Electronic **T**ensor **R**econstruction **A**lgorithm (**ELECTRA**)

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

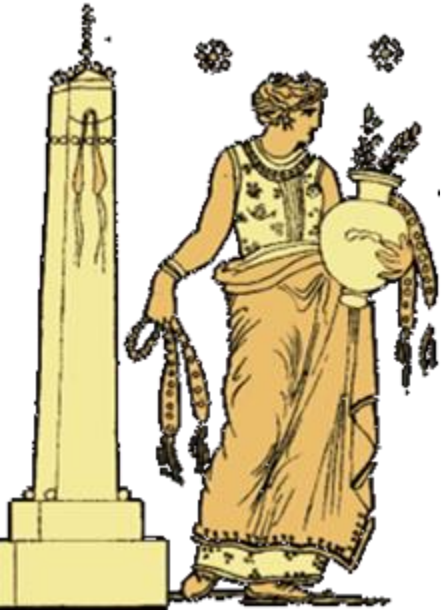
Jonas Elsborg, Luca Thiede, Alán Aspuru-Guzik, Tejs Vegge, Arghya Bhowmik

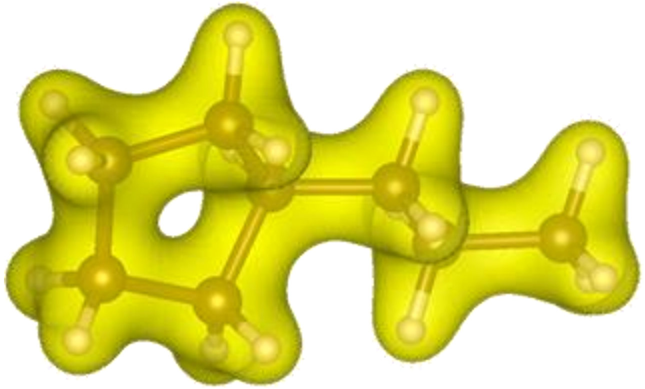
The bitter lesson of machine learning:

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin.

... Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation.

- Richard Sutton, 2019



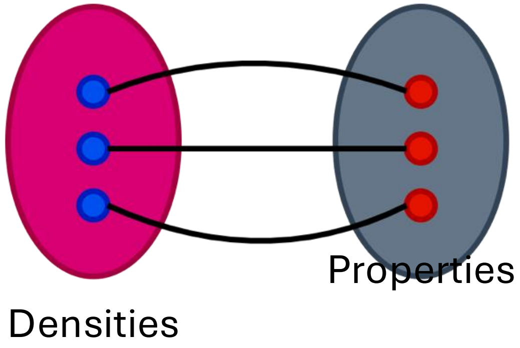


Learning densities via deep learning on graphs

DFT

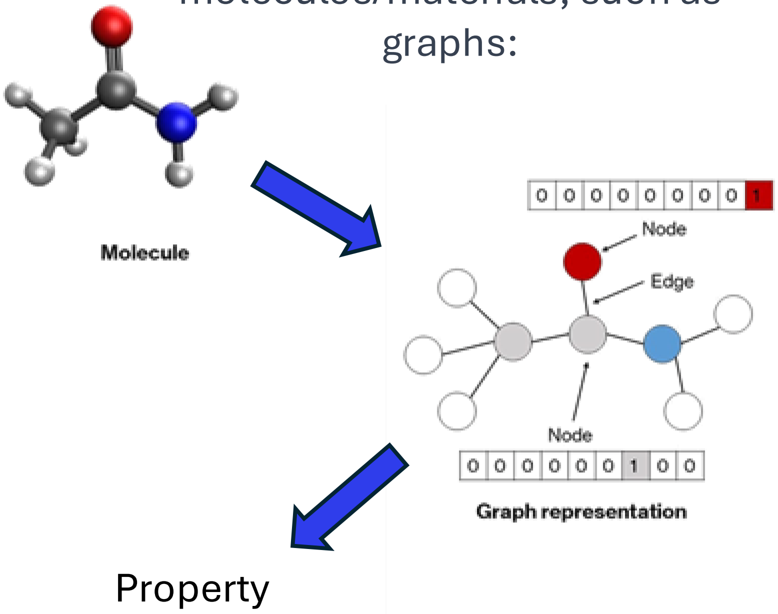
The road to fast and accurate ML-driven quantum chemistry requires efficient predictions of ground-state properties via electron density.

