
ELECTRA: A Cartesian Network for 3D Charge Density Prediction with Floating Orbitals

Jonas Elsborg^{1,2*} Luca Thiede^{3,4*}

Alán Aspuru-Guzik^{3,4,5} Tejs Vegge^{1,2} Arghya Bhowmik^{1,2} [†]

¹Technical University of Denmark

²CAPeX Pioneer Center for Accelerating P2X Materials Discovery

³University of Toronto

⁴Vector Institute for Artificial Intelligence

⁵Canadian Institute for Advanced Research (CIFAR)

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 concept 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 the ideal placement 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 the 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 Gaussian orbitals and predicting their weights and covariance matrices. Our method achieves a state-of-the-art balance between computational efficiency and predictive accuracy on established benchmarks. Furthermore, ELECTRA is able to lower the compute time required to arrive at converged DFT solutions - initializing calculations using our predicted densities yields an average 50.72 % reduction in self-consistent field (SCF) iterations on unseen molecules.

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., 2021c).

*These authors contributed equally.

[†]Corresponding authors: jels@dtu.dk, arbh@dtu.dk

An alternative data-efficient ML-accelerated physics simulation approach can be taken where the underlying fundamental variable of the DFT simulations, the electron density, is predicted directly from atomic structures, without self-consistent field (SCF) iterations. 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; Bogojeski et al., 2020). 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; Jørgensen & Bhowmik, 2020; Li et al., 2024) or as an expansion of atom-centered basis functions (Grisafi et al., 2018; Unke et al., 2021a; Cuevas-Zuviría & Pachos, 2021; Rackers et al., 2023) which usually take the form

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

$R_l(r)$ 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. 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 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 (Jørgensen & Bhowmik, 2022) 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 (CD) 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

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

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 higher computational costs, and the additional requirement of determining bonds. The latter point sounds trivial, but bonds are not always well defined making 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.

Once we have the density, two broad strategies emerge:

Use $\rho(\mathbf{r})$ as an initial guess: Plane-wave based KS-DFT can take $\rho(\mathbf{r})$ given on a grid and cheaply map it to plane-wave coefficients using fast Fourier transform (FFT), which allows to resume SCF. This approach was demonstrated by Jørgensen & Bhowmik (2020, 2022). The same idea applies to orbital-free DFT (OF-DFT) Weizsäcker (1935); Mi et al. (2023) which natively operates on grids and allows for much faster, albeit less accurate, SCF and energy predictions.

Use $\rho(\mathbf{r})$ directly: A potential alternative route is to train on KS-DFT densities and only evaluate the energy with OF-DFT without extra SCF, an approach that is standard for benchmarking OF-DFT functionals and that was shown to give better predictions than self-consistent OF-DFT densities Constantin & Ruzsinszky (2009); Iyengar et al. (2001); Perdew et al. (1988). Other properties that can be evaluated directly without extra SCF are Bader’s Theory of Atoms in Molecules, which shows how a topological analysis of $\rho(\mathbf{r})$ can be used to define bonds rigorously. This allows us to identify covalent, ionic, and noncovalent interactions Bader & Nguyen-Dang (1981); Bader & Essén (1984); Boto et al. (2017) and construct charge partitioning schemes with widespread use in quantum chemistry. Densities also provide qualitative insights into electrophilic and nucleophilic regions Bader et al. (1984), as well as weaker interactions such as hydrogen bonding Koch & Popelier (1995), and allow us to calculate multipole moments and field tensors. Recently developed usage of $\rho(\mathbf{r})$ includes density-based generative 3D drug design models Ragoza et al. (2022); Wang et al. (2022) and density-based descriptors to study redox processes in electrochemical systems (Laubach et al., 2009; de Blasio et al., 2023; Shang et al., 2022) and electrocatalysts Zheng et al. (2014); Koch et al. (2021). Densities themselves are also often used to build cheap surrogates for observables like ionic diffusion (Kahle et al., 2018).

2.2 Equivariance and Cartesian tensors

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

$$f(g_{\mathbf{X}}x) = g_{\mathbf{Y}}f(x) \quad (3)$$

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 (3) provides a strong inductive bias that usually leads to increased data efficiency (Brehmer et al., 2024).

If \mathbf{Y} is a function space, $\rho \in \mathbf{Y}$, $g_{\mathbf{Y}}$ acts on ρ via the left-regular representation, defined as (Brandstetter et al., 2021):

$$[g_{\mathbf{Y}}\rho](\mathbf{r}) = \rho(g^{-1}\mathbf{r}) \quad (4)$$

In this work, we are learning a model f that maps from molecular geometry $\text{mol} = \{\mathbf{r}_k\}_{k=0}^N$ to the electron density

$$[f(\text{mol})](\mathbf{r}) = \rho(\mathbf{r}|\text{mol}) \quad (5)$$

This mapping is equivariant under rotation \mathbf{R}

$$f(\mathbf{R} \text{mol}) = \rho(\mathbf{r}|\mathbf{R} \text{mol}) = \rho(\mathbf{R}^{-1}\mathbf{r}|\text{mol}) \quad (6)$$

Equivalently, the electron density is rotation invariant under joint rotation of both electron and molecular geometry:

$$\rho(\mathbf{R}\mathbf{r}|\mathbf{R} \text{mol}) = \rho(\mathbf{R}^{-1}\mathbf{R}\mathbf{r}|\text{mol}) = \rho(\mathbf{r}|\text{mol}) \quad (7)$$

For notational convenience, we will drop the conditioning on mol in our notation from now on and only write $\rho(\mathbf{r})$.

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

$$\mathbf{T}_{i_1 i_2 \dots i_l} \xrightarrow{\mathbf{R}} = (\mathbf{R}_{i_1 j_1}) (\mathbf{R}_{i_2 j_2}) \dots (\mathbf{R}_{i_l j_l}) \mathbf{T}_{j_1 j_2 \dots j_l} \quad (8)$$

where \mathbf{R} is an orthogonal matrix in Cartesian coordinates. Equivariant graph networks can be built to leverage operations on Cartesian tensors (Simeon & De Fabritiis, 2024; Wang et al., 2024) 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. One example of equivariant networks that operates 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_1, A_2} and a scalar distance d_{A_1, A_2} .

3 Methods

3.1 The ELECTRA model

For ELECTRA we make a simplified ansatz compared to the LCAO ansatz. Inspired by the idea in Gaussian Splatting (Kerbl et al., 2023) of approximating a complex 3D field with many inexpensive Gaussians, we here use Gaussians as an efficient density basis, because they are closed-form,

separable, and cheap to evaluate on grids (Kerbl et al., 2023; Charlier et al., 2021; Feydy et al., 2020). Concretely, we represent the charge density as a 3D Gaussian mixture model:

$$\rho(\mathbf{r}) = \text{ReLU} \left(\sum_{A \in M} \sum_{j=0}^{N_A} w_{A,j} \mathcal{N}(\mathbf{r} | \boldsymbol{\mu}_{A,j}, \boldsymbol{\Sigma}_{A,j}) \right) \quad (9)$$

where ReLU is the standard rectified linear unit function used to prevent negative densities¹, $A \in M$ 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 B 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 (7) of the predicted electron density (9), 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* tensor, see appendix C 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. The initialization is important to the final model and is 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 (9) using a readout head layer. Other important changes to the default HotPP implementation are detailed in the paragraphs below, and ELECTRA otherwise follows 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. We use an atomic embedding function inspired by SpookyNet (Unke et al., 2021b) to represent the atom in terms of its nuclear charge and electron configuration (details are in Appendix G).

