias Kaczun, Tim Ebert, Christof A. Gehrig, Dominik Geng, Gerrit Gerhartz, Marc K. Ickler, Manuel V. Klockow, Peter Lippmann, Johannes S. Schmidt, Simon Wagner, Andreas Dreuw, and Fred A. Hamprecht. Stable and Accurate Orbital-Free Density Functional Theory Powered by Machine Learning. *Journal of the American Chemical Society*, 147(32):28851–28859, August 2025. ISSN 0002-7863. doi: 10.1021/jacs.5c06219. URL https://doi.org/10.1021/jacs.5c06219. Publisher: American Chemical Society.

Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, November 2012. ISSN 1549-9596. doi: 10.1021/ci300415d. URL https://doi.org/10.1021/ci300415d. Publisher: American Chemical Society.

Guillem Simeon and Gianni De Fabritiis. TensorNet: Cartesian Tensor Representations for Efficient Learning of Molecular Potentials. November 2023. URL https://openreview.net/forum?id=BEHlPdBZ2e.

Feitong Song and Ji Feng. NeuralSCF: Neural network self-consistent fields for density functional theory, June 2024. URL http://arxiv.org/abs/2406.15873. arXiv:2406.15873 [physics].

Qiming Sun, Xing Zhang, Samragni Banerjee, Peng Bao, Marc Barbry, Nick S. Blunt, Nikolay A. Bogdanov, George H. Booth, Jia Chen, Zhi-Hao Cui, Janus J. Eriksen, Yang Gao, Sheng Guo, Jan Hermann, Matthew R. Hermes, Kevin Koh, Peter Koval, Susi Lehtola, Zhendong Li, Junzi Liu, Narbe Mardirossian, James D. McClain, Mario Motta, Bastien Mussard, Hung Q. Pham, Artem Pulkin, Wirawan Purwanto, Paul J. Robinson, Enrico Ronca, Elvira R. Sayfutyarova, Maximilian Scheurer, Henry F. Schurkus, James E. T. Smith, Chong Sun, Shi-Ning Sun, Shiv Upadhyay, Lucas K. Wagner, Xiao Wang, Alec White, James Daniel Whitfield, Mark J. Williamson, Sebastian Wouters, Jun Yang, Jason M. Yu, Tianyu Zhu, Timothy C. Berkelbach, Sandeep Sharma, Alexander Yu. Sokolov, and Garnet Kin-Lic Chan. Recent developments in the PySCF program package. *The Journal of Chemical Physics*, 153(2):024109, July 2020. ISSN 0021-9606. doi: 10.1063/5.0006074. URL https://doi.org/10.1063/5.0006074.

Ethan M. Sunshine, Muhammed Shuaibi, Zachary W. Ulissi, and John R. Kitchin. Chemical Properties from Graph Neural Network-Predicted Electron Densities. *The Journal of Physical Chemistry C*, 127(48):23459–23466, December 2023. ISSN 1932-7447. doi: 10.1021/acs.jpcc.3c06157. URL https://doi.org/10.1021/acs.jpcc.3c06157. Publisher: American Chemical Society.

Richard Van Noorden. These are the most-cited research papers of all time. *Nature*, 640(8059):591, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Zun Wang, Chang Liu, Nianlong Zou, He Zhang, Xinran Wei, Lin Huang, Lijun Wu, and Bin Shao. Infusing Self-Consistency into Density Functional Theory Hamiltonian Prediction via Deep Equilibrium Models, October 2024. URL http://arxiv.org/abs/2406.03794. arXiv:2406.03794 [cs].

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. *Physical Chemistry Chemical Physics*, 7(18):3297–3305, August 2005. ISSN 1463-9084. doi: 10.1039/B508541A. URL https://pubs.rsc.org/en/content/articlelanding/2005/cp/b508541a. Publisher: The Royal Society of Chemistry.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/488e4104520c6aab692863cc1dba45af-Abstract.html.

Brandon M. Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John R. Kitchin, Daniel S. Levine, Kyle Michel, Anuroop Sriram, Taco Cohen, Abhishek Das, Ammar Rizvi, Sushree Jagriti Sahoo, Zachary W. Ulissi, and C. Lawrence Zitnick. UMA: A Family of Universal Models for Atoms, June 2025. URL http://arxiv.org/abs/2506.23971. arXiv:2506.23971 [cs].