DFT works by constructing bijective maps from electron densities to properties:



Canonical quantum chemistry ML

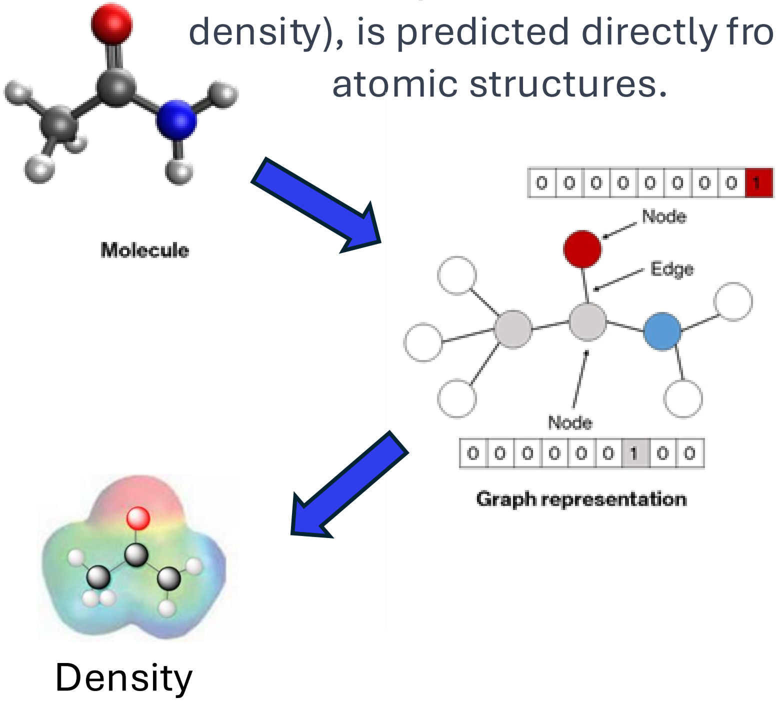
In QC-ML, the usual approach is to predict properties directly from representations of molecules/materials, such as graphs:



Typical architectures include Nequip, MACE, TensorNet, SchNet, PaiNN, etc.

ML density learning

An alternative data-efficient ML-accelerated physics simulation approach can be taken where the underlying variable (electron density), is predicted directly from atomic structures.



Different approaches exist, but the main approaches can be grouped into probe-based (e.g. DeepDFT) and orbital-based (e.g. SCDP)

Floating orbitals and the ELECTRA model

Traditionally, charge densities are represented with atom-centered spherical harmonics-based basis functions.

For ELECTRA we make a simplified Ansatz: a mixture of Gaussians $\rho(\mathbf{r}) = \sum_{A \in M} \sum_{j=0}^{N_A} w_{A,j} \mathcal{N}(\boldsymbol{r} | \boldsymbol{\mu}_{A,j}, \boldsymbol{\Sigma}_{A,j})$