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. However, equivariant networks prohibit their outputs from having a lower symmetry than their inputs (Smidt et al., 2021; Xie & Smidt, 2024). In particular, *local reflection symmetries* lead to strong restrictions on the outputs of equivariant models. For example, in Figure 1a we plot all vector-valued outputs of a randomly initialized equivariant model with a planar molecule as input. Clearly, the outputs are constrained to the reflection plane, which would severely limit a density constructed based on a Gaussian centered on the predicted locations. See appendix D for a detailed

¹Since the ReLU clamp enforces non-negativity, it could in principle create zero-density plateaus with weak local gradients. In practice, this did not impede training: we trained with and without the ReLU clamp and observed no measurable differences, and thus include it here solely out of principle.

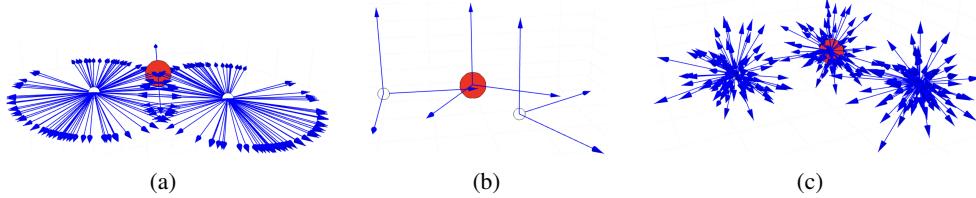


Figure 1: **(a)** Model output without symmetry-breaking: Equivariant neural networks constrain their output to have the same symmetry as the input. If the input molecule is highly symmetric, this leads to highly constrained Gaussian positions. **(b)** To solve this issue, ELECTRA initializes each atom's $l = 1$ 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.

mathematical explanation. 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:

$$I_{ij}^{(atom)} = \sum_{k=1}^N m_k \left(\|\mathbf{r}_k\|^2 \delta_{ij} - x_i^{(k)} x_j^{(k)} \right) \quad (10)$$

Where $k = 1, \dots, N$ runs over all atoms inside a local atomic neighborhood defined up to a cutoff radius from the current atom, and the vectors $\mathbf{r}_k = (x_1^{(k)}, x_2^{(k)}, x_3^{(k)})$ are calculated relative to the current atom. The three eigenvectors of (10) are also rotation equivariant. Thus, we can use them to initialize the first three $l = 1$ vector features of ELECTRA's GNN, from largest to smallest eigenvalue (Figure 1b), while maintaining rotational equivariance. They 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 Figure 1c. Similar ideas of using the moment of inertia vectors have been explored in slightly different contexts in previous work, for example Puny et al. (2021); Duval et al. (2023); Gao & Günemann (2021); Taniai et al. (2024). 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 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 \mathbf{v} , we switch the sign according to:

$$\mathbf{v}_{\text{canon}} = \begin{cases} \mathbf{v}, & \text{if } \mathbf{v} \cdot \mathbf{r}_{\text{COM}} \geq 0, \\ -\mathbf{v}, & \text{if } \mathbf{v} \cdot \mathbf{r}_{\text{COM}} < 0. \end{cases} \quad (11)$$

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. There are also some systems in which the moments of inertia degenerate. However, except for some special cases, for example, completely linear molecules, this rarely happens in practice. This is why many successful models rely on the moments of inertia.

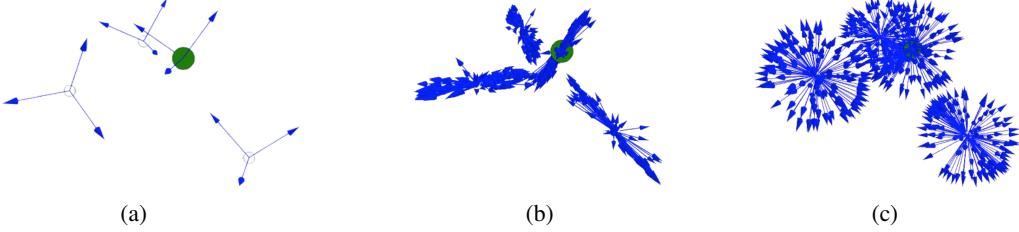


Figure 2: (a) The initial symmetry-breaking objects of the NH_3 molecule (Ammonia). (b) The output of a HotPP model with symmetry-breaking but without debiasing layers: The message passing induces a directional bias that concentrates vectors along certain directions, (c) The output of our model with 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 (see appendix D), we observe that the message-passing mechanism of HotPP induces a directional bias of the $l = 1$ features, particularly in highly symmetric molecules. In appendix E, we discuss the origin of this bias mathematically.

For example, in Figure 2b, we plot the output vectors from a randomly initialized HotPP model with the NH_3 molecule (Ammonia) and symmetry-breaking objects as input. We see, that the vectors tend to be parallel 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:

$$\mathbf{C}_A = \frac{1}{D} \sum_{j=1}^D \mathbf{v}_{A,j} \mathbf{v}_{A,j}^T \quad (12)$$

where $\mathbf{v}_{A,j}$ are the $l = 1$ features for atom A , and D is the channel dimension. We denote $\mathbf{u}_{1,A}$ as the eigenvector of \mathbf{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 :

$$\mathbf{v}_{A,j}^{\parallel} = (\mathbf{v}_{A,j} \cdot \mathbf{u}_{1,A}) \mathbf{u}_{1,A} \quad (13)$$

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 $l = 0$ features. 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:

$$\mathbf{v}_{A,j} \leftarrow \frac{\mathbf{v}_{A,j} - \mathbf{w}_{A,j} \cdot \hat{\mathbf{v}}_{A,j}^{\parallel}}{\|\mathbf{v}_{A,j} - \mathbf{w}_{A,j} \cdot \hat{\mathbf{v}}_{A,j}^{\parallel}\|}, \quad (14)$$

Where $\hat{\cdot}$ is the dot product. We normalize the vectors in (14) such that the $l = 1$ features only handle directionality, while scales are handled in the readout layer by $l = 0$ predictions. This mechanism, therefore, provides a way to determine and remove bias in the $l = 1$ features.

We note that the symmetry and bias limitations were not unique to the HotPP architecture, and we did initially try e.g. TensorNet (Simeon & De Fabritiis, 2024) and Equiformer v2 (Liao et al., 2023). However, these backbones faced the same issues as HotPP in terms of symmetry-breaking, and ultimately, we opted for HotPP as initial experiments suggested that it worked best.

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 $j \in [0, \dots, N_e(A) \times M_e]$, where N_e is the number of valence electrons of that atom, and M_e is the number of Gaussians per valence electrons. Intuitively, each of the three heads is specialized on a different distance scale away from the atoms. We use these predictions for the parameterization of the Gaussians in our ansatz (9). Each mean position is calculated as a weighted sum of three $l = 1$ features (one from each head). Similarly, each covariance matrix is a weighted sum of three symmetrized matrices based on the $l = 2$ features. The full details are available in Appendix F.

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:

$$\rho_{\text{pred}}(\mathbf{r}) = \rho(\mathbf{r}) \times \frac{n_{\text{elec}}}{\int_{\mathbb{R}^3} \rho(\mathbf{r}) dV}, \quad (15)$$

where dV represents the differential volume element on the grid. This ensures that

$$\int_{\mathbb{R}^3} |\rho_{\text{pred}}(\mathbf{r})| dV = n_{\text{elec}}, \quad (16)$$

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.

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

$$\mathcal{L} = \text{NMAE}(\rho_{\text{pred}}, \rho_{\text{ref}}) = \frac{\int_{\mathbb{R}^3} |\rho_{\text{ref}}(\mathbf{r}) - \rho_{\text{pred}}(\mathbf{r})| dV}{\int_{\mathbb{R}^3} |\rho_{\text{ref}}(\mathbf{r})| dV} \quad (17)$$

It is not necessary to compute the denominator in (17) during training since the reference grid must integrate to the number of valence electrons, i.e., $\int_{\mathbb{R}^3} |\rho_{\text{ref}}(\mathbf{r})| dV = n_{\text{elec}}$, and thus the denominator integral can be replaced with n_{elec} during training.

