ELECTRA: A SYMMETRY-BREAKING CARTESIAN NETWORK FOR CHARGE DENSITY PREDICTION WITH FLOATING ORBITALS

Jonas Elsborg[*], **Tejs Vegge & Arghya Bhowmik**[†]
Department of Energy Conversion and Storage
Technical University of Denmark
CAPeX Pioneer Center
Kgs. Lyngby, Denmark
{jels, teve, arbh}@dtu.dk

Luca Thiede[*] **& Alán Aspuru-Guzik**
Department of Computer Science
University of Toronto
Vector Institute
Toronto, Ontario, Canada
{luca.thiede, aspuru}@utoronto.ca

ABSTRACT

We present the Electronic Tensor Reconstruction Algorithm (ELECTRA) - an equivariant model for predicting electronic charge densities using "floating" orbitals. Floating orbitals are a long-standing idea in the quantum chemistry community that promises more compact and accurate representations by placing orbitals freely in space, as opposed to centering all orbitals at the position of atoms. Finding ideal placements of these orbitals requires extensive domain knowledge though, which thus far has prevented widespread adoption. We solve this in a data-driven manner by training a Cartesian tensor network to predict orbital positions along with orbital coefficients. This is made possible through a symmetry-breaking mechanism that is used to learn position displacements with lower symmetry than the input molecule while preserving the rotation equivariance of the charge density itself. Inspired by recent successes of Gaussian Splatting in representing densities in space, we are using Gaussians as our orbitals and predict their weights and covariance matrices. Our method achieves a state-of-the-art balance between computational efficiency and predictive accuracy on established benchmarks.

1 INTRODUCTION

High-accuracy simulations for the design of materials and molecules at the atomic scale are most often done with density functional theory (DFT) based simulations (Kohn & Sham, 1965), as DFT provides a good balance between cost and accuracy for quantum mechanical simulations of matter (Marzari et al., 2021). However, the O(n^3) scaling of DFT still limits the system sizes and time scales that can be simulated. Linear scaling ML surrogates such as neural network potentials trained with a large number of DFT simulations can alleviate this problem by learning a direct mapping between atomic structure and corresponding energy, forces, and other properties with accuracy similar to those from DFT simulations (Friederich et al., 2021). This approach, although first envisioned three decades ago (Blank et al., 1995), has become successful and popular in recent years based on multiple seminal developments (Behler, 2021; Deringer et al., 2021; Unke et al., 2021).

An alternative data-efficient ML-accelerated physics simulation approach can be taken where the underlying variable of the DFT simulations, the electron density, is predicted directly from atomic structures, without self-consistent field (SCF) iterations (Bogojeski et al., 2020). Following the Hohenberg-Kohn theorem (Hohenberg & Kohn, 1964), all ground-state properties can be calculated once this ground-state electron density is known (Grisafi et al., 2022). In recent years, researchers have addressed this task in multiple ways, differentiated by the representation of density data, the molecular representation, and the ML architecture itself (Grisafi et al., 2018; Chandrasekaran et al., 2019; Jørgensen & Bhowmik, 2022; Rackers et al., 2023). The target electron density is commonly predicted on real space grids (Chandrasekaran et al., 2019; Li et al., 2024) or as an expansion of

[*]Equal contribution.
[†]Corresponding author: arbh@dtu.dk

atom-centered basis functions (Grisafi et al., 2018; Bogojeski et al., 2020; Cuevas-Zuviría & Pacios, 2021; Rackers et al., 2023) which usually take the form

$$\Phi_{l,m}(\boldsymbol{r}) = R_l(r)Y_{lm}(\theta,\phi), \quad l = 0, ..., L \tag{1}$$

$R_l(r)$ represents the radial dependence relative to a center, while $Y_{lm}(\theta,\phi)$ captures the angular dependencies. Larger quantum numbers l correspond to higher-frequency components. The accuracy of the represented density is dependent on the "quality" of the basis set or the grid density. Two key attributes define the quality of a basis set: The number of basis functions per angular quantum number l and the maximum angular momentum quantum number L included in the expansion.

Different systems and properties necessitate varying levels of basis set complexity. There is no universal basis set that provides both, high accuracy and optimal computational efficiency for all types of systems. Instead, the selection of an appropriate basis set depends on the specific requirements of the system under investigation and requires deep domain expertise.

For example, accurate descriptions of systems involving highly polarizable molecules or those with diffuse electron distributions far away from atom centers may require augmented basis sets like aug-cc-pVTZ (Kendall et al., 1992), which include functions with high angular momentum and diffuse components that have long-tailed radial functions designed for modeling long-range dependencies. On the other hand, for smaller systems or those dominated by core-electron interactions, these basis sets lead to unnecessarily large compute costs. In particular, basis functions with higher angular quantum numbers L incur significant costs.

A more compact representation of densities can be achieved by putting extra basis functions at locations of presumed interest, particularly in areas far away from atoms, with rapidly varying densities. These basis functions are called "floating" orbitals, and their utility is well-established in electronic structure theory (Tao & Pan, 1992; Tao, 1993; Tasi & Császár, 2007). They date back to the floating spherical Gaussian orbital (FSGO) model (Frost, 1968). When chosen wisely, floating orbitals can lead to significant improvements in calculation speed and accuracy (Lorincz & Nagy, 2024) by reducing the need for diffuse and high angular momentum basis functions.

Well-placed floating orbitals can represent densities more efficiently, using fewer basis functions and lower maximal angular quantum numbers L. ELECTRA is the first model to predict floating orbital positions without human input.

However, the optimal locations of floating orbitals are often hard to determine (Zheng et al., 2021), and picking good locations therefore requires deep electronic structure domain expertise (Lorincz & Nagy, 2024). Our core contribution is a data-driven solution to this problem. We are training a model that, given a molecular graph, accurately reconstructs ground truth charge densities by predicting the 3D position of floating orbitals as well as the coefficients and parameters that define them. Since charge densities are rotation invariant, we use a rotation equivariant neural network as the backbone of our model. However, a naive implementation of equivariant neural networks is destined to fail, since good placements of floating orbitals can have lower symmetry than the input molecular graph, as we will discuss in later sections. We address this problem by developing a symmetry-breaking mechanism that retains rotational equivariance. We call the resulting model the Electronic Tensor Reconstruction Algorithm (ELECTRA). We test ELECTRA on the widely used QM9 charge density dataset and achieve results that are competitive with state-of-the-art while being consistently faster.