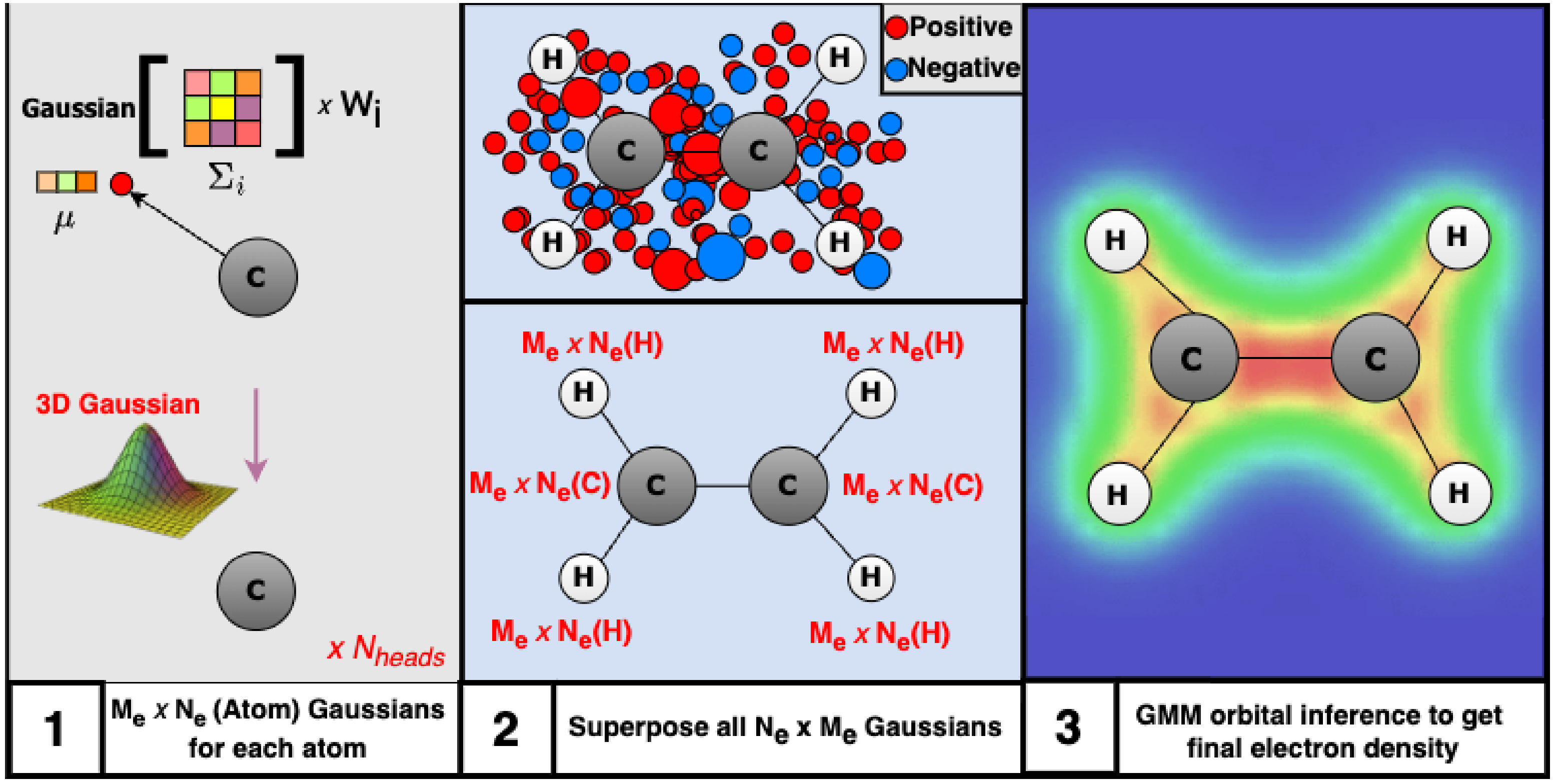
For the ansatz to be expressive enough, the Gaussians have to be ”floating” freely in space.

Why Gaussians? *They are very efficient to evaluate – see Gaussian splatting*.

Well-placed floating orbitals can represent densities more efficiently, using lower maximal angular quantum numbers l_{max}
ELECTRA is the first model to predict floating orbital positions using ML.