4 Experiments

Dataset and implementation. We train ELECTRA using the method outlined in Section 3 on reference densities from the QM9 density files. This charge density dataset is the most commonly used benchmark for charge density prediction models. The dataset was 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 test molecules, following the same scheme as in previous related work (Jørgensen & Bhowmik, 2022; Koker et al., 2024; Kim & Ahn, 2024; Cheng & Peng, 2024; Fu et al., 2024). On the QM9 dataset, we train ELECTRA for 864 GPU hours on a single NVIDIA RTX 3090-24GB GPU. We train using a batch size of 1, and do inference on the full grid for both training, validation and test. In comparison to our training protocol, the similar current state-of-the-art method from Fu et al. (2024) was trained for around 1152 GPU hours on NVIDIA A100-80GB GPUs. To further validate ELECTRA, we also perform experiments on the MD dataset consisting of different conformations of six different molecules Bogojeski et al. (2020); Brockherde et al. (2017). A full list of our hyperparameters is given in Table 3 in the Appendix. As an example of how ELECTRA distributes Gaussians around a molecule, we provide Figure 3, which shows the predicted and ground truth densities for $\text{C}_7\text{H}_9\text{NO}$ from the QM9 test set, along with the individual placement of all Gaussians and an error isosurface that visualizes the density errors.

Benchmark results. In Table 1 we report the mean accuracy in NMAE [%] (Equation 17) and average inference time per molecule (t_{inf}) on the QM9 test dataset of the main ELECTRA model 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, i.e., on RTX 3090-24GB GPUs. In Appendix J we report ablated versions of ELECTRA, which demonstrate the influence of the floating orbitals themselves, as well as symmetry-breaking and debiasing layers.

Table 1: QM9 test set performance of various models. **Top:** Results reproduced from Fu et al., 2024 (A100-80GB GPU). \dagger : This version of InfGCN only used Gaussian Type Orbitals (GTOs). **Bottom:** ELECTRA and SCDP (RTX 3090-24GB GPU, this work)

Metric	i-DeepDFT	e-DeepDFT	GPWNO	InfGCN \dagger	InfGCN	ChargE3Net
NMAE [%] \downarrow	0.357	0.284	0.730	3.720	0.869	0.196
t_{inf} [s] \downarrow	–	–	–	–	0.833	15.18
Metric	ELECTRA (ours)	SCDP ($L = 3$)	SCDP+BO ($L = 6$)			
NMAE [%] \downarrow	0.176	0.432	0.178			
t_{inf} [s] \downarrow	0.089	<u>0.395</u>	1.022			

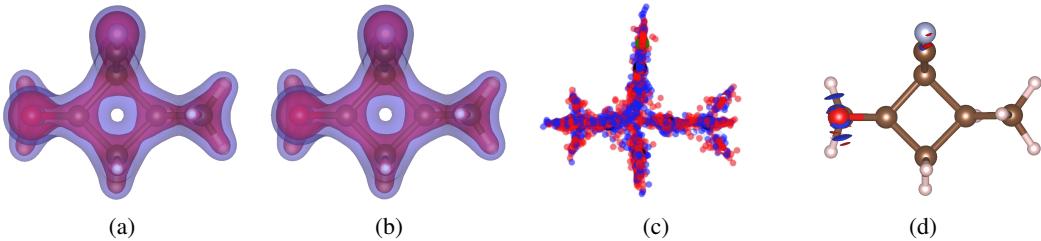


Figure 3: (a) Predicted density for C₇H₉NO using ELECTRA (NMAE = 0.19 %, red = high density, blue = lower density). (b) Ground-truth density. (c) Gaussian placements (red: $w_{A,j} > 0$, blue: $w_{A,j} < 0$). (d) 0.001 e/bohr³ error isosurfaces (blue = over-prediction, red = under-prediction).

Table 1 shows that ELECTRA is significantly more accurate than the DeepDFT, IncGCN and GPWNO models. Additionally, compared to all timed models in prior publications as well as this work, ELECTRA is faster by at least an order of magnitude. Compared to ChargE3Net, ELECTRA is 0.020 percentage points more accurate (0.176 vs 0.196), and roughly 170 times faster on inference, even when evaluated on inferior hardware (3090-24GB vs A100-80GB). Compared to SCDP, the current SOTA, ELECTRA is about 2.4 times more accurate while being 4.4 times faster compared to the fastest model ($L = 3$), and slightly more accurate than the best model ($L = 6 +$ bond-centered orbitals (BO)) while being over 11 times faster. On the MD dataset (Table 2), ELECTRA sets a new state-of-the-art on all six molecules by halving the NMAE % error compared to SCDP. In Figure 4, we also show how ELECTRA distributes the floating orbitals of a Benzene molecule, and how ELECTRA recreates the central density hole of the molecule. These fine details far away from any atom center are hard to model using only atom-centered orbitals. To test how close we are to the optimal solution allowed by our basis, we conduct overfitting experiments (Table 2), for details see H.

DFT initialization experiment. Because the iterations of self-consistent field calculations (SCF) dominate wall-clock time in routine electronic-structure DFT workflows, we assessed whether ELECTRA’s predicted densities accelerate SCF convergence on unseen molecules. Using VASP with the original dataset settings (Appendix K), we ran paired calculations initialized from (i) atomic superposition and (ii) ELECTRA-predicted densities, recording SCF iterations required to reach convergence at the same tolerance. The results showed that ELECTRA initialization reduces SCF steps by **50.72%** on average. Furthermore, we note that savings track density accuracy: the worst-error molecule (C₃H₅N₃O₂, NMAE = 1.43%), which is a significant outlier, shows only 17.4% reduction, whereas the best (C₉H₂O, NMAE = 0.153%) achieves a reduction in SCF steps of 60.87%.

ELECTRA uses floating orbitals and lower-order representations to achieve state-of-the-art density prediction accuracies while being an order of magnitude faster on inference than competing methods. This results in significant reductions in the number of SCF steps required to reach convergence when initializing new DFT calculations with predicted densities.

Table 2: NMAE[%] on MD test sets for ELECTRA, SCDP (Fu et al., 2024), GPWNO (Kim & Ahn, 2024) and InfGCN (Cheng & Peng, 2024), as well as the $N = 1$ Gaussian Overfit experiment. ELECTRA models were trained for 10 GPU hours.

Molecule	ELECTRA	SCDP	GPWNO	InfGCN	($N = 1$ Gaussian Overfit)
MD-ethanol	1.02	2.34	4.00	8.43	(0.18)
MD-benzene	0.45	1.13	2.45	5.11	(0.08)
MD-phenol	0.56	1.29	2.68	5.51	(0.08)
MD-resorcinol	0.62	1.35	2.73	5.95	(0.24)
MD-ethane	0.91	2.05	3.67	7.01	(0.28)
MD-malonaldehyde	0.80	2.71	5.32	10.34	(0.16)

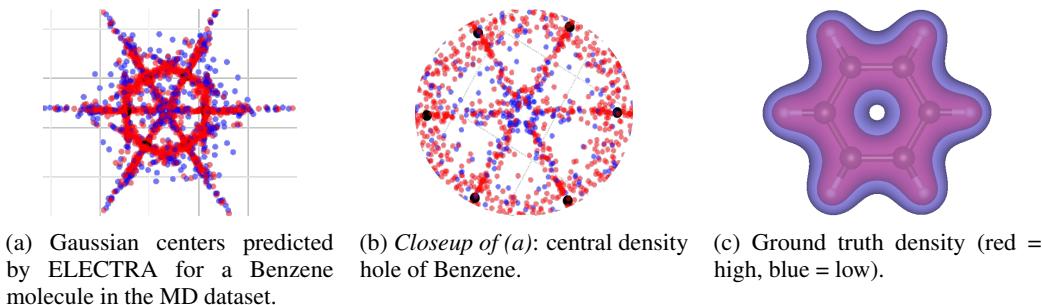


Figure 4: Gaussian placements vs. ground-truth electron density for Benzene.