Zilong Yuan, Zechen Tang, Honggeng Tao, Xiaoxun Gong, Zezhou Chen, Yuxiang Wang, He Li, Yang Li, Zhiming Xu, Minghui Sun, Boheng Zhao, Chong Wang, Wenhui Duan, and Yong Xu. Deep learning density functional theory Hamiltonian in real space, July 2024. URL http://arxiv.org/abs/2407.14379. arXiv:2407.14379 [physics].

He Zhang, Chang Liu, Zun Wang, Xinran Wei, Siyuan Liu, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Self-Consistency Training for Density-Functional-Theory Hamiltonian Prediction. June 2024a. URL https://openreview.net/forum?id=Vw4Yar2fmW.

He Zhang, Siyuan Liu, Jiacheng You, Chang Liu, Shuxin Zheng, Ziheng Lu, Tong Wang, Nanning Zheng, and Bin Shao. Overcoming the barrier of orbital-free density functional theory for molecular systems using deep learning. *Nature Computational Science*, 4(3):210–223, March 2024b. ISSN 2662-8457. doi: 10.1038/s43588-024-00605-8. URL https://www.nature.com/articles/s43588-024-00605-8. Publisher: Nature Publishing Group.

Xing Zhang and Garnet Kin-Lic Chan. Differentiable quantum chemistry with PySCF for molecules and materials at the mean-field level and beyond. *The Journal of Chemical Physics*, 157(20): 204801, November 2022. ISSN 0021-9606. doi: 10.1063/5.0118200. URL https://doi.org/10.1063/5.0118200.

A Appendix

A.1 Ablation Studies

A.1.1 Radial Correction and Smooth Absolute Value

An ablation study of the radial correction factor and taking the smooth absolute value of one of the basis functions in each pair in the expansion of the density is performed. The results are shown in Table 3. For each configuration, three training runs are performed and the mean and standard error are reported. Both the radial correction and taking the absolute value significantly improve the performance, with the radial correction having the larger effect. The best performance is achieved when both techniques are used together.

Table 3: **Ablation study of taking the absolute value of one of the basis functions in the pair and applying the radial correction.** The small BOA version is trained on the QM9 VASP dataset. Errors are reported as NMAE [%].

Absolute Value	Radial Correction	NMAE [%]
×	×	0.204 ± 0.002
✓	×	0.167 ± 0.004
×	✓	0.1423 ± 0.0006
✓	✓	$\mathbf{0.1381 \pm 0.0003}$

A.1.2 Choice of Basis Set

The impact of the choice of basis sets is studied on two additional smaller basis sets, `def2-SVP` and `def2-TZVP` (Weigend & Ahlrichs, 2005). All hyperparameters other than the used basis are chosen as in BOA small and the models are trained on the QM9 VASP data. The results are shown in Table 4. Smaller basis sets still perform reasonably well, with the medium size basis `def2-TZVP` still surpassing the previous state of the art. Still, using larger basis sets improves the NMAE significantly.

Table 4: **Basis set ablation.** All settings other than the basis set are kept constant to the small BOA version and the models are trained on the QM9 VASP dataset. Three training runs are performed for each configuration and the mean and standard error are reported.

	NMAE [%]	Number of basis functions
`def2-SVP`	0.194 ± 0.001	103
`def2-TZVP`	0.1504 ± 0.0003	192
`def2-QZVPPD`	0.1381 ± 0.0003	374

A.1.3 Quadratic Expansion

To study the impact of the quadratic expansion we train additional models using the same hyperparameters as BOA small. When evaluating the density on the grid we drop $\hat{g}_a^{(r)}(\mathbf{r})$ and $g_{abo}^{(r)}(\mathbf{r})$ and expand the density linearly as $\rho(\mathbf{r}) = \sum_{a\in\mathcal{N}} \hat{g}_a^{(l)}(\mathbf{r}) + \sum_{(a,b)\in\mathcal{E}_e} \sum_o^{N^o} g_{abo}^{(l)}(\mathbf{r}) = \sum_{a\in\mathcal{N}} \sum_\mu \hat{g}_{a\mu}^{(l)} \omega_\mu^{Z_a}(\mathbf{r} - \mathbf{r}_a) + \sum_{(a,b)\in\mathcal{E}_e} \sum_o^{N^o} \sum_\mu g_{abo\mu}^{(l)} \omega_\mu^{Z_a}(\mathbf{r} - \mathbf{r}_a)$ where the sum over b can be carried out before evaluating the basis functions $\omega_\mu^{Z_a}(\mathbf{r} - \mathbf{r}_a)$ on the grid. Three models were trained on the QM9 VASP data using this linear expansion and evaluated on the test set. The results are shown in Table 5. The quadratic expansion significantly outperforms the linear expansion, showing the importance of the quadratic expansion for accurate density predictions in BOA.