Floating orbitals and the ELECTRA model

Backbone is HotPP – an equivariant Cartesian GNN.
It outputs:
Rank 0 (scalars)
Rank 1 (vectors)
Rank 2 (matrices)



Gaussian
Σ_i
x W_i
μ
C
3D Gaussian
C
x N_{heads}
1
M_e x N_e (Atom) Gaussians for each atom
● Positive
● Negative
H
C
M_e x N_e(H)
M_e x N_e(C)
2
Superpose all N_e x M_e Gaussians
3
GMM orbital inference to get final electron density

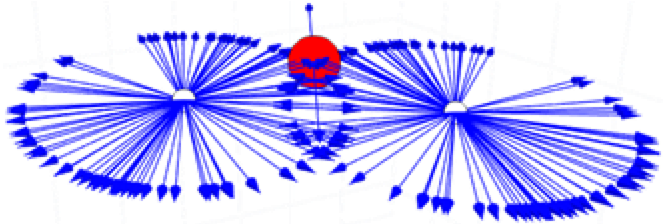
+ ***theoretical guarantee:*** There exists a universal approximation theorem for Gaussians which show that a model of N Gaussians can approximate a sufficiently smooth density to an arbitrary accuracy

Symmetry-breaking and debiasing

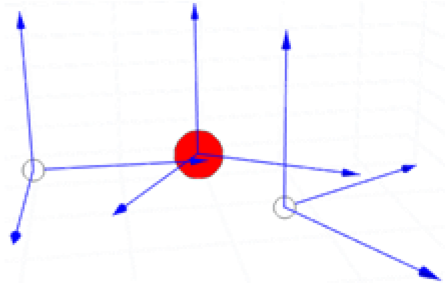
To construct an expressive method for placing floating orbitals, the model must allow for output vectors that belong to a lower symmetry group than the input structures. Thus, a symmetry-breaking mechanism is needed.

Symmetry-breaking

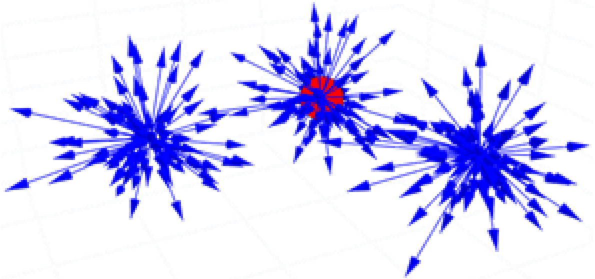
Without symmetry-breaking, the positions of Gaussians cannot be predicted with high expressivity, and end up in symmetry planes after message-passing



We design an equivariant symmetry-breaking mechanism that enhances expressivity, allowing Gaussians to be placed freely, by initializing the *l=1* features with moment-of-inertia eigenvectors



Initial MOI eigenvectors

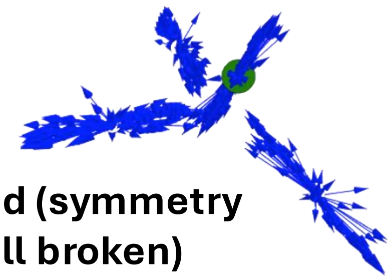


After message-passing, symmetry is now broken

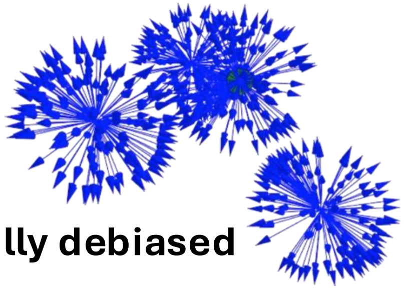
Debiasing

However, even with symmetry-breaking, GNNs can hang on to bias directions due to the way directional messages are constructed:

$$\begin{aligned} m_{ij,c}^{l=1} &= \underbrace{f_{l_r=0}(|r_{ij}|)h_{l_r=0,j,c}}_{\text{Scalars}=:s_c}\vec{r}_{ij} + \underbrace{f_{l_r=1}(|r_{ij}|)h_{l_r=1,j,c} + \text{Higher body terms}}_{\text{randomly initialized}} \\ &= s_c \qquad \underbrace{\vec{r}_{ij}}_{\text{Observed bias direction}} \qquad + \text{random directions}_c \end{aligned}$$



Biased (symmetry still broken)



Fully debiased

We thus also construct *debiasing layers* in our GNN model, which calculates bias directions using projections of the *l=1* features onto the dominant direction, and learns to debias by removing a learnable proportion of the bias direction from each vector

Results – QM9

ELECTRA uses floating orbitals and lower-order representations to achieve state-of-the-art density prediction accuracies while being an order of magnitude faster during inference.