5 Discussion

Since no existing basis sets or handcrafted orbital placements are used, ELECTRA is fully data-driven. It represents a shift from a human-designed to a machine-learned density representation, unlike the traditional DFT paradigm, where users must make heuristic element-based choices regarding the basis function type, size, and placement. However, there are still promising opportunities to explore.

Orbital placement. Fu et al. (2024) show that adding bond-centered orbitals increases expressivity. For this method to work, bonds must be identified in real-time during training and inference, and 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. For crystal lattices, issues 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 in theory. ELECTRA would theoretically be able to learn the non-local behavior of the density through (39) or variations thereof. However, this would require specialized methods to replicate floating orbitals periodically. Additionally, floating orbitals scale with the number of atoms rather than the number of bonds, making them computationally more efficient.

Hybrid Atom-Centered and Floating-Orbital Models. 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 that the benefits of floating orbitals could be combined with the flexibility of the spherical harmonics-based SCDP models. A model with both atom-centered and floating orbitals likely represents the most efficient use of computational resources, since even in ELECTRA, many orbitals are placed around atomic centers (Figures 1a and 3a).

Acknowledgements

The authors acknowledge financial support from the Independent Research Foundation Denmark, grant number 0217-00326B (DELIGHT), the Novo Nordisk Foundation grant number 87929 (NitroScale), and the Pioneer Center for Accelerating P2X Materials Discovery (CAPeX), DNRF grant number P3.

References

Bader, R., MacDougall, P., and Lau, C. Bonded and nonbonded charge concentrations and their relation to molecular geometry and reactivity. *Journal of the american Chemical Society*, 106(6):1594–1605, 1984.

Bader, R. F. and Essén, H. The characterization of atomic interactions. *The Journal of chemical physics*, 80(5):1943–1960, 1984.

Bader, R. F. and Nguyen-Dang, T. Quantum theory of atoms in molecules–dalton revisited. In *Advances in quantum chemistry*, volume 14, pp. 63–124. Elsevier, 1981.

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

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

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

Bogojeski, M., Vogt-Maranto, L., Tuckerman, M. E., Müller, K.-R., and Burke, K. Quantum chemical accuracy from density functional approximations via machine learning. *Nature communications*, 11(1):5223, 2020.

Boto, R. A., Piquemal, J.-P., and Contreras-García, J. Revealing strong interactions with the reduced density gradient: a benchmark for covalent, ionic and charge-shift bonds. *Theoretical Chemistry Accounts*, 136:1–9, 2017.

Brandstetter, J., Hesselink, R., van der Pol, E., Bekkers, E. J., and Welling, M. Geometric and physical quantities improve e (3) equivariant message passing. *arXiv preprint arXiv:2110.02905*, 2021.

Brehmer, J., Behrends, S., de Haan, P., and Cohen, T. Does equivariance matter at scale? *arXiv preprint arXiv:2410.23179*, 2024.

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

Brockherde, F., Vogt, L., Li, L., Tuckerman, M. E., Burke, K., and Müller, K.-R. Bypassing the kohn-sham equations with machine learning. *Nature communications*, 8(1):872, 2017.

Calcaterra, C. and Boldt, A. Approximating with gaussians. *arXiv preprint arXiv:0805.3795*, 2008.

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

Chandrasekaran, A., Kamal, D., Batra, R., Kim, C., Chen, L., and Ramprasad, R. Solving the electronic structure problem with machine learning. *npj Computational Materials*, 5(1):22, 2019.

Charlier, B., Feydy, J., Glaunès, J. A., Collin, F.-D., and Durif, G. Kernel operations on the gpu, with autodiff, without memory overflows. *Journal of Machine Learning Research*, 22(74):1–6, 2021. URL <http://jmlr.org/papers/v22/20-275.html>.

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

Constantin, L. A. and Ruzsinszky, A. Kinetic energy density functionals from the airy gas with an application to the atomization kinetic energies of molecules. *Physical Review B—Condensed Matter and Materials Physics*, 79(11):115117, 2009.

Cuevas-Zuviría, B. and Pacios, L. F. Machine learning of analytical electron density in large molecules through message-passing. *Journal of chemical information and modeling*, 61(6):2658–2666, 2021.

de Blasio, P. V. F., Jorgensen, P. B., Lastra, J. M. G., and Bhowmik, A. Nanosecond md of battery cathode materials with electron density description. *Energy Storage Materials*, 63:103023, 2023.

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

Deringer, V. L., Bartók, A. P., Bernstein, N., Wilkins, D. M., Ceriotti, M., and Csányi, G. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16):10073–10141, 2021.

Duval, A. A., Schmidt, V., Hernández-García, A., Miret, S., Malliaros, F. D., Bengio, Y., and Rolnick, D. Faenet: Frame averaging equivariant gnn for materials modeling. In *International Conference on Machine Learning*, pp. 9013–9033. PMLR, 2023.

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

Fabrizio, A., Grisafi, A., Meyer, B., Ceriotti, M., and Corminboeuf, C. Electron density learning of non-covalent systems. *Chemical science*, 10(41):9424–9432, 2019.

Febrer, P., Jørgensen, P., Pruneda, M., Garcia, A., Ordejon, P., and Bhowmik, A. Graph2mat: Universal graph to matrix conversion for electron density prediction. *Chemrxiv*, 2024.

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

Feydy, J., Glaunès, J., Charlier, B., and Bronstein, M. Fast geometric learning with symbolic matrices. *Advances in Neural Information Processing Systems*, 33, 2020.

Friederich, P., Häse, F., Proppe, J., and Aspuru-Guzik, A. Machine-learned potentials for next-generation matter simulations. *Nature Materials*, 20(6):750–761, 2021.

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

Fu, X., Rosen, A., Bystrom, K., Wang, R., Musaelian, A., Kozinsky, B., Smidt, T., and Jaakkola, T. A recipe for charge density prediction. *Advances in Neural Information Processing Systems*, 37: 9727–9752, 2024.

Gao, N. and Günemann, S. Ab-initio potential energy surfaces by pairing gnns with neural wave functions. *arXiv preprint arXiv:2110.05064*, 2021.

Geiger, M. and Smidt, T. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

Gong, S., Xie, T., Zhu, T., Wang, S., Fadel, E. R., Li, Y., and Grossman, J. C. Predicting charge density distribution of materials using a local-environment-based graph convolutional network. *Physical Review B*, 100(18):184103, 2019.

Grisafi, A., Fabrizio, A., Meyer, B., Wilkins, D. M., Corminboeuf, C., and Ceriotti, M. Transferable machine-learning model of the electron density. *ACS central science*, 5(1):57–64, 2018.

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

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

Huang, N., Levie, R., and Villar, S. Approximately equivariant graph networks. *Advances in Neural Information Processing Systems*, 36, 2024.

Iyengar, S. S., Ernzerhof, M., Maximoff, S. N., and Scuseria, G. E. Challenge of creating accurate and effective kinetic-energy functionals. *Physical Review A*, 63(5):052508, 2001.

Jørgensen, P. B. and Bhowmik, A. Deepdft: Neural message passing network for accurate charge density prediction. *arXiv preprint arXiv:2011.03346*, 2020.

Jørgensen, P. B. and Bhowmik, A. Equivariant graph neural networks for fast electron density estimation of molecules, liquids, and solids. *npj Computational Materials*, 8(1):183, 2022.

Kaba, S.-O. and Ravanbakhsh, S. Symmetry breaking and equivariant neural networks. *arXiv preprint arXiv:2312.09016*, 2023.