Table 5: **Quadratic expansion ablation.** All settings other than the expansion type are kept constant to the small BOA version and the models are trained on the QM9 VASP dataset. Three training runs are performed for each configuration and the mean and standard error are reported.

	NMAE [%]
Linear Expansion	0.2716 ± 0.0007
Quadratic Expansion	0.1381 ± 0.0003

A.1.4 Radial Cutoff

We conducted an additional experiment with larger cutoffs, specifically a message passing cutoff of $r_{mp} = 8$Å (instead of $r_{mp} = 6$Å) and an edge feature cutoff of $r_e = 4$Å (instead of $r_e = 3$Å) were chosen. Other than the cutoffs the BOA Small settings are used and three models are trained on the QM9 VASP data. Mean and standard error on the test set are reported in Table 6. Using larger cutoffs results in a small but consistent improvement in performance.

Table 6: **Radial cutoff ablation.** All settings other than the cutoffs are kept constant to the small BOA version and the models are trained on the QM9 VASP dataset. Three training runs are performed for each configuration and the mean and standard error are reported.

	NMAE [%]
Standard cutoffs	0.1381 ± 0.0003
Larger cutoffs	0.1343 ± 0.0007

A.2 DFT Initialization and Coulomb Energy Error

To evaluate the quality of the predicted electron densities on metrics closer to practical applications, we calculate the Coulomb energy error and investigate the savings in KS-DFT calculations when using the predicted densities as initialization. Both of these evaluations are performed on the full QM9 PySCF test set of 10000 molecules using a BOA large model. BOA achieves a mean absolute error of 66 meV between the Coulomb energy calculated on the predicted densities and the converged label densities, comparing favorably to the 167 meV reported for the ResNet model (Li et al., 2025).

KS-DFT calculations using our densities as initial guesses and using a standard superposition of atomic densities as initial guesses are performed. The same settings as in the data generation described by Li et al. (2025) are used. With these settings the BOA initialization reduces the mean number of SCF iterations needed from 15.7 to 10.2, a reduction by 35%. This is a result competitive with the 35% reduction reported by Li et al. (2025) on the same data.

A.3 Efficiency

We evaluate the efficiency of ELECTRA [2], SCDP [1], standard BOA, and BOA with `def2-TZVP` basis models. We note that there is no difference in inference efficiency between the small and large settings, since only the number of probe points and the batch size during training are changed, and we therefore do not need to differentiate between them here. All models where evaluated on a 40GB A100 GPU using a block size maximizing the VRAM usage. The time per molecule is evaluated over the whole QM9 Vasp test set and the mean and standard deviation over these molecules are reported in Table 7. While BOA is roughly 2 times slower than SCDP, using the smaller `def2-TZVP` mitigates this effect to some extend and achieves a smaller error than the previous state of the art with efficiency comparable to SCDP. The efficiency/accuracy trade-off is easily steerable in BOA by employing basis sets of different size.

Table 7: **Time Efficiency.** The time per molecule needed during inference is evaluated on the QM9 VASP test set for ELECTRA (Elsborg et al., 2025), SCDP (Fu et al., 2024), standard BOA and BOA with `def2-TZVP` basis. The mean and standard deviation over all molecules in the test set are reported.

Model	Time per molecule [s]
ELECTRA	0.14 ± 0.03
SCDP	0.58 ± 0.16
BOA	1.27 ± 0.27
BOA `def2-TZVP`	0.64 ± 0.15

A.4 Error Distribution

The distribution of the electron density prediction errors on QM9 VASP is analyzed in more detail. Figures 5A and 5B show 2D histograms of the absolute errors of the predicted density for BOA large and SCDP (Fu et al., 2024), binned by the distance to the nearest and second-nearest atom. In each bin the absolute errors are summed. The errors are evaluated on 100 randomly sampled molecules and normalized by the total error of the SCDP model over all evaluated grid points. Additionally, Figure 5C shows the label electron density distribution binned in the same way and normalized by the total density over all evaluated grid points. Both models show the largest errors in the regions with high electron density. In these critical regions with the nearest atom at a distance of less than approximately 1.0Å and the second-nearest atom at a distance of 0.5Å to 1.5Å, the BOA model, however, shows significantly smaller errors than SCDP.