2 Related work

2.1 Charge density prediction

Prior work on machine learning prediction of charge density generally falls into two main approaches, inspired by earlier non-ML methods. Orbital-based methods are rooted in linear combinations of atom-centered orbitals (LCAO), which take the form

$$\rho(\boldsymbol{r}) = \sum_{i}^{N} \sum_{j}^{N_b^i} \sum_{m=-l_{i,j}}^{l_{i,j}} c_{i,j,m} \Phi_{\alpha_{i,j},l_{i,j},m,\boldsymbol{r}_i}(\boldsymbol{r}), \tag{2}$$

where the first sum runs over all atoms, and the other two sum index into all basis functions per atom. Φ usually takes a form as in 1. In the ML community, methods based on this construction typically predict coefficients $c_{i,j,m}$ extracted from ground truth DFT calculations (Fabrizio et al., 2019; Qiao et al., 2022; Rackers et al., 2023; Cheng & Peng, 2024; del Rio et al., 2023; Febrer et al., 2024) as well as refined radial functions $R_l(r)$ (Fu et al., 2024). This is computationally efficient at inference, and orbital-decomposed density representations can offer enhanced accuracy in describing both total and orbital energies by utilizing flexible, orbital-specific potentials that align closely with many-body spectral properties (Ferretti et al., 2014). However, the fixed choice of basis set often limits representation power unless a large expensive basis set is used, particularly for complex inter-atomic electronic features. By placing additional orbitals on bond midpoints, (Fu et al., 2024) achieved higher expressivity, albeit at the cost of higher compute costs, and the additional requirement of determining bonds. The latter point sounds trivial, but bonds are not always well defined which can make this difficult. The second method is inspired by viewing the charge density as a numerical grid (Cerjan, 2013) which must be probed at each point to construct the density. By inserting a graph node that can receive messages from the atomic graph representation (Gong et al., 2019; Jørgensen & Bhowmik, 2022; Koker et al., 2024; Pope & Jacobs, 2024; Li et al., 2024) in each grid point, these models directly predict scalar charge values at grid points, offering high expressiveness and accuracy. Even for small molecules, charge density data contains hundreds of thousands of points, and thus probe-based models are generally more computationally intensive than orbital-based models.

2.2 Equivariance and Cartesian tensors

Many objects in physics transform predictably under symmetry transformations. This property is called equivariance. Formally, a function $\boldsymbol{f} : \boldsymbol{X} \to \boldsymbol{Y}$ is equivariant with respect to a group $\mathcal{G}$ which elements $\boldsymbol{g} \in \mathcal{G}$ act on $\boldsymbol{X}$ and $\boldsymbol{Y}$, if

$$\mathbf{f}(\boldsymbol{g}\boldsymbol{x}) = \boldsymbol{g}\mathbf{f}(\boldsymbol{x}) \tag{3}$$

For example, we are often interested in the case where $\mathcal{G}$ is the group of translations, rotations, and reflections, see (Thomas et al., 2018; Geiger & Smidt, 2022; Simeon & De Fabritiis, 2024) for more details. Constructing a machine learning model with the equivariance property (equation 3) provides a strong inductive bias that usually leads to increased data efficiency (Brehmer et al., 2024). The electron density is rotation invariant, the special case of equivariance for scalar-valued functions:

$$\rho(\boldsymbol{R}\boldsymbol{r}) = \rho(\boldsymbol{r}) \tag{4}$$

Cartesian tensors provide a systematic way to handle rotation equivariance. An lth-rank Cartesian tensor $\mathbf{T}$ is an lth-rank tensor that transforms under rotation as

$$\boldsymbol{T}_{i_1 i_2 \cdots i_l} \xrightarrow{\boldsymbol{R}} = (\boldsymbol{R}_{i_1 j_1})(\boldsymbol{R}_{i_2 j_2}) \cdots (\boldsymbol{R}_{i_l j_l})\, \boldsymbol{T}_{j_1 j_2 \cdots j_l} \tag{5}$$

where $\mathbf{R}$ is an orthogonal matrix in Cartesian coordinates. Equivariant graph networks can be built to leverage operations on Cartesian tensors such as linear combinations, tensor contractions, and partial derivatives (Simeon & De Fabritiis, 2024; Wang et al., 2024) that ensure that the network's outputs are equivariant.

2.3 HotPP tensor message-passing neural network

One example of equivariant networks that operate on Cartesian tensors is the High-order Tensor Passing Potential (HotPP) (Wang et al., 2024). HotPP's node features and messages are arbitrary order Cartesian tensors and the operations are constrained such that the outputs remain Cartesian Tensors. This allows predictions on higher order physical quantities like dipole moments (rank 1 tensors, i.e., vectors) and polarizability tensors (rank 2 tensors, i.e., 3x3 matrices), and similarly allows for more complex atomic environments to be distinguished. In HotPP, an atomistic system is represented as a graph $G = (V, E)$, where V is the set of atoms (nodes) and E is the set of edges (defined up to a cutoff radius) in a molecule. Each atom A is characterized by a feature vector v_A, and each edge ($e_{A_1,A_2} \in E$) between atoms A_1 and A_2 is associated with an edge vector v_{A_2,A_2} and a scalar distance d_{A_2,A_2}.

3 METHODS

3.1 THE ELECTRA MODEL

For ELECTRA we make a simplified ansatz compared to the LCAO ansatz. Inspired by the recent success of Gaussian splatting (Kerbl et al., 2023), we represent the charge density as a 3D Gaussian mixture model:

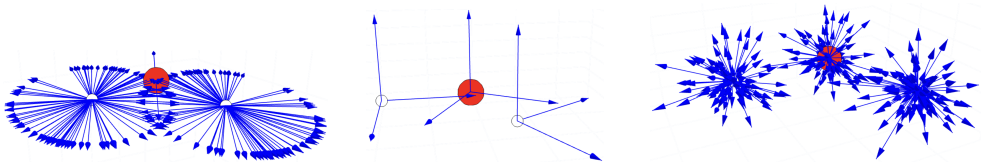
$$\rho(\mathbf{r}) = \sum_{A \in M} \sum_{j=0}^{N_A} w_{A,j} \mathcal{N}(\boldsymbol{r} | \boldsymbol{\mu}_{A,j}, \boldsymbol{\Sigma}_{A,j}) \tag{6}$$

where $A \in M$ represents the atoms in the molecule, and N_A is the number of Gaussians for each atom, which can depend on the atom type. The weights $w_{A,j}$ are signed, which improves expressivity. This can, e.g., be used to construct shell-like structures by inserting a negative density at the center of a larger sphere. In the following sections, we elaborate on how all Gaussians are constructed in a per-atom way, using the model output specific to each atom. Gaussians are equivalent to traditional Cartesian orbital functions with angular quantum number $l = 2$, an extra nonlinearity, and simplified radial dependency; see appendix A for why. This is the reason we are saying that ELECTRA uses $l = 2$ orbitals. We will see that, if made "floating", these simplified orbitals are enough to achieve strong performance, in contrast to atom-centered basis functions, which require a high maximum angular quantum number L for good performance. In principle, we could also use more conventional basis functions and make them "floating". To enforce rotational invariance (equation 4) of the predicted electron density (equation 6), we need the weights $w_{A,j}$ to be rotation invariant, while the means $\boldsymbol{\mu}_{A,j}$ and covariance matrices $\boldsymbol{\Sigma}_{A,j}$ need to be rotation equivariant. In particular, the Gaussian means and covariance matrices need to transform like a *Cartesian* tensors, see appendix B for details.

Equivariant backbone neural network. To enforce the constraints on $w_{A,j}$, $\boldsymbol{\mu}_{A,j}$ and $\boldsymbol{\Sigma}_{A,j}$, we use a modified version of the HotPP (Wang et al., 2024) equivariant message-passing network to represent atomistic systems in ELECTRA. We initialize the scalar features as well as the first three rank-1 features in each atom using a tailored embedding function. Both initialization methods are important to the final model and are detailed in the paragraphs below. The graph is updated through a series of HotPP's update layers. We then use the resulting features to predict the parameterization $w_{A,i}$, $\boldsymbol{\mu}_{A,i}$ and $\boldsymbol{\Sigma}_{A,i}$ for the Gaussians in equation 6 using a readout head layer. Other important changes to the default HotPP implementation are detailed in the paragraphs below, and ELECTRA otherwise follows mostly the reference implementation.