Kahle, L., Marcolongo, A., and Marzari, N. Modeling lithium-ion solid-state electrolytes with a pinball model. *Physical Review Materials*, 2(6):065405, 2018.

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

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

Kim, S. and Ahn, S. Gaussian plane-wave neural operator for electron density estimation. *arXiv preprint arXiv:2402.04278*, 2024.

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

Koch, D., Chaker, M., Ihara, M., and Manzhos, S. Density-based descriptors of redox reactions involving transition metal compounds as a reality-anchored framework: a perspective. *Molecules*, 26(18):5541, 2021.

Koch, U. and Popelier, P. L. Characterization of cho hydrogen bonds on the basis of the charge density. *The Journal of Physical Chemistry*, 99(24):9747–9754, 1995.

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

Koker, T., Quigley, K., Taw, E., Tibbetts, K., and Li, L. Higher-order equivariant neural networks for charge density prediction in materials. *npj Computational Materials*, 10(1):161, 2024.

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

Laubach, S., Laubach, S., Schmidt, P. C., Ensling, D., Schmid, S., Jaegermann, W., Thißen, A., Nikolowski, K., and Ehrenberg, H. Changes in the crystal and electronic structure of licoo 2 and linio 2 upon li intercalation and de-intercalation. *Physical Chemistry Chemical Physics*, 11(17):3278–3289, 2009.

Li, C., Sharir, O., Yuan, S., and Chan, G. K. Image super-resolution inspired electron density prediction. *arXiv preprint arXiv:2402.12335*, 2024.

Liao, Y.-L. and Smidt, T. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. *arXiv preprint arXiv:2206.11990*, 2022.

Liao, Y.-L., Wood, B., Das, A., and Smidt, T. EquiformerV2: Improved equivariant transformer for scaling to higher-degree representations. *arXiv preprint arXiv:2306.12059*, 2023.

Liu, J., Kumar, A., Ba, J., Kiros, J., and Swersky, K. Graph normalizing flows. *Advances in Neural Information Processing Systems*, 32, 2019.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

Lorincz, B. D. and Nagy, P. R. 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.

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

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

Mi, W., Luo, K., Trickey, S., and Pavanello, M. Orbital-free density functional theory: An attractive electronic structure method for large-scale first-principles simulations. *Chemical Reviews*, 123(21):12039–12104, 2023.

Perdew, J. P., Levy, M., Painter, G., Wei, S., and Lagowski, J. B. Chemical bond as a test of density-gradient expansions for kinetic and exchange energies. *Physical Review B*, 37(2):838, 1988.

Perdew, J. P., Burke, K., and Ernzerhof, M. Generalized gradient approximation made simple. *Physical review letters*, 77(18):3865, 1996.

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

Puny, O., Atzmon, M., Ben-Hamu, H., Misra, I., Grover, A., Smith, E. J., and Lipman, Y. Frame averaging for invariant and equivariant network design. *arXiv preprint arXiv:2110.03336*, 2021.

Qiao, Z., Christensen, A. S., Welborn, M., Manby, F. R., Anandkumar, A., and Miller III, T. F. Informing geometric deep learning with electronic interactions to accelerate quantum chemistry. *Proceedings of the National Academy of Sciences*, 119(31):e2205221119, 2022.

Rackers, J. A., Tecot, L., Geiger, M., and Smidt, T. E. A recipe for cracking the quantum scaling limit with machine learned electron densities. *Machine Learning: Science and Technology*, 4(1):015027, 2023.

Ragoza, M., Masuda, T., and Koes, D. R. Generating 3d molecules conditional on receptor binding sites with deep generative models. *Chemical science*, 13(9):2701–2713, 2022.

Ribaldone, C. and Desmarais, J. K. Spherical to cartesian coordinates transformation for solid harmonics revisited: Construction of the hartree potential. *The Journal of Chemical Physics*, 163(7), 2025.

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

Shang, T., Xiao, D., Meng, F., Rong, X., Gao, A., Lin, T., Tang, Z., Liu, X., Li, X., Zhang, Q., et al. Real-space measurement of orbital electron populations for li_{1-x} co₂. *Nature communications*, 13(1):5810, 2022.

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

Smidt, T. E., Geiger, M., and Miller, B. K. Finding symmetry breaking order parameters with euclidean neural networks. *Physical Review Research*, 3(1):L012002, 2021.

Taniai, T., Igarashi, R., Suzuki, Y., Chiba, N., Saito, K., Ushiku, Y., and Ono, K. Crystalformer: infinitely connected attention for periodic structure encoding. *arXiv preprint arXiv:2403.11686*, 2024.

Tao, F.-M. 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.

Tao, F.-M. and Pan, Y.-K. Mo/lle–plessset perturbation investigation of the he₂ potential and the role of midbond basis functions. *The Journal of chemical physics*, 97(7):4989–4995, 1992.

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

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Unke, O., Bogojeski, M., Gastegger, M., Geiger, M., Smidt, T., and Müller, K.-R. Se (3)-equivariant prediction of molecular wavefunctions and electronic densities. *Advances in Neural Information Processing Systems*, 34:14434–14447, 2021a.

Unke, O. T., Chmiela, S., Gastegger, M., Schütt, K. T., Sauceda, H. E., and Müller, K.-R. Spookynet: Learning force fields with electronic degrees of freedom and nonlocal effects. *Nature communications*, 12(1):7273, 2021b.

Unke, O. T., Chmiela, S., Sauceda, H. E., Gastegger, M., Poltavsky, I., Schütt, K. T., Tkatchenko, A., and Müller, K.-R. Machine learning force fields. *Chemical Reviews*, 121(16):10142–10186, 2021c.

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

Wang, J., Wang, Y., Zhang, H., Yang, Z., Liang, Z., Shi, J., Wang, H.-T., Xing, D., and Sun, J. E (n)-equivariant cartesian tensor message passing interatomic potential. *Nature Communications*, 15(1):7607, 2024.

Wang, L., Bai, R., Shi, X., Zhang, W., Cui, Y., Wang, X., Wang, C., Chang, H., Zhang, Y., Zhou, J., et al. A pocket-based 3d molecule generative model fueled by experimental electron density. *Scientific reports*, 12(1):15100, 2022.

Weigend, F. and Ahlrichs, R. 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.

Weizsäcker, C. v. Zur theorie der kernmassen. *Zeitschrift für Physik*, 96(7):431–458, 1935.

Xie, Y. and Smidt, T. Equivariant symmetry breaking sets, 2024. URL <https://arxiv.org/abs/2402.02681>.

Zheng, Q., Feng, T., Hachtel, J. A., Ishikawa, R., Cheng, Y., Daemen, L., Xing, J., Idrobo, J. C., Yan, J., Shibata, N., et al. Direct visualization of anionic electrons in an electride reveals inhomogeneities. *Science Advances*, 7(15):eabe6819, 2021.

Zheng, Y., Jiao, Y., Zhu, Y., Li, L. H., Han, Y., Chen, Y., Du, A., Jaroniec, M., and Qiao, S. Z. Hydrogen evolution by a metal-free electrocatalyst. *Nature communications*, 5(1):3783, 2014.

A Group theory

We provide a short introduction to the relevant group theoretical background, mainly adapted from (Liao et al., 2023; Liao & Smidt, 2022).

Definition of Groups. A *group* is an algebraic structure consisting of a set G together with a binary operation $* : G \times G \rightarrow G$. A group, satisfies the following four axioms:

1. **Closure:** For all $g, h \in G$, we have $g * h \in G$.
2. **Identity:** There exists an element $e \in G$ such that $g * e = e * g = g$ for all $g \in G$.
3. **Inverse:** For every $g \in G$, there exists an element $g^{-1} \in G$ such that $g * g^{-1} = g^{-1} * g = e$.
4. **Associativity:** For all $g, h, i \in G$, we have $g * (h * i) = (g * h) * i$.