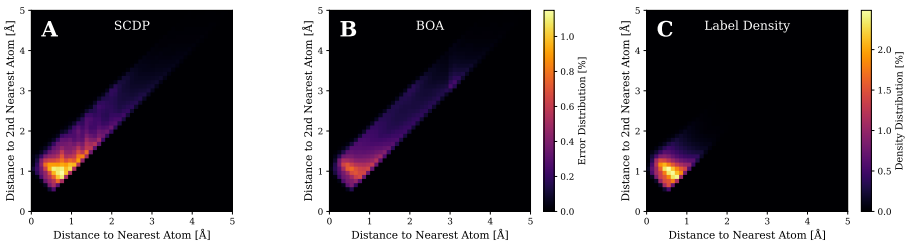

Figure 5: **Error distribution analysis.** The distribution of the absolute errors of the predicted electron density on the QM9 VASP dataset is shown as a function of the distance to the nearest and second-nearest atom. The errors are evaluated on 100 randomly sampled molecules and normalized by the total error of the SCDP model (Fu et al., 2024). **(A)** shows the error distribution of BOA large, **(B)** shows the error distribution of SCDP, and **(C)** shows the label electron density distribution, normalized by the total density over all evaluated grid points. Both models show the largest errors in the regions with high electron density, but BOA shows significantly smaller errors in these critical regions.

A.5 Function Message Passing Derivation

We show that the messages $m_{abm\mu}$ are the coefficients in the basis of node a that represent the feature functions $h_{bn}(\mathbf{r})$ of node b best in the least squares sense. The message passing step therefore solves the optimization problem

$$\min_{m_{abm\mu}} \left\| h_{bm}(\mathbf{r}) - \sum_{\mu} m_{abm\mu} \omega_{\mu}^{Z_a}(\mathbf{r} - \mathbf{r}_a) \right\|^2 \tag{22}$$

$$\Rightarrow \min_{m_{abm\mu}} \left\| \sum_{\mu} h_{bm\mu} \omega_{\mu}^{Z_b}(\mathbf{r} - \mathbf{r}_b) - \sum_{\mu} m_{abm\mu} \omega_{\mu}^{Z_a}(\mathbf{r} - \mathbf{r}_a) \right\|^2 . \tag{23}$$

Taking the derivative with respect to $m_{abm\mu}$ and setting it to zero results in

$$0 = 2\left(\sum_{\mu} h_{bm\mu}\omega_{\mu}^{Z_b}(\mathbf{r}-\mathbf{r}_b) - \sum_{\mu} m_{abm\mu}\omega_{\mu}^{Z_a}(\mathbf{r}-\mathbf{r}_a)\right) \tag{24}$$

$$\Rightarrow \sum_{\mu} h_{bm\mu}\omega_{\mu}^{Z_b}(\mathbf{r}-\mathbf{r}_b) = \sum_{\mu} m_{abm\mu}\omega_{\mu}^{Z_a}(\mathbf{r}-\mathbf{r}_a). \tag{25}$$

Multiplying both sides with $\omega_{\nu}^{Z_a}(\mathbf{r}-\mathbf{r}_a)$ and integrating over $\mathbf{r}$ results in

$$\int d\mathbf{r}\ \sum_{\mu} h_{bm\mu}\omega_{\mu}^{Z_b}(\mathbf{r}-\mathbf{r}_b)\omega_{\nu}^{Z_a}(\mathbf{r}-\mathbf{r}_a) = \int d\mathbf{r}\ \sum_{\mu} m_{abm\mu}\omega_{\mu}^{Z_a}(\mathbf{r}-\mathbf{r}_a)\omega_{\nu}^{Z_a}(\mathbf{r}-\mathbf{r}_a) \tag{26}$$

$$\Rightarrow \sum_{\mu} W_{\nu\mu}^{ab} h_{bm\mu} = \sum_{\mu} W_{\nu\mu}^{aa} m_{abm\mu} \tag{27}$$

$$\Rightarrow m_{abm\mu} = \sum_{\nu} (W^{aa})^{-1}_{\mu\nu} \sum_{\kappa} W_{\nu\kappa}^{ab} h_{bm\kappa}, \tag{28}$$

which is the solution to the optimization problem in Equation 23 and exactly what is implemented in the message passing step.

