Recent Advances in Charge Density Prediction

Jiaxi Zhao (NUS)

June 6, 2025
Multiscale Modelling and Machine Learning Seminar

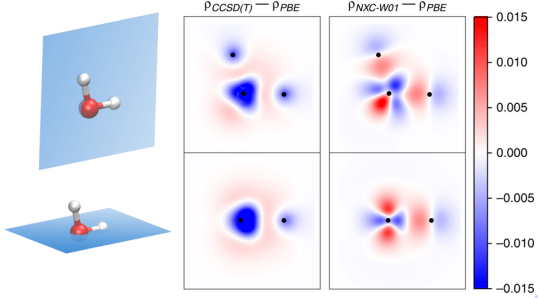
Outline

What is Charge Density?

- Charge density $\rho(\mathbf{r})$ represents the probability distribution of electrons in a system

$$\rho(\mathbf{r}) = N \int |\Psi(\mathbf{r}, \mathbf{r}_2, \cdots, \mathbf{r}_N)|^2 d\mathbf{r}_2 \cdots d\mathbf{r}_N, \quad \rho(\mathbf{r}) = \sum_{i=1}^{N} |\phi_i(\mathbf{r})|^2,$$

  for Hartree–Fock and density functional theory (DFT) where orbitals are defined.
- Key properties:
  - Non-negative: $\rho(\mathbf{r}) \geq 0$.
  - Normalized: $\int \rho(\mathbf{r}) d\mathbf{r} = N$ (number of electrons).



Why is Charge Density Important?

- Fundamental quantity of the DFT, required for most downstream calculations:
  - Band structure, phonon properties, etc.
  - High-throughput screening
  - Property optimization & Structure prediction
- Contains more information than the energy and force prediction. For a system with N atoms.
  - Total energy + Forces: dimension $= 3N + 1$.
  - Charge density: dimension $= n^3$ (n is the plane-wave grid size) or the number of basis functions.
  - An analog: Text2Text v.s. Text2Image.

Challenges in Charge Density Prediction

Physical constraints:

- $\mathrm{E}(3) = \mathrm{O}(3) \oplus \mathbb{R}^3$ equivariance: given two systems $\{(\mathbf{r}_a, Z_a)\}$ and $\{(\mathbf{r}'_a, Z'_a)\}$ related by $\mathbf{r}'_a = \mathrm{R}\,\mathbf{r}_a + \mathbf{t}$, $Z'_a = Z_a$, the charge density should satisfy:

$$\rho(\mathbf{r}) = \rho'(\mathrm{R}\,\mathbf{r} + \mathbf{t}), \quad \mathrm{R} \in \mathrm{O}(3), \mathbf{t} \in \mathbb{R}^3.$$

  Much more difficult than the energy and force prediction, especially for the rotational equivariance. Equivariance on the level of operator.

- If $E = E(\{(\mathbf{r}_a, Z_a)\})$ is G-invariant, then the gradient of it $F = \nabla_{\mathbf{r}_a} E(\{(\mathbf{r}_a, Z_a)\})$ is G-equivariant.

Two network parametrization for equivariance

- (Basis-based) For molecular systems with basis functions, we require:

$$\rho(\mathbf{r}) = \sum_a \sum_{l=0}^{\infty} \sum_{m=-l}^{l} c_{alm} R_{al}(|\mathbf{r} - \mathbf{r}_a|) Y_{lm}(\widehat{\mathbf{r} - \mathbf{r}_a}), \quad r = |\mathbf{r}|, \quad \widehat{\mathbf{r}} = \frac{\mathbf{r}}{r}.$$

  The equivariance of the density can be expressed as the equivariance of the vector of coefficients c_{alm}.

$$\{(\mathbf{r}_a, Z_a)\} \to \{c_{alm}\}$$

- (Probe-based) For general systems, motivated by implicit neural representation, we require the following map to be invariant:

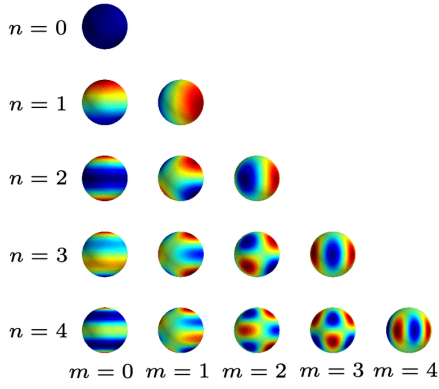
$$\{(\mathbf{r}_a, Z_a), \mathbf{r}'_1, ..., \mathbf{r}'_n\} \to \{\rho(\mathbf{r}'_1), \rho(\mathbf{r}'_2), ..., \rho(\mathbf{r}'_n)\}.$$

- (Mesh-based) Mesh representation of density can model the translational equivariance but not the rotational equivariance.