Atomic embeddings and variable basis set size. In quantum chemistry, different atoms require differently sized basis sets, since the complexity of the electronic structure generally depends on the atomic number (Weigend & Ahlrichs, 2005) and the number of valence electrons. Inspired by this, ELECTRA predicts a variable number of Gaussians depending on the number of valence electrons. This is achieved by assigning each output channel of each atom in HotPP to one Gaussian. Denoting M_e as the number of Gaussians per valence electron, we can use the first $n_e \cdot M_e$ channels of each atom to represent the Gaussians, where n_e is the number of valence electrons for that atom. For this to work, a channel width of $N_c = 8 \cdot M_e$ is sufficient in HotPP. Each atom A then uses only its first $M_e \cdot n_{e,A}$ channels. For example, oxygen ($n_e = 6$ from the 2s and 2p shells) utilizes $6M_e$ output channels, while hydrogen ($n_e = 1$ from the 1s shell) uses only M_e output channels. In typical graph neural network interatomic potentials, atom-wise representations are initially encoded through categorical atom-type embeddings (Reiser et al., 2022). In contrast, to encode a maximum amount of information about the electronic and nuclear properties of each atom into the scalar features, ELECTRA uses a different embedding function, which we call *Aufbau Embedding*. Each atom's electronic and nuclear properties are encoded as $\mathbf{f}_{\mathrm{Auf}} = [P, N, V, E_1, \ldots, E_n, F_1, \ldots, F_{\mathrm{n}}]$, where P, N and V are the numbers of protons, neutrons and valence electrons while E_i and F_i denote orbital occupancies and free-electron counts, respectively. Thus, the encoding $\mathbf{f}_{\mathrm{Auf}}$ reflects the Aufbau (Bohr, 1913) and Pauli exclusion principles (Pauli, 1925). A multi-layer perceptron (MLP) maps $\mathbf{f}_{\mathrm{Auf}}$ to the embedding space, $\tilde{\mathbf{f}_{\mathrm{Auf}}} = \mathrm{MLP}_{\mathrm{Emb}}(f_{Auf})$, which is used as the initial scalar features.

Symmetry-breaking Since ELECTRA's orbitals are not as expressive as standard spherical harmonics-based orbitals, the model needs to have maximum freedom in placing the orbitals in space to achieve comparable expressivity. However, the symmetries of equivariant networks prohibit



(a) (b) (c)

Figure 1: **(a)** Model output without symmetry-breaking: Equivariant neural networks constrain their output to have the same symmetry as the input. If the input molecule is highly symmetric, this leads to highly constrained Gaussian positions. **(b)** To solve this issue, ELECTRA initializes each atom's $l = 1$ vector features with the eigenvectors of the moment of inertia tensor as calculated in that specific atom, which breaks the input symmetry but retains rotational equivariance. **(c)** Model output after first linear layer with symmetry-breaking: The model can learn its own set of symmetry-breaking vectors, allowing output to not be constrained by the symmetry of the input molecule.

their outputs from having a lower symmetry than their inputs (Smidt et al., 2021; Xie & Smidt, 2024). This would prohibit any individual Gaussian from escaping, e.g., the symmetry plane of a planar 2D molecule (Figure 1a), thereby severely limiting the possible richness of the final density. Since many ground-state geometries are highly symmetric, this poses a big issue for equivariant networks. Previous work has investigated both indirect and direct ways of breaking symmetries. Indirect methods typically relax the equivariance constraints (van der Ouderaa et al., 2022; Kaba & Ravanbakhsh, 2023; Huang et al., 2024), which is undesirable for electron densities since these are exactly equivariant (Rackers et al., 2023). Other methods break symmetries by constructing symmetry-breaking inputs (Liu et al., 2019; Locatello et al., 2020; Xie & Smidt, 2024) or by learning order-breaking parameters during training (Smidt et al., 2021).

To construct an expressive method for placing floating orbitals, the model must allow for output vectors that belong to a lower symmetry group than the input structures. Thus, a symmetry-breaking mechanism is needed.

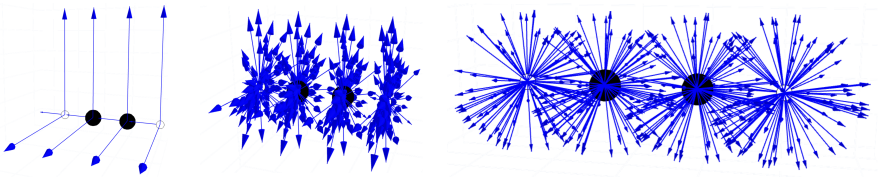
Our approach to symmetry-breaking with ELECTRA broadly falls into the category of symmetry-breaking inputs. For our construction, we are first calculating a local moment of inertia (MOI) tensor for each atom:

$$I_{ij}^{(atom)} = \sum_{k=1}^{N} m_k \left(\|\mathbf{r}_k\|^2 \delta_{ij} - x_i^{(k)} x_j^{(k)} \right) \tag{7}$$

Where $k = 1..N$ runs over all atoms inside a local atomic neighborhood defined up to a cutoff radius from the current atom, and the vectors $\mathbf{r_k} = \left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}\right)$ are calculated relative to the current atom. The three eigenvectors of equation 7 are also rotation equivariant, and thus, we can use them to initialize the first three $l = 1$ vector features of ELECTRA's GNN, from largest to smallest eigenvalue (Fig. 1c), while maintaining rotational equivariance. They are linearly independent and define a local coordinate system on which the model can learn its own set of symmetry-breaking objects using a linear embedding layer, the output of which is depicted in Fig.1c. However, the eigenvectors of a matrix are only defined up to a sign flip. One can resolve this issue for example by averaging all possible sign combinations, similar to Duval et al. (2023). However, in our case, this averages out all meaningful anisotropies. Instead, we opt for canonicalizing the eigenvectors. Canonicalization of eigenvectors is a research topic that is studied independently (Ma et al., 2024). In this work, we use a sign convention that maximizes the dot product of each eigenvector with the position vector of the center of mass (COM) of the molecule. Mathematically for an eigenvector $\boldsymbol{v}$, we switch the sign according to:

$$\boldsymbol{v}_{\text{canon}} = \begin{cases} \boldsymbol{v}, & \text{if } \boldsymbol{v} \cdot \boldsymbol{r}_{\text{COM}} \geq 0, \\ -\boldsymbol{v}, & \text{if } \boldsymbol{v} \cdot \boldsymbol{r}_{\text{COM}} < 0. \end{cases} \tag{8}$$

There are transformations where this canonicalization will lead to a sign flip, and the model has to learn to compensate for them. However, we find, empirically, that this does not hinder strong performance.



(a) (b) (c)

Figure 2: (**a**) The symmetry-breaking objects the C_2H_2 molecule is initialized with. (**b**) The output of a HotPP model with symmetry-breaking but without debiasing layers: The message passing induces a directional bias that concentrates vectors along certain directions, (**c**) The output of our model with debiasing layers: The output vectors don't show any visible bias.