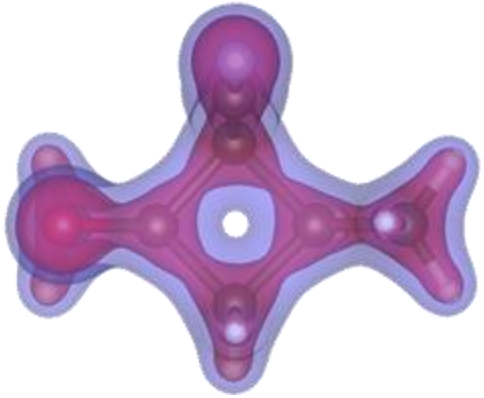
Metric	i-DeepDFT	e-DeepDFT	GPWNO	InfGCN[†]	InfGCN	ChargE3Net
NMAE [%] ↓	0.357	0.284	0.730	3.720	0.869	0.196
t_{inf} [s] ↓	–	–	–	–	0.833	15.18

Metric	ELECTRA (ours)	SCDP ($L = 3$)	SCDP+BO ($L = 6$)
NMAE [%] ↓	**0.176**	0.432	0.178
t_{inf} [s] ↓	**0.089**	0.395	1.022

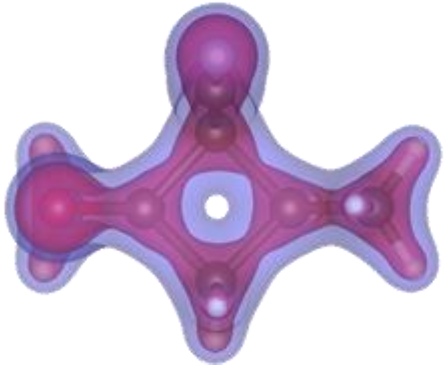
$$\mathrm{NMAE}\left(\rho_{\mathrm{pred}}, \rho_{\mathrm{ref}}\right) = \frac{\int_{\mathbb{R}^3} |\rho_{\mathrm{ref}}(\boldsymbol{r}) - \rho_{\mathrm{pred}}(\boldsymbol{r})| \mathrm{d}V}{\int_{\mathbb{R}^3} |\rho_{\mathrm{ref}}(\boldsymbol{r})| \mathrm{d}V}$$

+ Trained on 1x RTX3090 (24GB) vs 4x A100(80GB)

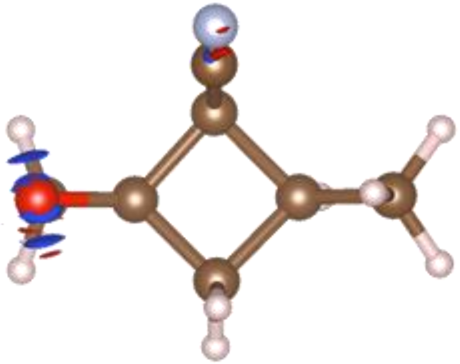
+ 51 % mean reduction in SCF cycles when used as initial guess (scales inv. with NMAE error)