Symmetry Groups in Three Dimensions. In this work, we focus on three-dimensional Euclidean symmetry, where the relevant groups are:

1. The Euclidean group in three dimensions $E(3)$: rotations, translations, and inversion in \mathbb{R}^3 .
2. The special Euclidean group in three dimensions $SE(3)$: rotations and translations in \mathbb{R}^3 .
3. The orthogonal group in three dimensions $O(3)$: rotations and inversion in \mathbb{R}^3 .
4. The special orthogonal group in three dimensions $SO(3)$: rotations in \mathbb{R}^3 .

Group Representations. Let X be a vector space. A *representation* of a group G on X is a map

$$D_X : G \rightarrow \text{End}(X),$$

where for each $g \in G$, $D_X(g) : X \rightarrow X$ is a linear transformation. In practice, $D_X(g)$ is represented as a matrix acting on X , so that group actions are matrix multiplications. These representations respect the group structure: for all $g, h \in G$,

$$D(g)D(h) = D(g * h).$$

Two representations $D(g)$ and $D'(g)$ are said to be *equivalent* if there exists an invertible $N \times N$ change-of-basis matrix P such that

$$P^{-1}D(g)P = D'(g) \quad \text{for all } g \in G.$$

A representation $D(g)$ is called *reducible* if it can be transformed into block-diagonal form, i.e.,

$$P^{-1}D(g)P = \begin{bmatrix} D_1(g) & 0 \\ 0 & D_2(g) \end{bmatrix},$$

so that it acts independently on multiple subspaces of X . Otherwise, the representation is *irreducible*. Such *irreducible representations* (irreps) are especially useful for building and classifying group representations.

In the case of $SO(3)$, the Wigner D -matrices provide a family of irreducible representations. Any representation of $SO(3)$ can be expressed as a direct sum (concatenation) of these irreps:

$$D(g) = P^{-1} \left(\bigoplus_i D^{(l_i)}(g) \right) P = P^{-1} \begin{bmatrix} D^{(l_0)}(g) & & & \\ & D^{(l_1)}(g) & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} P, \quad (18)$$

where $D^{(l)}(g)$ denotes the Wigner D -matrix of degree l .

The degree l can be thought of as frequencies in a Fourier decomposition; the higher the l the higher frequency.

B Gaussians as Cartesian basis functions

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

$$\Phi_{\alpha,l,m}(\mathbf{r}) = R_{\alpha,l}(|\mathbf{r}|)Y_{lm}(\theta, \phi) \quad (19)$$

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, a popular alternative in quantum chemistry is Cartesian basis functions. With $\mathbf{r} = (x, y, z)$ Cartesian basis functions can be written as:

$$\Phi_{\alpha,n_x,n_y,n_z}(x, y, z) = N(\alpha, n_x, n_y, n_z)R_{\alpha}(|\mathbf{r}|)x^{n_x}y^{n_y}z^{n_z} \quad (20)$$

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 $l = n_x + n_y + n_z$. Note that there is a strong relation between the angular momentum quantum number for spherical basis functions and Cartesian basis functions. For each l , there is an invertible transform between the spherical l -shell and the harmonic part of the cartesian l -shell (Ribaldone & Desmarais, 2025).

We can collect all the terms above belonging to the same l in one tensor:

$$\Phi_{\alpha,l}(x, y, z) = \mathbf{N}(\alpha, l) \cdot R_{\alpha}(|\mathbf{r}|) \underbrace{\mathbf{r} \otimes \mathbf{r} \otimes \dots \otimes \mathbf{r}}_{l \text{ times}} \quad (21)$$

In particular, for $l = 2$ we get

$$\Phi_{l=2}(x, y, z) = \mathbf{N}(\alpha, 2) \cdot R_{\alpha}(|\mathbf{r}|) \mathbf{r} \mathbf{r}^{\top} \quad (22)$$

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

$$C_{i,j}(\Phi_{l=2})_{i,j} = R_{\alpha}(|\mathbf{r}|) \mathbf{r}^{\top} \underbrace{(\mathbf{N}(\alpha, l) \cdot \mathbf{C}) \mathbf{r}}_{:= -2\mathbf{\Sigma}^{-1}} \quad (23)$$

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

$$\exp\left(-\frac{1}{2}\mathbf{r}^{\top}\mathbf{\Sigma}^{-1}\mathbf{r}\right) \propto \mathcal{N}(\mathbf{r}|0, \mathbf{\Sigma}) \quad (24)$$

C Equivariance of Gaussians leads to invariant density

The electron density is rotation invariant. In the main text, we claimed that our ansatz (9)

$$\rho(\mathbf{r}) = \sum_{A \in M} \sum_{j=0}^{N_A} w_{A,j} \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_{A,j}, \mathbf{\Sigma}_{A,j}) \quad (25)$$

is rotation invariant, if the weights $w_{A,j}$ are rotation invariant, and the position $\boldsymbol{\mu}_{A,j}$ and covariance matrices $\mathbf{\Sigma}_{A,j}$ 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

$$\mathcal{N}(\mathbf{r}|\boldsymbol{\mu}, \mathbf{\Sigma}) = \mathcal{N}(\mathbf{R}\mathbf{r}|\mathbf{R}\boldsymbol{\mu}, \mathbf{R}\mathbf{\Sigma}\mathbf{R}^{\top}) \quad (26)$$

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

$$\mathcal{N}(\mathbf{R}\mathbf{r}|\mathbf{R}\boldsymbol{\mu}, \mathbf{R}\mathbf{\Sigma}\mathbf{R}^{\top}) = \exp\left(-\frac{1}{2}(\mathbf{R}(\mathbf{r} - \boldsymbol{\mu}))^{\top}(\mathbf{R}\mathbf{\Sigma}\mathbf{R}^{\top})^{-1}\mathbf{R}(\mathbf{r} - \boldsymbol{\mu})\right) \quad (27)$$

$$= \exp\left(-\frac{1}{2}(\mathbf{r} - \boldsymbol{\mu})^{\top}\mathbf{R}^{\top}\mathbf{R}\mathbf{\Sigma}^{-1}\mathbf{R}^{\top}\mathbf{R}(\mathbf{r} - \boldsymbol{\mu})\right) \quad (28)$$

$$= \exp\left(-\frac{1}{2}(\mathbf{r} - \boldsymbol{\mu})^{\top}\mathbf{\Sigma}^{-1}(\mathbf{r} - \boldsymbol{\mu})\right) \quad (29)$$

$$= \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}, \mathbf{\Sigma}) \quad (30)$$

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

D Local reflection symmetry constraints

We claim that "local", vide infra, reflection symmetries lead to restrictions on the expressivity of equivariant neural networks. Equivariant neural networks are constrained such that the output transforms with the input:

$$f(gG) = g f(G) \quad (31)$$

where G describes our molecules and g is the action of a group. In particular, we are interested in $g \in O(3)$, so equivariance to rotations and reflections, in which case we have:

$$f_{l,p}(\mathbf{R}G) = \det(\mathbf{R})^p D_l(\mathbf{R}') f_{l,p}(G) \quad (32)$$

where $D_l(\mathbf{R})$ is the Wigner-D matrix of a rotation, $\mathbf{R}' \cdot \det(\mathbf{R}) = \mathbf{R}$ is the rotation part of an (im)proper rotation and p is the parity of the output, with $p = 1$ called odd and $p = 0$ called even. Next, we define local invariance:

Definition D.1. Let $G = \{(r_i - r_j), Z_i, Z_j\}_{i,j=0,\dots,N-1}$ be the description of a molecule using pairwise displacement vectors and associated atom types. Let $G_i = \{(r_i - r_j), Z_i, Z_j\}_{j \in \mathcal{N}(i,r,L)} \subset G$ be the subset of the graph with nodes within some neighborhood $\mathcal{N}(i,r,L)$ of node i , defined by L -hop message passing with cutoff distance r . Let \mathbf{R}_P be the representation of some reflection symmetry through some arbitrary plane. We call G_i locally symmetric/invariant under \mathbf{R}_P if $\mathbf{R}_P G_i = G_i$.