Debiasing layers. Even though our symmetry-breaking mechanism allows ELECTRA theoretically to break any symmetry, we observe that the message-passing mechanism of HotPP induces a directional bias of the $l = 1$ features, particularly in highly symmetric molecules. For example, in Fig. 2b, we plot the output vectors from a randomly initialized HotPP model with the C_2H_2 molecule and symmetry-breaking objects as input. We see, that the vectors tend to be perpendicular to the bond axis of the molecule, which is problematic if we were to use these vectors as Gaussian positions because the ground truth density has a lot of density around the bond axis. Empirically the model was not able to overcome this bias and place the Gaussians efficiently in space, which led to low performance. To address this issue, we are designing a layer that learns to dynamically remove directional biases in the vector features. Our debiasing layer, which we place after every message passing layer, first calculates the covariance matrix of all the $l = 1$ node features associated with each atom:

$$\mathbf{C}_A = \frac{1}{D} \sum_{j=1}^{D} \boldsymbol{v}_{A,j} \boldsymbol{v}_{A,j}^T \tag{9}$$

where $\boldsymbol{v}_{A,j}$ are the $l = 1$ features for atom A, and D is the channel dimension. We denote $\mathbf{u_{1,A}}$ as the eigenvector of $\boldsymbol{C_A}$ with the largest eigenvalue λ_1. If there is a directional bias in the features of an atom, $\mathbf{u_1}$ points in the direction of the largest variation. The stronger the directional bias, the larger the magnitude of $\mathbf{u_1}$. We calculate the projection of each $l = 1$ feature onto this principal axis $\mathbf{u}_1$:

$$\mathbf{v}_{A,j}^{\parallel} = (\mathbf{v}_{A,j} \cdot \mathbf{u}_{1,A})\, \mathbf{u}_{1,A} \tag{10}$$

Note that the sign ambiguity of $\mathbf{u}_{1,A}$ is not important in this case, as $\mathbf{u}_{1,A}$ appears twice in the projection. By subtracting $\mathbf{v}_{A,j}^{\parallel}$ from $\mathbf{v}_{A,j}$ we can reduce the directional bias. To let the model decide how much to subtract, we predict a weight $\mathbf{w}_{A,j}$ using a small neural network, conditioned on rotationally invariant features:

$$\begin{aligned} \mathbf{inp}_{A,j} &= \left[\hat{\mathbf{v}}_{A,j}^{\parallel} \cdot \hat{\mathbf{v}}_{A,j}, \hat{\mathbf{v}}_{A,j}^{\parallel} \cdot \hat{\mathbf{r}}_{A,j}, \hat{\mathbf{r}}_{A,j} \cdot \hat{\mathbf{v}}_{A,j},\ ||\mathbf{v}_{A,j}||, ||\mathbf{v}_{A,j}^{\parallel}||\right] \\ \mathbf{w}_{A,j} &= \mathrm{MLP}_{\mathrm{D}}\left(\mathbf{inp}_{A,j}\right) \end{aligned} \tag{11}$$

where $\hat{\mathbf{v}}_{A,j}^{\parallel}$ are the normalized parallel components for each atom's jth vector feature $\mathbf{v}_{A,j}$, $\hat{\mathbf{v}}_i$ are the normalized vector features, $\hat{\mathbf{r}}_{A,j}$ are the normalized direction vectors to the center of mass, while $\|\mathbf{v}_A\|$ and $||\mathbf{v}_A^{\parallel}||$ are the lengths of the vector features and the components parallel to the principal axis for atom A, respectively. The output $\mathbf{w}_{A,j} \in [0, 1]$ is a number that determines how much of the principal direction to remove in each vector, such that the debiased vector is updated as:

$$\mathbf{v}_{A,j} \leftarrow \frac{\mathbf{v}_{A,j} - \mathbf{w}_{A,j} \cdot \hat{\mathbf{v}}_{A,j}^{\parallel}}{||\mathbf{v}_A - \mathbf{w}_{A,j} \cdot \hat{\mathbf{v}}_{A,j}^{\parallel}||}, \tag{12}$$

We normalize the vectors in equation 12 such that the $l = 1$ features mainly handle directionality rather than scale, which is instead handled in the readout layer by $l = 0$ predictions. Feeding geometry-rich information to $\mathrm{MLP}_{\mathrm{D}}$ and learning $\mathbf{w}_{A,j}$ thus provides a way to determine and remove bias in the $l = 1$ features.

3.2 DENSITY CONSTRUCTION

After several message-passing layers, we have a set of features for each atom that we feed into three readout heads. Each readout head produces a set of $l = 0$ ($\mathbf{s}$), $l = 1$ ($\mathbf{v}$) and $l = 2$ ($\mathbf{M}$) features, $(\mathbf{s_1}, \mathbf{v}_1, \mathbf{M}_1)_{A,j}$, $(\mathbf{s_2}, \mathbf{v}_2, \mathbf{M}_2)_{A,j}$ and $(\mathbf{s_3}, \mathbf{v}_3, \mathbf{M}_3)_{A,j}$, where A indexes the atoms in the molecule, and j the channel. Depending on the atom type, the channel index is $j \in [0, ..., N_e(A) \times M_e]$, where N_e is the number of valence electrons of that atom, and M_e is the number of Gaussians per valence electrons. We will use these predictions for the parameterization of the Gaussians in our ansatz (equation 6) as described in the following sections. Intuitively, each of the three heads is specialized on a different distance scale away from the atoms.

Scalar factors. As a first step, we use the scalar features together with the Aufbau embedding to construct an input for three different MLPs: $\mathbf{s_{inp}}_{A,j} = \left[\tilde{\mathbf{f}}_{\mathbf{Auf}}, \mathbf{s_1}_{A,j}, \mathbf{s_2}_{A,j}, \mathbf{s_3}_{A,j}\right]$. The MLPs then predict the Gaussian mixture weights $w_{A,j} \in \mathbb{R}$ together with two other sets of scalars, $\mathbf{s_p}_{A,j} \in \mathbb{R}^3$ and $\mathbf{s_m}_{A,j} \in \mathbb{R}^3$ to use in the mean and covariance predictions.

Gaussian positions. ELECTRA places Gaussians (i.e., predicts the mean positions $\boldsymbol{\mu}_{A,j}$) equivariantly by using the $l = 1$ outputs $\mathbf{v}_{1,A,j}, \mathbf{v_2}_{A,j}, \mathbf{v_3}_{A,j}$ and the position scaling factors $\mathbf{s_p}$ of the framework as displacement vectors to the atomic positions:

$$\boldsymbol{\mu}_{A,j} = \mathbf{r}_A + \exp\left(\mathbf{s_{p_1}}_{A,j}\right)\mathbf{v_1}_{A,j} + \mathbf{s^2_{p_2}}_{A,j}\mathbf{v_2}_{A,j} + \mathbf{s_{p_3}}_{A,j}\mathbf{v_3}_{A,j} \tag{13}$$

Therefore, each Gaussian is associated with a position equal to the position $\mathbf{r}_A$ of the atom A it originates from, plus three displacement vectors multiplied by scaling factors. We transform the scaling factors in different ways (exponential, square, and identity) to provide different scales of position displacement, thereby aiming to capture different levels of detail of the output density with each readout head.