A.6 Equivariance of BOA

To understand the equivariance of the different BOA layers, it is necessary to understand how the basis functions $\omega_{\mu}^{Z}(\mathbf{r})$ transform under rotations. The angular part of the basis functions is given by spherical harmonics, which are known to transform under irreducible representations of the rotation group SO(3). These irreducible representations are indexed by l with dimension $2l+1$. The basis functions can now be grouped according to their l value, i.e. there are groups of $2l+1$ basis functions that transform under the irreducible representation indexed by l. A basis function $\omega_m^l(\mathbf{r})$ in the group indexed by l transforms under a rotation $\mathbf{R} \in \mathrm{SO}(3)$ as

$$\omega_m^l(\mathbf{R}\mathbf{r}) = \sum_{m'} D_{mm'}^l(\mathbf{R})\omega_{m'}^l(\mathbf{r}), \tag{29}$$

where $D_{mm'}^l(\mathbf{R})$ is the Wigner D-matrix. A set of basis functions $\omega_{\mu}^{Z}(\mathbf{r})$ can therefore be separated into irreducible representations

$$\boldsymbol{\omega}^Z(\mathbf{r}) = \left(\underbrace{\omega_{-l_1}^{l_1}(\mathbf{r}),\ldots,\omega_{l_1}^{l_1}(\mathbf{r})}_{2l_1+1}, \underbrace{\omega_{-l_2}^{l_2}(\mathbf{r}),\ldots,\omega_{l_2}^{l_2}(\mathbf{r})}_{2l_2+1},\ldots\right)^T, \tag{30}$$

resulting in a block diagonal transformation behavior under rotations

$$\omega^Z(\mathbf{R}\mathbf{r}) = \mathbf{D}^Z(\mathbf{R})\boldsymbol{\omega}^Z(\mathbf{r}), \tag{31}$$

where $\mathbf{D}^Z(\mathbf{R})$ is a block diagonal matrix with blocks given by the Wigner D-matrices corresponding to the l values of the basis functions. The Wigner D-matrices are orthogonal matrices, so that $\mathbf{D}^Z(\mathbf{R})$ is also an orthogonal matrix, i.e. $\left(\mathbf{D}^Z(\mathbf{R})\right)^T = \left(\mathbf{D}^Z(\mathbf{R})\right)^{-1}$.

BOA employs the overlap and Coulomb matrices between the basis functions of different nodes, which are defined as

$$W_{\mu\nu}^{ab} = \int d\mathbf{r}\ \omega_{\mu}^{Z_a}(\mathbf{r}-\mathbf{r}_a)\omega_{\nu}^{Z_b}(\mathbf{r}-\mathbf{r}_b), \tag{32}$$

$$C_{\mu\nu}^{ab} = \int d\mathbf{r} \int d\mathbf{r}'\ \frac{\omega_{\mu}^{Z_a}(\mathbf{r}-\mathbf{r}_a)\omega_{\nu}^{Z_b}(\mathbf{r}'-\mathbf{r}_b)}{\|\mathbf{r}-\mathbf{r}'\|}. \tag{33}$$

Considering a rotation $\mathbf{R}$ of the molecule geometry, without changing the orientation of the basis functions, the overlap matrix transforms as

$$\bar{W}^{ab}_{\mu\nu} = \int d\mathbf{r}\ \omega^{Z_a}_{\mu}(\mathbf{r} - \mathbf{R}\mathbf{r}_a)\omega^{Z_b}_{\nu}(\mathbf{r} - \mathbf{R}\mathbf{r}_b) = \int d\mathbf{r}'\ \omega^{Z_a}_{\mu}(\mathbf{R}\mathbf{r}' - \mathbf{R}\mathbf{r}_a)\omega^{Z_b}_{\nu}(\mathbf{R}\mathbf{r}' - \mathbf{R}\mathbf{r}_b) \tag{34}$$

$$= \int d\mathbf{r}'\ \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R})\omega^{Z_a}_{\mu'}(\mathbf{r}' - \mathbf{r}_a) \sum_{\nu'} D^{Z_b}_{\nu\nu'}(\mathbf{R})\omega^{Z_b}_{\nu'}(\mathbf{r}' - \mathbf{r}_b) \tag{35}$$