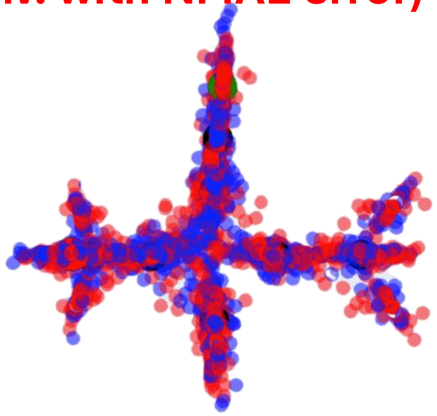
Ground truth



ELECTRA



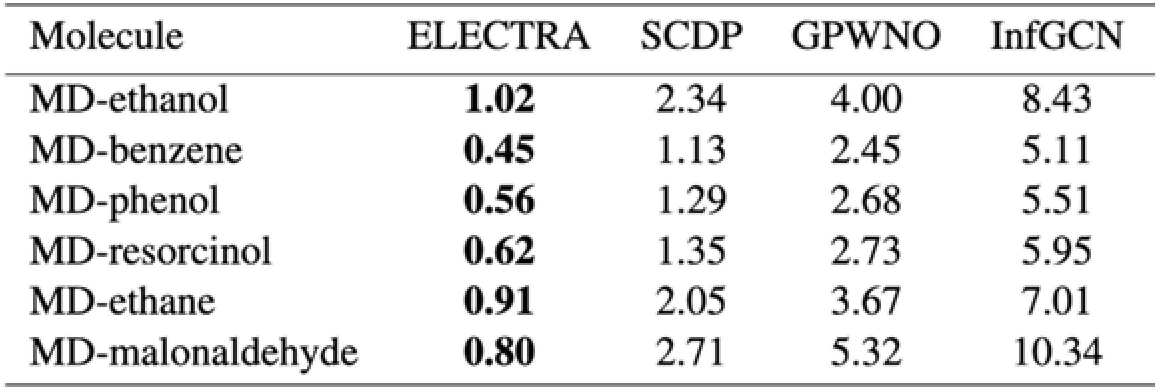
Errors



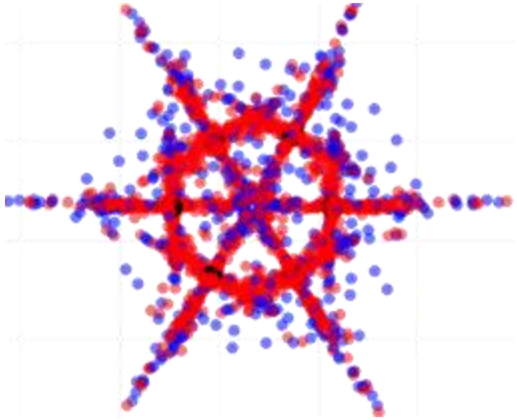
Gaussians

Results – MD17

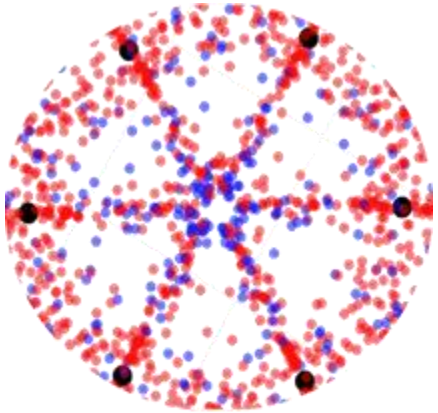
ELECTRA uses floating orbitals and lower-order representations to achieve state-of-the-art density prediction accuracies while being an order of magnitude faster during inference.



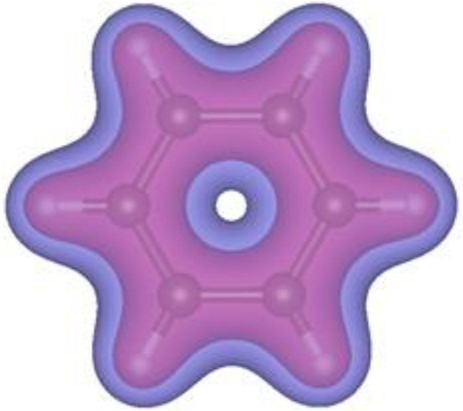
Molecule	ELECTRA	SCDP	GPWNO	InfGCN
MD-ethanol	**1.02**	2.34	4.00	8.43
MD-benzene	**0.45**	1.13	2.45	5.11
MD-phenol	**0.56**	1.29	2.68	5.51
MD-resorcinol	**0.62**	1.35	2.73	5.95
MD-ethane	**0.91**	2.05	3.67	7.01
MD-malonaldehyde	**0.80**	2.71	5.32	10.34



Gaussians



Gaussians (closeup)



Ground truth density