Density prediction using Gaussian mixture models. To construct the Gaussian's covariance matrices $\boldsymbol{\Sigma}_{A,j}$ we calculate a weighted sum of the $l = 2$ outputs $\mathbf{M_1}, \mathbf{M_2}, \mathbf{M_3}$. To ensure symmetry and positive semi-definiteness of the covariance matrix, we symmetrize the matrices by constructing the Gram matrices of $\mathbf{M}$, i.e. using the transformation $\mathbf{M} \to \mathbf{M}\mathbf{M}^\top$. For notational simplicity, we omit the A, j subscript for all matrices and scalars in the equations below:

$$\begin{aligned}
&\mathbf{s_{G_1}}, \mathbf{s_{G_2}}, \mathbf{s_{G_3}} = \mathrm{softmax}\left([\mathbf{s_{m_1}}, \mathbf{s_{m_2}}, \mathbf{s_{m_3}}]\right)] \\
&\boldsymbol{\Sigma} = \mathbf{s_{G_1}}\frac{\mathbf{M_1}\mathbf{M_1}^\top}{\|\mathbf{M_1}\|_F} + \mathbf{s_{G_2}}\frac{\mathbf{M_2}\mathbf{M_2}^\top}{\|\mathbf{M_2}\|_F} + \mathbf{s_{G_3}}\frac{\mathbf{M_3}\mathbf{M_3}^\top}{\|\mathbf{M_3}\|_F}.
\end{aligned} \tag{14}$$

The Gram matrices are normalized by the Frobenius norm of the original output matrices to preserve the scale for the symmetrized matrices.

Model evaluation. Due to the large number of grid points, evaluation of all the orbitals on them is the biggest bottleneck of density prediciton models (not the neural network). To make the orbital evaluation efficient, we employ a cutoff radius: Given a grid point $\boldsymbol{r}$, we evaluate our model (equation 6), by considering all Gaussians with mean inside the cutoff radius, since we know that Gaussians far away will contribute exponentially less to the density. This yields significant performance gains. Using a cutoff radius of 3.0 Angstrom, we have to evaluate only about 15%, and for 2.5 Angstrom only about 9.0% of our Gaussians on average on our dataset (see next section). Compare that to the LCAO framework, which has to evaluate all basis functions on atoms within a cutoff radius, even if the basis functions don't contribute to the density at the given point.

Normalization. Prior work has shown that density prediction models whose output does not integrate to the number of electrons can lead to errors in downstream property predictions (Briling et al., 2021). Thus, as a final step, the densities predicted by ELECTRA are normalized to the number of valence electrons in the system:

$$\rho_{\text{pred}}(\mathbf{r}) = \rho(\mathbf{r}) \times \frac{n_{\text{elec}}}{\int_{\mathbb{R}^3} \rho(\mathbf{r})\,\mathrm{d}V} \implies \int_{\mathbb{R}^3} |\rho_{\text{pred}}(\boldsymbol{r})|\mathrm{d}V = n_{\text{elec}} \tag{15}$$

Table 1: Performance of various models with numbers adapted from Fu et al. (2024). *Numbers from original publications. Note that InfGCN and GPWNO use a different split than the standard 10,000 molecule test split. †Timings are performed on an A100-80GB (Fu et al., 2024)

Model	NMAE [%] ↓	t [s] ↓
i-DeepDFT*	0.357	-
e-DeepDFT*	0.284	-
InfGCN, GTO*	3.720	-
GPWNO*	0.730	-
ChargE3Net	0.196	15.18†
InfGCN	0.869	0.833†

Table 2: Performance of various models tested in this work. For the model that is most comparable to ELECTRA, SCDP, we report the fastest model (L=3) as well as the slowest (+BO L=6), and the model with comparable accuracy (SCDP L=6). BO stands for bond orbitals.

This work	NMAE [%] ↓	t [s] ↓
SCDP (L=3)	0.432	0.395
ELECTRA (2.5 Å)	**0.371**	**0.357**
SCDP (L=6)	0.350	0.471
ELECTRA (3.0 Å)	**0.345**	**0.441**
SCDP + BO (L=6)	**0.178**	1.022

where $\mathrm{d}V$ represents the differential volume element on the grid. Since n_{elec} is simply the number of valence electrons, this number is already provided as an input to standard DFT codes or can easily be obtained via summation over the valence electrons of each atom in the system.

Objective function We train ELECTRA on a loss function $\mathcal{L}$ based on the normalized mean absolute error:

$$\mathcal{L} = \mathrm{NMAE}\left(\rho_{\mathrm{pred}}, \rho_{\mathrm{ref}}\right) = \frac{\int_{\mathbb{R}^3} |\rho_{\mathrm{ref}}(\boldsymbol{r}) - \rho_{\mathrm{pred}}(\boldsymbol{r})| \mathrm{d}V}{\int_{\mathbb{R}^3} |\rho_{\mathrm{ref}}(\boldsymbol{r})| \mathrm{d}V} \tag{16}$$

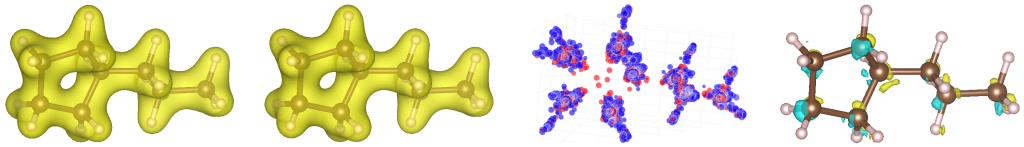
It is not necessary to compute the denominator in equation 16 during training since the reference grid must integrate to the number of valence electrons, i.e., $\int_{\mathbb{R}^3} |\rho_{\mathrm{ref}}(\boldsymbol{r})| \mathrm{d}V = n_{elec}$, and thus the denominator integral can be replaced with n_{elec} during training.

4 Experiments

Dataset and implementation. We train ELECTRA on reference densities from the QM9 density files which were generated in VASP (Kresse & Hafner, 1993) using the PBE (Perdew et al., 1996) functional and the Projector-Augmented Wave (PAW) (Blöchl, 1994) method (Jørgensen & Bhowmik, 2022). We use the full split consisting of 123,835 training molecules, 50 validation molecules, and 10,000 testing molecules. We train ELECTRA for two weeks using ten NVIDIA RTX 3090-24GB GPUs. During validation and testing, we use a single 3090 GPU and process each molecule's grid points sequentially in chunks, similar to other implementations (Fu et al., 2024). A full list of our hyperparameters is given in Table 3. As an example of how ELECTRA distributes individual Gaussians around a molecule, we provide Figure 3, which also shows the resulting density prediction and the ground truth density for C_8H_{16}.

Results. In Tables 1 and 2, we report the mean accuracy and inference time of two ELECTRA models, differing only by the inference cutoff radius. We compare ELECTRA to previous and concurrent charge density prediction models, using results from the original papers as well as results from the model testing carried out in Fu et al. (2024). The SCDP models are the current state-of-the-art, and thus to ensure a fair comparison we tested them on our own hardware. Finally, in Table 4 in the appendix we report ablated versions of ELECTRA; one without floating orbitals (w/o FO), where all Gaussians are atom centered, and one with floating orbitals but without symmetry-breaking (w/o SB). The ablated models demonstrate that both mechanisms are indispensable to achieving good results. ELECTRA's performance on accuracy is similar to the state-of-the-art performance of SCDP; the 2.5 Å version of ELECTRA is both faster and more accurate than $\mathrm{SCDP}(L = 3)$, and similarly for the 3.0 Å version of ELECTRA vs $\mathrm{SCDP}(L = 6)$. On inferior hardware (3090-24GB vs A100-80GB), ELECTRA is faster than both ChargE3Net and InfGCN. For ChargE3Net (Koker et al., 2024), this difference is more than an order of magnitude (inference time of 0.441 seconds vs. 15 seconds per molecule), although ELECTRA is not as accurate (NMAE[%] of 0.345 vs. 0.198).