$$= \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) \sum_{\nu'} D^{Z_b}_{\nu\nu'}(\mathbf{R}) \int d\mathbf{r}'\ \omega^{Z_a}_{\mu'}(\mathbf{r}' - \mathbf{r}_a)\omega^{Z_b}_{\nu'}(\mathbf{r}' - \mathbf{r}_b) \tag{36}$$

$$= \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) \sum_{\nu'} D^{Z_b}_{\nu\nu'}(\mathbf{R}) W^{ab}_{\mu'\nu'}. \tag{37}$$

$$\Rightarrow \bar{\mathbf{W}}^{ab} = \mathbf{D}^{Z_a}(\mathbf{R})\mathbf{W}^{ab}\left(\mathbf{D}^{Z_b}(\mathbf{R})\right)^T, \tag{38}$$

where $\bar{W}$ is the transformed matrix. This notation will be used throughout this section. Completely analogously, the Coulomb matrix transforms

$$\bar{\mathbf{C}}^{ab} = \mathbf{D}^{Z_a}(\mathbf{R})\mathbf{C}^{ab}\left(\mathbf{D}^{Z_b}(\mathbf{R})\right)^T. \tag{39}$$

Similarly, one can consider a function $f(\mathbf{r}) = \sum_\mu f_\mu \omega^Z_\mu(\mathbf{r})$. The rotated function $\bar{f}(\mathbf{r})$ represented in the same basis with rotated molecule geometry is given by

$$\bar{f}(\mathbf{r}) = f(\mathbf{R}^{-1}\mathbf{r}) = \sum_\mu f_\mu \omega^Z_\mu(\mathbf{R}^{-1}\mathbf{r}) = \sum_\mu f_\mu \sum_{\mu'} D^Z_{\mu\mu'}(\mathbf{R}^{-1})\omega^Z_{\mu'}(\mathbf{r}) = \sum_{\mu'} \bar{f}_{\mu'}\omega^Z_{\mu'}(\mathbf{r}), \tag{40}$$

$$\Rightarrow \bar{f}_{\mu'} = \sum_\mu D^Z_{\mu\mu'}(\mathbf{R}^{-1}) f_\mu. \tag{41}$$

$$\Rightarrow \bar{\mathbf{f}} = \left(\mathbf{D}^Z(\mathbf{R}^{-1})\right)^T \mathbf{f} = \mathbf{D}^Z(\mathbf{R})\mathbf{f}. \tag{42}$$

A.6.1 Message Passing

The overlap integrals $o_{abm\mu}$ defined in Equation 9 transform as

$$\bar{o}_{abm\mu} = \sum_\nu \bar{W}^{ab}_{\mu\nu}\bar{h}_{bm\nu} = \sum_\nu \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) \sum_{\nu'} D^{Z_b}_{\nu\nu'}(\mathbf{R}) W^{ab}_{\mu'\nu'} \sum_{\nu''} D^{Z_b}_{\nu\nu''}(\mathbf{R}) h_{bm\nu''} \tag{43}$$

$$= \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) o_{abm\mu'}, \tag{44}$$

leading to the transformation behavior

$$\bar{m}_{abm\mu} = \sum_\nu (\bar{W}^{aa})^{-1}_{\mu\nu}\bar{o}_{abm\nu} = \sum_\nu \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R})(W^{aa})^{-1}_{\mu'\nu} \sum_{\nu'} D^{Z_a}_{\nu\nu'}(\mathbf{R}) o_{abm\nu'} \tag{45}$$

$$= \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) m_{abm\mu'} \tag{46}$$

of the messages $m_{abm\mu}$. The overlap integrals α_{abm} are scalars because

$$\bar{\alpha}_{abmn} = \sum_{\mu\nu} \bar{h}_{am\mu}\bar{W}^{ab}_{\mu\nu}\bar{h}_{bn\nu} \tag{47}$$

$$= \sum_{\mu\nu} \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) h_{am\mu'} \sum_{\mu''} D^{Z_a}_{\mu\mu''}(\mathbf{R}) \sum_{\nu'} D^{Z_b}_{\nu\nu'}(\mathbf{R}) W^{ab}_{\mu''\nu'} \sum_{\nu''} D^{Z_b}_{\nu\nu''}(\mathbf{R}) h_{bn\nu''} \tag{48}$$

$$= \sum_{\mu'} h_{am\mu'} \sum_{\nu'} W^{ab}_{\mu'\nu'} h_{bn\nu'} = \alpha_{abmn}. \tag{49}$$