Importantly, part of a molecule can be locally symmetric, even if the molecule as a whole does not have the same symmetry globally. Take, for example, water in its ground state geometry with O being located at $(0, 0, 0)$ and H_1 and H_2 lying in the xz-plane. Then $G_{H_1} = \{(r_{H_1} - r_O, 0), (r_{H_1} - r_{H_2}, H_2)\}$ is locally symmetric to reflections along the xz-plane, represented by the matrix

$$\sigma_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (33)$$

and we have $\sigma_v G_{H_1} = G_{H_1}$, so G_{H_1} is locally symmetric under σ_v .

Local symmetry leads to constraints on the outputs of $f(G_i)$. In particular, the output of an equivariant function needs to be an eigenfunction of the symmetry operator R_P with eigenvalue 1:

$$f_{l,p}(\mathbf{R}_P G_i) = f_{l,p}(G_i) = \det(\mathbf{R}_P)^p D_l(\mathbf{R}'_P) f_{l,p}(G_i) \quad (34)$$

For example, in our water molecule example, we have $R_P = \sigma_v$. For $l = 1$ we have $D_l(\mathbf{R}') = \mathbf{R}'$ and therefore for our $p = 1, l = 1$ we get

$$f_{1,1}(\mathbf{R}_P G_i) = f_{1,1}(G_i) = \det(\mathbf{R}_P) \mathbf{R}'_P f_{1,1}(G_i) = \mathbf{R}_P f_{1,1}(G_i) \quad (35)$$

Reflection operators in three dimensions always have exactly one eigenvalue of -1 and two eigenvalues of 1. Therefore, the requirement above restricts the output of the equivariant network to live in the degenerate subspace with eigenvalue 1. For example, the eigenspace of σ_v for the eigenvalue 1 is spanned by $(1,0,0)$ and $(0,0,1)$. This means any vector in the xz-plane is an eigenvector of σ_v . This is exactly what we are seeing in Figure 1a, where all output vectors from the hydrogen atom lie in the xz-plane. It becomes even more restrictive for neighborhoods with more than one reflection symmetry. For example, G_O , the descriptor around the Oxygen atom in our water example, has two symmetries. In general, the highest number of symmetries we can have in a molecule is 9 (the full octahedral group O_h). However, for most organic molecules it will be at most 6, for example, CH_4 with the tetrahedral group T_d . If we have more than one reflection symmetry, we have more eigenvalue equations of the form 34, one for each symmetry. Our output has to fulfill all of them simultaneously. This lowers the dimension of the allowed subspace for the output of our equivariant functions by one, since the output needs to live in the union of the allowed subspaces from each reflection individually. For example, if we have both $\sigma_v(xz)$ and $\sigma_v(yz)$ our $l = 1, p = 1$ outputs need to live in the intersection $\text{span}((1, 0, 0), (0, 0, 1)) \cap \text{span}((1, 0, 0), (0, 1, 0)) = \text{span}((1, 0, 0))$, which again limits the expressivity of our function. An example of this can be seen on the oxygen atom in Figure 1a.

The discussion shows that local reflection symmetries lead to restrictions on the outputs of equivariant networks. This is a problem since we also need to place Gaussians outside the xz-plane to model the density faithfully.

By augmenting G_{H_1} with a set of vectors $\{s_k\}_{k=0,\dots,K}$: $G'_{H_1} = G_{H_1} \cup \{s_k\}_{k=0,\dots,K}$, that are not eigenvectors to σ_v , we get $\sigma_v G'_{H_1} \neq G'_{H_1}$, and therefore break the symmetry and increase expressivity. Since reflections are never threefold degenerate, such vectors always exist. Given an orthonormal coordinate system as in Figure 1(b), we can learn any vector with a simple linear layer, including $\{s_k\}_{k=0,\dots,K}$, and therefore break any local reflection symmetry.

E Directional bias

As described in the main text 3.1, we observe that message passing in equivariant networks induces a directional bias, see figure 2b. While this is mainly an empirical observation, by examining the message construction in equivariant networks, we get an idea of the origin of this bias. Consider the vector ($l = 1$) valued messages $m_{ij,c}^{l=1}$ from node j to node i with channel dimension c , displacement vector between nodes \mathbf{r}_{ij} , neural network $f_{l_r}(\cdot)$ and node features $h_{l_r,j,c}$:

$$m_{ij,c}^{l=1} = \underbrace{f_{l_r=0}(|\mathbf{r}_{ij}|)h_{l_r=0,j,c}}_{\text{Scalars} = :s_c} \mathbf{r}_{ij} + \underbrace{f_{l_r=1}(|\mathbf{r}_{ij}|)h_{l_r=1,j,c}}_{\text{randomly initialized}} + \text{Higher body terms} \quad (36)$$

$$= s_c \underbrace{\mathbf{r}_{ij}}_{\text{observed bias direction}} + \text{random directions}_c \quad (37)$$

So in each layer, we add the displacement vectors while everything else is randomly initialized, leading to the observed bias.

F Density construction details

Scalar factors. As a first step, we use the scalar features together with the node embedding \mathbf{f}_j to construct an input for three different MLPs: $\mathbf{s}_{\text{inp},A,j} = [\mathbf{f}_j, \mathbf{s}_{1A,j}, \mathbf{s}_{2A,j}, \mathbf{s}_{3A,j}]$. The MLPs then predict the Gaussian mixture weights $w_{A,j}$ together with two other sets of scalars to use in the mean and covariance predictions:

$$\begin{aligned} \mathbf{s}_{p,A,j} &\in \mathbb{R}^3 = \text{MLP}_p(\mathbf{s}_{\text{inp},A,j}), \\ \mathbf{s}_{m,A,j} &\in \mathbb{R}^3 = \text{MLP}_m(\mathbf{s}_{\text{inp},A,j}), \\ w_{A,j} &\in \mathbb{R} = \text{MLP}_w(\mathbf{s}_{\text{inp},A,j}). \end{aligned} \quad (38)$$

Gaussian positions. ELECTRA places Gaussians (i.e., predicts the mean positions $\boldsymbol{\mu}_{A,j}$) equivariantly by using the $l = 1$ outputs $\mathbf{v}_{1,A,j}, \mathbf{v}_{2A,j}, \mathbf{v}_{3A,j}$ and the position scaling factors \mathbf{s}_p of the framework as displacement vectors to the atomic positions:

$$\begin{aligned} \boldsymbol{\mu}_{A,j} &= \mathbf{r}_A + \exp(\mathbf{s}_{p1A,j})\mathbf{v}_{1A,j} \\ &\quad + \mathbf{s}_{p2A,j}^2 \mathbf{v}_{2A,j} + \mathbf{s}_{p3A,j} \mathbf{v}_{3A,j} \end{aligned} \quad (39)$$

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 $\mathbf{M} \rightarrow \mathbf{M}\mathbf{M}^\top$. For notational simplicity, we omit the A, j subscript for all matrices and scalars in the equations below:

$$\begin{aligned} \mathbf{s}_{G_1}, \mathbf{s}_{G_2}, \mathbf{s}_{G_3} &= \text{softmax}([\mathbf{s}_{m_1}, \mathbf{s}_{m_2}, \mathbf{s}_{m_3}]) \\ \boldsymbol{\Sigma} &= \mathbf{s}_{G_1} \frac{\mathbf{M}_1 \mathbf{M}_1^\top}{\|\mathbf{M}_1\|_F} + \mathbf{s}_{G_2} \frac{\mathbf{M}_2 \mathbf{M}_2^\top}{\|\mathbf{M}_2\|_F} + \mathbf{s}_{G_3} \frac{\mathbf{M}_3 \mathbf{M}_3^\top}{\|\mathbf{M}_3\|_F}. \end{aligned} \quad (40)$$

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

G Embedding function

To encode information about the electronic and nuclear properties of each atom into the scalar features, ELECTRA uses an embedding function similar to SpookyNet(Unke et al., 2021b), which uses nuclear and electronic embeddings to encode information about the ground state electron configuration of each element. In ELECTRA, each atom’s electronic and nuclear properties are encoded as $\mathbf{f} = [P, N, V, E_1, \dots, E_n, F_1, \dots, F_n]$, 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, for each orbital. A multi-layer perceptron (MLP) maps \mathbf{f} to the embedding space, $s_{\text{init}} = \text{MLP}_{\text{Emb}}(\mathbf{f})$, which is used as the initial scalar features.