For InfGCN (Cheng & Peng, 2024), the inference time for ELECTRA is roughly half (0.441 seconds versus 0.83 seconds per molecule), while the accuracy is also significantly higher (NMAE[%] of 0.345 vs. 0.833). ELECTRA is comparable to the DeepDFT models and SCDP models without



(a) (b) (c) (d)

Figure 3: **(a)** Predicted density for C_8H_{16} using the best ELECTRA model (NMAE [%] $= 0.21\%$). **(b)** Ground truth density for C_8H_{16}. **(c)** Gaussian placements for C_8H_{16} - red Gaussians have $w_{A,j} > 0$ in equation 6, while blue Gaussians have $w_{A,j} < 0$. **(d)** Isosurfaces for errors for C_8H_{16} - predicted density is larger than the ground truth in yellow areas and smaller in blue areas.

bond-centered orbitals in terms of accuracy, while having lower inference times. Thus, ELECTRA achieves state-of-the-art in a time-vs-accuracy tradeoff. While the SCDP + BO model leads to better accuracy than ELECTRA, this model is significantly slower. The likely reasons for this are, that ELECTRA uses a lower maximum angular momentum number ($L = 2$ vs $L = 6$) and can be more aggressive with the orbital inference cutoff (3 Å vs 5 Å).

ELECTRA uses floating orbitals to achieve density prediction accuracies that rival state-of-the-art while being faster on inference due to efficient orbital inference cutoffs and lower-order tensor representations.

5 Discussion

Future directions. Fu et al. (2024) show that adding bond-centered orbitals increases expressivity. However, for this method to work, bonds must be identified in real-time during training and inference. This may fail for complex systems with non-classical bonding and delocalized interactions, such as partially formed or broken bonds, variable bonding radii, weak interactions like π-backbonding, and coordination variability. A particular issue for crystal lattices would arise for, e.g., color centers, where vacancies are occupied by unpaired atomic electrons (Seitz, 1946). Since the vacancy itself does not contain a bond or atom, centered orbitals would likely fail. Similarly, in electrides, the electrons effectively function as anions, requiring non-centered positions (Dye, 2003). In all the above cases, using freely placable orbitals originating from atoms is still viable. Our model would theoretically be able to learn the non-local behavior of the density through equation 13 or variations thereof. Additionally, floating orbitals scale with the number of atoms rather than the number of bonds, making them computationally more efficient for large and complex systems where bond identification is challenging or ambiguous. This is only amplified by the drastic reduction in the number of orbitals that must be evaluated in each grid point, which we have demonstrated in this work. Naturally, it may be possible to construct floating orbitals using spherical coordinates, which was also suggested in Fu et al. (2024). We thus believe that ELECTRA is complementary to their work, and we believe that it would allow the benefits of floating orbitals to be combined with the flexibility of the spherical harmonics-based SCDP models. A model with both atom-centered and floating orbitals would likely represent the most efficient use of computational resources, since even in ELECTRA, many of the orbitals are placed around atomic centers, as shown in Fig. 3. Beyond charge density prediction, there are other applications within geometric point cloud data that might benefit from the general idea of modeling complex geometric point cloud or grid data as a sum or mixture of simple distributions originating from a graph representation, e.g. aerodynamical flows or protein surfaces (Helwig et al., 2024; Zhao et al., 2024).

Scaling and chemical universality. ELECTRA uses Cartesian coordinates since the $l = 1$ vector outputs of Cartesian tensor networks can be straightforwardly interpreted as displacement vectors, while the Gram matrices constructed from the $l = 2$ output can be used as covariance matrices. Since any continuous probability density function can be approximated arbitrarily well as a weighted sum

of Gaussian components (Plataniotis & Hatzinakos, 2017), our approach is guaranteed to be sufficient in the limit of an infinite number of components. However, practical implementations must use finite mixtures, which necessitates trade-offs between model complexity and accuracy. For ELECTRA, the main tunable parameter is the number of Gaussians M_e dedicated to each valence electron. The results in Tables 1 and 2 provide reasonable evidence that such a trade-off can be achieved while keeping inference time favorable.

Grid inference, subsampling, and loss function. ELECTRA was trained on the standard QM9 benchmarking dataset with a batch size of 1, doing inference on the full grid at every step. In future work, we aim to explore grid subsampling methods to improve training speeds and scalability. Furthermore, the loss could be designed to enhance downstream property prediction accuracy - an established bottleneck in ML-driven charge density prediction (Briling et al., 2021). This could, for example, be achieved through multi-task learning on properties such as energy.

REFERENCES

Jorg Behler. Four generations of high-dimensional neural network potentials. *Chemical Reviews*, 121 (16):10037–10072, 2021.

Thomas B Blank, Steven D Brown, August W Calhoun, and Douglas J Doren. Neural network models of potential energy surfaces. *The Journal of chemical physics*, 103(10):4129–4137, 1995.

Peter E Blöchl. Projector augmented-wave method. *Physical review B*, 50(24):17953, 1994.

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, 2020.

Niels Bohr. I. on the constitution of atoms and molecules. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 26(151):1–25, 1913. doi: 10.1080/14786441308634955. URL https://doi.org/10.1080/14786441308634955.

Johann Brehmer, Sönke Behrends, Pim de Haan, and Taco Cohen. Does equivariance matter at scale? *arXiv preprint arXiv:2410.23179*, 2024.

Ksenia R Briling, Alberto Fabrizio, and Clemence Corminboeuf. Impact of quantum-chemical metrics on the machine learning prediction of electron density. *The Journal of Chemical Physics*, 155(2), 2021.

Charles Cerjan. *Numerical grid methods and their application to Schrödinger's equation*, volume 412. Springer Science & Business Media, 2013.

Anand Chandrasekaran, Deepak Kamal, Rohit Batra, Chiho Kim, Lihua Chen, and Rampi Ramprasad. Solving the electronic structure problem with machine learning. *npj Computational Materials*, 5 (1):22, 2019.

Chaoran Cheng and Jian Peng. Equivariant neural operator learning with graphon convolution. *Advances in Neural Information Processing Systems*, 36, 2024.

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, 2021.

Beatriz G del Rio, Brandon Phan, and Rampi Ramprasad. A deep learning framework to emulate density functional theory. *npj Computational Materials*, 9(1):158, 2023.

Volker L Deringer, Albert P Bartók, Noam Bernstein, David M Wilkins, Michele Ceriotti, and Gábor Csányi. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16): 10073–10141, 2021.

Alexandre Agm Duval, Victor Schmidt, Alex Hernández-Garcıa, Santiago Miret, Fragkiskos D Malliaros, Yoshua Bengio, and David Rolnick. Faenet: Frame averaging equivariant gnn for materials modeling. In *International Conference on Machine Learning*, pp. 9013–9033. PMLR, 2023.