For scalar features any MLP is equivariant, since scalars are invariant under rotations. The attention weights $\tilde{\alpha}_{abmn}$ are therefore also scalars. The updated node features $\tilde{h}_{am\mu}$ transform as

$$\bar{\tilde{h}}_{am\mu} = \sum_{b \in \mathcal{N}_{\text{mp}}(a)} \sum_{n} \bar{\tilde{\alpha}}_{abmn} \bar{m}_{abn\mu} \tag{50}$$

$$= \sum_{b \in \mathcal{N}_{\text{mp}}(a)} \sum_{n} \tilde{\alpha}_{abmn} \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) m_{abn\mu'} \tag{51}$$

$$= \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) \tilde{h}_{am\mu'}, \tag{52}$$

showing that the message passing layer is equivariant.

A.6.2 Nonlinearity

The nonlinearity is based on calculating scalar quantities, which are then transformed by an MLP, and weighing the original features with these scalars. The scalar features l_{amn} invariant under rotations because

$$\bar{l}_{amn} = \sum_{\mu\nu} \bar{h}_{am\mu} \bar{C}^{aa}_{\mu\nu} \bar{h}_{an\nu} \tag{53}$$

$$= \sum_{\mu\nu} \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) h_{am\mu'} \sum_{\mu''} D^{Z_a}_{\mu\mu''}(\mathbf{R}) \sum_{\nu'} D^{Z_a}_{\nu\nu'}(\mathbf{R}) C^{aa}_{\mu''\nu'} \sum_{\nu''} D^{Z_a}_{\nu\nu''}(\mathbf{R}) h_{an\nu''} \tag{54}$$

$$= \sum_{\mu'} h_{am\mu'} \sum_{\nu'} C^{aa}_{\mu'\nu'} h_{an\nu'} = l_{amn}. \tag{55}$$

Therefore, the weights w_{amn} are also scalars and the updated node features $\tilde{h}_{am\mu}$ transform as

$$\bar{\tilde{h}}_{am\mu} = \sum_{n} \bar{w}_{amn} \bar{h}_{an\mu} = \sum_{n} w_{amn} \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) h_{an\mu'} \tag{56}$$

$$= \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) \tilde{h}_{am\mu'}, \tag{57}$$

showing that the nonlinearity is equivariant.

A.6.3 L2 Normalization

The L2 normalization of the features is done per node and feature channel, i.e. for each a and m. The normalization factor is a scalar because

$$\bar{n}_{am} = \sqrt{\sum_{\mu\nu} \bar{h}_{am\mu} \bar{W}^{aa}_{\mu\nu} \bar{h}_{am\nu}} \tag{58}$$

$$= \sqrt{\sum_{\mu\nu} \sum_{\mu'} D^{Z_a}_{\mu\mu'}(\mathbf{R}) h_{am\mu'} \sum_{\mu''} D^{Z_a}_{\mu\mu''}(\mathbf{R}) \sum_{\nu'} D^{Z_a}_{\nu\nu'}(\mathbf{R}) W^{aa}_{\mu''\nu'} \sum_{\nu''} D^{Z_a}_{\nu\nu''}(\mathbf{R}) h_{am\nu''}} \tag{59}$$

$$= \sqrt{\sum_{\mu'} h_{am\mu'} \sum_{\nu'} W^{aa}_{\mu'\nu'} h_{am\nu'}} = n_{am}. \tag{60}$$

Therefore, the normalized features $\tilde{h}_{am\mu}$ are straightforwardly equivariant.

A.6.4 Edge Update

Exactly the same arguments as for the L2 normalization and the nonlinearity can be used to show that $o^{(n)}_{abmn}$ and $o^{(e)}_{abmn}$ are scalars. Therefore, also the generated weights $s^{(\cdot,\cdot)}_{ab}$ and $\tilde{w}^{(\cdot,\cdot)}_{abmn}$ are scalars, making the update step in Equation 21 equivariant.

A.7 Hyperparameters and Training Details

The model is trained in two stages, the pre-training of the initial guess followed by the full training. During training an exponential moving average (EMA) of the previous weights is used. For testing the epoch that performed best in validation is chosen in all experiments. A mean absolute error loss on the electron density values on the grid is used. When evaluating the density on the grid, edge function values are set to 0 if both of the edge functions have a distance of more than 3 Å from the grid point. The electron density prediction on the grid is multiplied with the standard deviation of the labels calculated over the whole training set. The predicted density is additionally multiplied with 0.1, as this stabilized training in initial experiments.