H Basis limitation experiment.

Errors are markedly higher on the MD dataset than on QM9 for ELECTRA and baselines. Specifically for ELECTRA, we tested whether this reflects a basis limit of our simplified Gaussian ansatz (Eq. 9). Thus, for each molecule, we overfit an unconstrained Gaussian mixture with the same number of components as ELECTRA, initializing means as atom-center displacements. Using identical loss and optimizer, we trained for 100,000 steps on a single snapshot per molecule. As Table 2 shows, this overfit GMM achieves much lower errors, indicating that the gap is predominantly model error rather than a limitation of the Gaussian basis. Given GMM non-convexity and local minima, these fits are a conservative lower bound on basis expressivity, and we expect that longer or stabilized schedules could do better, and the results are consistent with the theoretical universality of Gaussian mixtures in $L^2(\mathbb{R}^3)$ (Calcaterra & Boldt, 2008).

I Hyperparameter table

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

Table 3: Hyperparameters for the main ELECTRA model.

Hyperparameter	Value
Graph radius cutoff	8.0 Å
HotPP # body layers	3
Gaussians per electron (M_e)	90
Graph network channel width	750
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 $LR_{initial}$	3.5×10^{-5}
Learning rate scheduler	Linear ($LR = LR_{initial} \times \gamma^{Epoch}$)
Learning rate gamma (γ)	0.85

J Ablation studies

In Table 4 we ablate the main ELECTRA model in five different ways. Not allowing for negative Gaussians ("No negative Gaussians" in the Table) means that the weights $w_{A,j}$ in Equation 9 are restricted to being positive. Removal of scaling ("No scaling") means removing the predicted scaling factors from the displacement vectors in Equation 39 and from the weighted sum of symmetrized covariance matrices in Equation 40. Removal of the debiasing layers is self-explanatory, while removal of the symmetry-breaking mechanism ("No symmetry-breaking") means that we do not initialize the $l = 1$ features of the GNN with the moment of inertia eigenvectors as described in Section 3. Finally, removal of floating orbitals ("No floating orbitals") means that the displacement vectors of Equation 39 are set to zero such that for the the mean position of the j 'th Gaussian of the A 'th atom, $\mu_{A,j} = r_A$. I.e., all Gaussians become atom-centered.

Table 4: Ablation studies of ELECTRA.

Model	NMAE [%] ↓
ELECTRA	0.176
ELECTRA - No negative Gaussians	2.027
ELECTRA - No scaling	0.356
ELECTRA - No debiasing layers	0.584
ELECTRA - No debiasing layers and no symmetry-breaking	0.705
ELECTRA - No floating orbitals	6.069

K QM9 DFT initialization experiment parameters

The parameters used to initialize fresh DFT runs with or without an initial density follow the original parameters used by Jørgensen & Bhowmik (2022) to create the QM9 dataset. They are reported in Table 5 below.

Table 5: VASP parameters used to reproduce the DFT initialization experiment on QM9. Baseline uses **ICHARG=2** (atomic superposition). ELECTRA-initialized runs use **ICHARG=1** to read the predicted CHGCAR as the starting density; all other tags are identical.

Setting (shorthand)	Value
Exchange–correlation functional (XC)	PBE
Job start mode (ISTART)	0
Electronic minimization algorithm (ALGO)	Normal
Charge density initialization (ICHARG)	2
Spin treatment (ISPIN)	1
Max electronic steps (NELM)	180
Initial non-SCF steps (NELMDL)	6
Symmetry usage (ISYM)	0
Correction output (LCORR)	TRUE
Ionic time step (POTIM)	0.1
Min electronic steps (NELMIN)	5
k -point mesh (KPOINTS)	$1 \times 1 \times 1$
Smearing method (ISMEAR)	0
Electronic conv. criterion (EDIFF)	1.0×10^{-6}
Smearing width (SIGMA)	0.1
Ionic steps (NSW)	0
Subspace diagonalization (LDIAG)	TRUE
Real-space projectors (LREAL)	Auto
Write wavefunctions (LWAVE)	FALSE
Write charge density (LCHARG)	TRUE
Plane-wave cutoff (ENCUT)	400

L Broader impact

This paper and the ELECTRA model presented in it can dramatically speed up the prediction of charge densities and can potentially be applied to model other 3D densities. Charge density prediction is a foundational problem in computational chemistry, and by accelerating it our approach can advance scientific discovery, though its impact — positive or negative — depends on how it's applied. The greatest benefit is in materials design and drug-discovery workflows that address real-world challenges in e.g. green transition or medicine.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: **[Yes]**

Justification: Our main claim is that you can learn floating orbitals to efficiently represent charge densities. We verify this on several datasets and show that our method competes for state-of-the-art accuracy while being orders of magnitude faster.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: **[Yes]**

Justification: In our Methods section, we described how the symmetry-breaking mechanism relies on non-degenerate moment-of-inertia tensors. We also point out the issues of canonicalization of these vectors. Finally, while we outlined the theoretical advantage of ELECTRA compared to other models, it will need further work to be adapted to periodic systems.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: As in the previous section, we point out our assumptions for the canonicalization of the moment-of-inertia eigenvectors. Appendices A, B and C further provide mathematical background for our model.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have described the main steps needed to construct the model we propose in the paper, including the graph neural network backbone, the density construction method and the symmetry-breaking mechanisms. Furthermore, the code is available at https://github.com/Jotels/ELECTRA_2025.git.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: **[Yes]**

Justification: All data used in the paper is publicly available (Jørgensen & Bhowmik, 2022; Bogojeski et al., 2020; Brockherde et al., 2017), and as mentioned above the code is available at https://github.com/Jotels/ELECTRA_2025.git.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: **[Yes]**

Justification: Section 4, Appendix I.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: **[No]**

Justification: Error bars are not reported because it would be too computationally expensive. However, we are using the same standard split that all competing models use, and the improvements (in particular, the inference times) are orders of magnitude better than prior work. This makes it unlikely that the results are not statistically valid.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: Section 4 provides details on the GPU type and how many GPU hours we used to train all models.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: We believe that we conform to the NeurIPS Code of Ethics in every way.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: Appendix L

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [\[NA\]](#)

Justification: We believe that our paper poses no such risk.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: All creators of models, data and code used to build the ELECTRA model have been properly cited in this paper. Specific Python libraries used to build the code are included in our codebase.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: **[Yes]**

Justification: As mentioned, our code is available at https://github.com/Jotels/ELECTRA_2025.git, and has been deposited under an MIT License.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: **[NA]**

Justification: Our work includes not research with human subjects and has not been crowd-sourced in any way.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: **[NA]**

Justification: See above (Question 14).

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development of our model and paper did not involve LLMs in any capacity.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.