James L Dye. Electrons as anions. *Science*, 301(5633):607–608, 2003.

Alberto Fabrizio, Andrea Grisafi, Benjamin Meyer, Michele Ceriotti, and Clemence Corminboeuf. Electron density learning of non-covalent systems. *Chemical science*, 10(41):9424–9432, 2019.

Pol Febrer, Peter Jørgensen, Miguel Pruneda, Alberto Garcia, Pablo Ordejon, and Arghya Bhowmik. Graph2mat: Universal graph to matrix conversion for electron density prediction. *Chemrxiv*, 2024.

Andrea Ferretti, Ismaila Dabo, Matteo Cococcioni, and Nicola Marzari. Bridging density-functional and many-body perturbation theory: Orbital-density dependence in electronic-structure functionals. *Physical Review B*, 89(19):195134, 2014.

Pascal Friederich, Florian Häse, Jonny Proppe, and Alán Aspuru-Guzik. Machine-learned potentials for next-generation matter simulations. *Nature Materials*, 20(6):750–761, 2021.

Arthur A Frost. A floating spherical gaussian orbital model of molecular structure. iii. first-row atom hydrides. *The Journal of Physical Chemistry*, 72(4):1289–1293, 1968.

Xiang Fu, Andrew Rosen, Kyle Bystrom, Rui Wang, Albert Musaelian, Boris Kozinsky, Tess Smidt, and Tommi Jaakkola. A recipe for charge density prediction. *arXiv preprint arXiv:2405.19276*, 2024.

Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

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, 2019.

Andrea Grisafi, Alberto Fabrizio, Benjamin Meyer, David M Wilkins, Clemence Corminboeuf, and Michele Ceriotti. Transferable machine-learning model of the electron density. *ACS central science*, 5(1):57–64, 2018.

Andrea Grisafi, Alan M Lewis, Mariana Rossi, and Michele Ceriotti. Electronic-structure properties from atom-centered predictions of the electron density. *Journal of Chemical Theory and Computation*, 19(14):4451–4460, 2022.

Jacob Helwig, Xuan Zhang, Haiyang Yu, and Shuiwang Ji. A geometry-aware message passing neural network for modeling aerodynamics over airfoils. *arXiv preprint arXiv:2412.09399*, 2024.

Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.

Ningyuan Huang, Ron Levie, and Soledad Villar. Approximately equivariant graph networks. *Advances in Neural Information Processing Systems*, 36, 2024.

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, 2022.

Sékou-Oumar Kaba and Siamak Ravanbakhsh. Symmetry breaking and equivariant neural networks. *arXiv preprint arXiv:2312.09016*, 2023.

Rick A Kendall, Thom H Dunning Jr, and Robert J Harrison. Electron affinities of the first-row atoms revisited. systematic basis sets and wave functions. *The Journal of chemical physics*, 96(9): 6796–6806, 1992.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.

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, 2024.

Georg Kresse and Jürgen Hafner. Ab initio molecular dynamics for liquid metals. *Physical review B*, 47(1):558, 1993.

Chenghan Li, Or Sharir, Shunyue Yuan, and Garnet K Chan. Image super-resolution inspired electron density prediction. *arXiv preprint arXiv:2402.12335*, 2024.

Jenny Liu, Aviral Kumar, Jimmy Ba, Jamie Kiros, and Kevin Swersky. Graph normalizing flows. *Advances in Neural Information Processing Systems*, 32, 2019.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

Balázs D Lorincz and Péter R Nagy. Advancing non-atom-centered basis methods for more accurate interaction energies: benchmarks and large-scale applications. *The Journal of Physical Chemistry A*, 128(47):10282–10298, 2024.

George Ma, Yifei Wang, and Yisen Wang. Laplacian canonization: A minimalist approach to sign and basis invariant spectral embedding. *Advances in Neural Information Processing Systems*, 36, 2024.

Nicola Marzari, Andrea Ferretti, and Chris Wolverton. Electronic-structure methods for materials design. *Nature materials*, 20(6):736–749, 2021.

Wolfgang Pauli. Über den zusammenhang des abschlusses der elektronengruppen im atom mit der komplexstruktur der spektren. *Zeitschrift für Physik*, 31(1):765–783, 1925.

John P Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Physical review letters*, 77(18):3865, 1996.

Kostantinos N Plataniotis and Dimitris Hatzinakos. Gaussian mixtures and their applications to signal processing. *Advanced signal processing handbook*, pp. 89–124, 2017.

Phillip Pope and David Jacobs. Towards combinatorial generalization for catalysts: a kohn-sham charge-density approach. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhuoran Qiao, Anders S Christensen, Matthew Welborn, Frederick R Manby, Anima Anandkumar, and Thomas F Miller III. Informing geometric deep learning with electronic interactions to accelerate quantum chemistry. *Proceedings of the National Academy of Sciences*, 119(31):e2205221119, 2022.

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, 2023.

Patrick Reiser, Marlen Neubert, André Eberhard, Luca Torresi, Chen Zhou, Chen Shao, Houssam Metni, Clint van Hoesel, Henrik Schopmans, Timo Sommer, et al. Graph neural networks for materials science and chemistry. *Communications Materials*, 3(1):93, 2022.

Frederick Seitz. Color centers in alkali halide crystals. *Reviews of modern physics*, 18(3):384, 1946.

Guillem Simeon and Gianni De Fabritiis. Tensornet: Cartesian tensor representations for efficient learning of molecular potentials. *Advances in Neural Information Processing Systems*, 36, 2024.

Tess E Smidt, Mario Geiger, and Benjamin Kurt Miller. Finding symmetry breaking order parameters with euclidean neural networks. *Physical Review Research*, 3(1):L012002, 2021.

Fu-Ming Tao. The use of midbond functions for ab initio calculations of the asymmetric potentials of he–ne and he–ar. *The Journal of chemical physics*, 98(4):3049–3059, 1993.

Fu-Ming Tao and Yuh-Kang Pan. Mo/ller–plesset perturbation investigation of the he2 potential and the role of midbond basis functions. *The Journal of chemical physics*, 97(7):4989–4995, 1992.

Gyula Tasi and Attila G Császár. Hartree–fock-limit energies and structures with a few dozen distributed gaussians. *Chemical physics letters*, 438(1-3):139–143, 2007.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Oliver T Unke, Stefan Chmiela, Huziel E Sauceda, Michael Gastegger, Igor Poltavsky, Kristof T Schütt, Alexandre Tkatchenko, and Klaus-Robert Muüller. Machine learning force fields. *Chemical Reviews*, 121(16):10142–10186, 2021.

Tycho van der Ouderaa, David W Romero, and Mark van der Wilk. Relaxing equivariance constraints with non-stationary continuous filters. *Advances in Neural Information Processing Systems*, 35: 33818–33830, 2022.

Junjie Wang, Yong Wang, Haoting Zhang, Ziyang Yang, Zhixin Liang, Jiuyang Shi, Hui-Tian Wang, Dingyu Xing, and Jian Sun. E (n)-equivariant cartesian tensor message passing interatomic potential. *Nature Communications*, 15(1):7607, 2024.

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, 2005.

YuQing Xie and Tess Smidt. Equivariant symmetry breaking sets, 2024. URL `https://arxiv.org/abs/2402.02681`.