A.7.1 Model Hyperparameters

The detailed list of hyperparameters of the model is given in Table 8. Settings changed for the pre-training of the initial guess are listed in Table 9. One special layer is the MLP for the edge update, which consists of a two linear layers, with a SiLU activation and a LayerNorm between them. The first layer has $2 \cdot (N^c)^2$ input and output neurons, and the second has $4 \cdot N^c \cdot (N^c + 1)$ output neurons.

Table 8: The hyperparameters on the QM9 dataset using the small settings. Differences on other datasets are in the text.

Hyperparameter	Value
Batch Size	12
Initial Learning Rate	0.001
Learning Rate Schedule	CosineAnnealingLR
T_{max}	500000 steps
Optimizer	Adam
β_1, β_2, ϵ	(0.9, 0.999, 10^{-8})
Weight Decay	0.0
Max Training Steps	500000
Gradient Clipping Value	0.5
EMA Decay	0.995
# BOA blocks	4
# hidden channels N^c	32
# edge functions after partial channel mean	8
MLP activation	SiLU
Initial Guess Channels	1
Absolute Value Scale λ	1000
scale factor	0.1
Probed Points	5000
Message passing cutoff r_{mp}	6 Å
Edge function cutoff r_e	3 Å
Basis Set	def2-QZVPPD

Table 9: The pretraining hyperparameters that differ from the default settings in Table 8.

Hyperparameter	Value
Initial Learning Rate	0.001
Learning Rate Scheduler	CosineAnnealingWarmRestarts
T_0	250
T_{mult}	1
# Steps	1000

The MLP used in the nonlinearity consists of two linear layers with $(N^c)^2$ input and output neurons, separated by first a SiLU activation and then a LayerNorm.

The Gaussian radial embedding employed in the radial correction uses 50 Gaussians evenly spaced between $0a_0$ and $3a_0$ (with a_0 being the Bohr radius). Therefore, 50 values are fed into the following MLP, which consists of first a linear layer with dimension 50×50 then SiLU activation and a linear layer with dimension $50 \times N^{\text{exp}}$. Here N^{exp} is the number of different exponents used in the radial parts of the basis functions, i.e. the number of unique radial functions. For basis functions belonging to the same irreducible representations of SO(3), and therefore having the same radial part, the same correction factor is used.

Differences between BOA sizes Going from the small to the large version of BOA, the number of randomly probed points per molecule is increased from 5000 to 6000. Additionally, the batch size is increased from 12 to 24 which also leads to an increase in "epochs", since the maximum number of steps remains at 500000.

Differences between datasets On the MD datasets, the number of maximum training steps was reduced to 200000.

A.7.2 Basis Set

The number of basis functions per atom type and shell of `def2-QZVPPD` basis set is given in Table 10.

Table 10: Uncontracted (primitive) shells per ℓ in def2-QZVPPD and total spherical functions.

Element	0	1	2	3	4	Total (sph.)
H	7	4	2	1	0	36
C	16	8	4	2	1	83
N	16	8	4	2	1	83
O	16	9	4	2	1	86
F	16	9	4	2	1	86

A.8 Smooth Absolute Value of one of the Edge Functions

Taking the smooth absolute value

$$|x|_s = \begin{cases} \frac{\lambda}{2}x^2, \text{ if } |x| < \frac{1}{\lambda} \\ |x| - \frac{1}{2\lambda}, \text{ otherwise} \end{cases} \tag{61}$$

of $g^{(r)}_{abo}(\mathbf{r})$ resulted in slightly improved performance compared to keeping the function unmodified, see Appendix Section A.1.1. The factor λ steers the scale of the smooth absolute value function and is here chosen as $\lambda = 1000$. The density is then given by $\hat{\rho}(\mathbf{r}) = \sum_{(a,b)\in\mathcal{E}_e} \sum_o^{N^o} g^{(l)}_{abo}(\mathbf{r}) \left| g^{(r)}_{abo}(\mathbf{r}) \right|_s + \sum_{a\in\mathcal{N}} \hat{g}^{(l)}_a(\mathbf{r}) \left| \hat{g}^{(r)}_a(\mathbf{r}) \right|_s$.

A.9 LLM usage

Large Language Models (LLMs) were used to assist with writing (formulation and wording), coding (AI autocomplete), and gathering resources in the preparation of this article.