Yanpeng Zhao, Song He, Yuting Xing, Mengfan Li, Yang Cao, Xuanze Wang, Dongsheng Zhao, and Xiaochen Bo. A point cloud graph neural network for protein–ligand binding site prediction. *International Journal of Molecular Sciences*, 25(17):9280, 2024.

Qiang Zheng, Tianli Feng, Jordan A Hachtel, Ryo Ishikawa, Yongqiang Cheng, Luke Daemen, Jie Xing, Juan Carlos Idrobo, Jiaqiang Yan, Naoya Shibata, et al. Direct visualization of anionic electrons in an electride reveals inhomogeneities. *Science Advances*, 7(15):eabe6819, 2021.

A GAUSSIANS AS CARTESIAN BASIS FUNCTIONS

The most often found form of basis functions in quantum chemistry is

$$\Phi_{\alpha,l,m}(\boldsymbol{r}) = R_{\alpha,l}(|\boldsymbol{r}|)Y_{lm}(\theta,\phi) \tag{17}$$

where l is the angular quantum number. To build our densities we contract the basis functions with a set of coefficients $C_{l,m}$, such that the contribution of all basis functions centered at the same spot can be written in Einstein notation as $C_{\alpha,l,m}\Phi_{\alpha,l,m}$

However, we can also equivalently use cartesian basis functions. With $\boldsymbol{r} = (x, y, z)$ we define:

$$\Phi_{\alpha,n_x,n_y,n_z}(x,y,z) = N(\alpha,n_x,n_y,n_z)R_\alpha(|\boldsymbol{r}|)x^{n_x}y^{n_y}z^{n_z} \tag{18}$$

where $N(\alpha, n_x, n_y, n_z)$ is a normalization factor. The angular quantum number in the case of cartesian basis functions is defined as $l = n_x + n_y + n_z$. We can collect all the terms above belonging to the same l in one tensor:

$$\boldsymbol{\Phi}_{\alpha,l}(x,y,z) = \boldsymbol{N}(\alpha,l)\cdot R_\alpha(|\boldsymbol{r}|)\underbrace{\boldsymbol{r}\otimes\boldsymbol{r}\otimes...\otimes\boldsymbol{r}}_{l\text{ times}} \tag{19}$$

In particular, for $l = 2$ we get

$$\boldsymbol{\Phi}_{l=2}(x,y,z) = \boldsymbol{N}(\alpha,2)\cdot R_\alpha(|\boldsymbol{r}|)\boldsymbol{r}\boldsymbol{r}^\top \tag{20}$$

When we contract this with a coefficient matrix $\boldsymbol{C}_{i,j}$, to calculate the contributions of the basis functions to our density, we get

$$C_{i,j}(\boldsymbol{\Phi}_{l=2})_{i,j} = R_\alpha(|\boldsymbol{r}|)\boldsymbol{r}^\top\underbrace{(\boldsymbol{N}(\alpha,l)\cdot\boldsymbol{C})}_{:=-2\boldsymbol{\Sigma}^{-1}}\boldsymbol{r} \tag{21}$$

If we set the radial term $R_\alpha(|\boldsymbol{r}|) = 1$, wrap the remaining term in an exponential function and choose $\boldsymbol{C}$ such that $\boldsymbol{\Sigma}$ is positive definite, we get, up to a normalization constant, a Gaussian:

$$\exp\left(-\frac{1}{2}\boldsymbol{r}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{r}\right) \propto \mathcal{N}(r|0,\boldsymbol{\Sigma}) \tag{22}$$

B EQUIVARIANCE OF GAUSSIANS LEADS TO INVARIANT DENSITY

The electron density is rotation invariant. In the main text, we claimed that our ansatz (equation 6)

$$\rho(\mathbf{r}) = \sum_{A\in M}\sum_{j=0}^{N_A} w_{A,j}\mathcal{N}(\boldsymbol{r}|\boldsymbol{\mu}_{A,j},\boldsymbol{\Sigma}_{A,j}) \tag{23}$$

is rotation invariant, if the weights $w_{A,j}$ are rotation invariant, and the position $\boldsymbol{\mu}_{A,j}$ and and covariance matrices $\boldsymbol{\Sigma}_{A,j}$ transform as $l = 1$ and $l = 2$ cartesian tensors. It is clear, that the entire ansatz is invariant if each Gaussian is individually invariant. So we need to show, that

$$\mathcal{N}(\boldsymbol{r}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{R}\boldsymbol{r}|\boldsymbol{R}\boldsymbol{\mu},\boldsymbol{R}\boldsymbol{\Sigma}\boldsymbol{R}^\top) \tag{24}$$

for a rotation matrix $\boldsymbol{R}$. For simplicity, we omit the normalization constant of the Gaussian, since it is rotation invariant. Then we can write

$$\mathcal{N}(\boldsymbol{R}\boldsymbol{r}|\boldsymbol{R}\boldsymbol{\mu},\boldsymbol{R}\boldsymbol{\Sigma}\boldsymbol{R}^\top) = \exp\left(-\frac{1}{2}(\boldsymbol{R}(\boldsymbol{r}-\boldsymbol{\mu}))^\top(\boldsymbol{R}\boldsymbol{\Sigma}\boldsymbol{R}^\top)^{-1}\boldsymbol{R}(\boldsymbol{r}-\boldsymbol{\mu})\right) \tag{25}$$

$$= \exp\left(-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{\mu})^\top\boldsymbol{R}^\top\boldsymbol{R}\boldsymbol{\Sigma}^{-1}\boldsymbol{R}^\top\boldsymbol{R}(\boldsymbol{r}-\boldsymbol{\mu})\right) \tag{26}$$

$$= \exp\left(-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{r}-\boldsymbol{\mu})\right) \tag{27}$$

$$= \mathcal{N}(\boldsymbol{r}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \tag{28}$$

where we have used that $\boldsymbol{R}^{-1} = \boldsymbol{R}^\top$. This shows our claim.

C HYPERPARAMETER TABLE

Summarized in Table 3 below are standard hyperparameters used for all experiments.

Table 3: Hyperparameters for the main ELECTRA model.

Hyperparameter	Value
Gaussian inference cutoff	3.0 Å
Graph radius cutoff	8.0 Å
HotPP # body layers	3
Gaussians per electron (M_e)	50
Graph network channel width	450
GNN L_{max} - Body layers	3
GNN L_{max} - Head layers and inference	2
Body order (N_{max})	3
Precision	Float32
Optimizer	Adam (Default parameters) (Kingma, 2014)
Weight decay	0
Initial learning rate $LR_{initial}$	1×10^{-5}
Learning rate scheduler	Linear ($LR = LR_{initial} \times \gamma^{Epoch}$)
Learning rate gamma (γ)	0.6
Testing batch size	1
Testing no. of grid points per forward pass	100,000

D ABLATION STUDIES

We ablate an ELECTRA model with 3.0 Å inference cutoff radius in two ways. In the first instance by excluding the floating orbitals completely (**w/o FO** in the table), placing all Gaussians in the atom centers, and in the second one by excluding the symmetry-breaking mechanism (**w/o SB**) while still allowing displacement vectors.

Table 4: Ablation studies of ELECTRA.

Model	NMAE [%] ↓	t_{inf} [s] ↓
ELECTRA (3.0 Å)	**0.345**	**0.441**
ELECTRA w/o FO	5.951	0.477
ELECTRA w/o SB	6.836	0.455