Representation of SO(3): spherical harmonics

- Spherical harmonics $Y_{lm}(\hat{\mathbf{r}})$ are eigenfunctions of angular momentum operators.
- $\{Y_{lm}\}_{m=-l}^{l}$ forms an irreducible representation of SO(3) of dimension $2l+1$.

$$Y_{lm}(\mathrm{R}\,\hat{\mathbf{r}}) = \sum_{m'=-l}^{l} D^{l}_{m'm}(\mathrm{R})\, Y_{lm'}(\hat{\mathbf{r}}) \Longrightarrow (\mathrm{R}\,c)_{lm} = \sum_{m'=-l}^{l} D^{l}_{m'm}(\mathrm{R})\, c_{lm'}.$$



Add equivariance into graph neural networks (GNN)

- A GNN is defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is permutation equivariant.
- A series of functions $f : \mathbb{R}^{|\mathcal{V}| \times F_i} \to \mathbb{R}^{|\mathcal{V}| \times F_{i+1}}$, locally aggregates all the features belonging to the neighborhood. For example, a GCN layer:

$$f(X) = \sigma(D^{-1/2}\widetilde{A}D^{-1/2}XW), \quad A, D \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}, X \in \mathbb{R}^{|\mathcal{V}| \times F_i}, W \in \mathbb{R}^{F_i \times F_{i+1}}, \widetilde{A} = A + I,$$

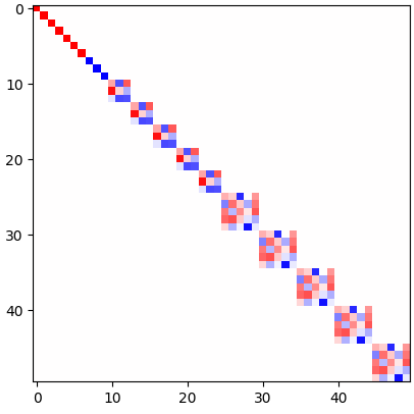
where W is the learnable weight matrix.

To enforce the equivariance $\mathrm{NN}(gX) = g\mathrm{NN}(X), g \in G$:

- The set of features X should have a SO(3)-action: construct X via spherical harmonics.
- The function f should be equivariant for each layer: special attention on the architecture design.

Build an equivariant GNN: irreps as features

- For each graph node, the feature vector is the direct sum of all the irreps of dimension less than or equal to L.
- Invariant scalar corresponds to $l = 0$; vector corresponds to $l = 1$; below is the transformation matrix of a rotation on reps 10x0o + 5x01o + 5x02o.



Fully Connected Tensor Product Operation

- Mathematical definition: reps = reps1 $\otimes$ reps2.

$$(U^{(l_1)} \otimes V^{(l_2)})^{(l)}_{cm} = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} C^{(l,m)}_{(l_1,m_1)(l_2,m_2)} U^{(l_1)}_{cm_1} V^{(l_2)}_{cm_2},$$

where $|l_1 - l_2| \leq l \leq |l_1 + l_2|$, $C^{(l,m)}_{(l_1,m_1)(l_2,m_2)}$ are Clebsch-Gordan coefficients

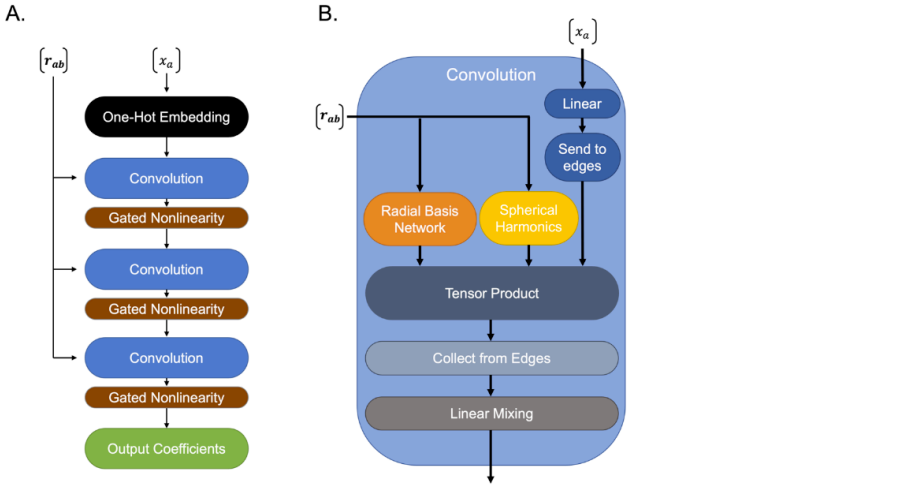
- **Equivariant convolution combines features of different angular momenta**:

$$\text{Conv}(\mathbf{r}_a, V^c_{alm}) = \sum_{m_1,m_2} C^{(l,m)}_{(l_1,m_1)(l_2,m_2)} \sum_{b \in \partial a} V^c_{bl_1m_1} R(r_{ab}) Y_{l_2m_2}(\hat{\mathbf{r}}_{ab}),$$

$$R(r_{ab}) = W_n \sigma(\cdots \sigma(W_2 \sigma(W_1 B(r_{ij})))), \quad B(r_{ab}) = \frac{2}{r_c} \frac{\sin\left(\frac{n\pi}{r_c} r_{ij}\right)}{r_{ij}} f(r_{ij}, r_c).$$

- Equivariance follows from the definition of the tensor product of the reps and the second reps depends both on the parameters and $\mathbf{r}$.

Equivariant graph convolution



Other equivariant operations

- Gated nonlinearity:

$$\sigma_{\text{Gated}}(X) = \sigma(X_{l=0}) \oplus (\sigma(X_{l=0}) \cdot X_{l\neq 0}).$$

- Self-interaction (similar to 1×1 convolution)

$$X^{\text{out}}_{c_1 l} = \sum_{c_2} W_{c_1 c_2} X^{\text{in}}_{jl}, \quad c_1 \in [F_{\text{out}}], c_2 \in [F_{\text{in}}], l \in [(L+1)^2].$$

- Point-wise spherical non-linearity.
- Selfmix layers (similar to self-attention), tensor product with itself.
- Pairmix layers.

Literature

- Thomas, Nathaniel, et al. "Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds." arXiv preprint arXiv:1802.08219 (2018).
- Batzner, Simon, et al. "E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials." Nature communications 13.1 (2022): 2453.
- Passaro, Saro, and C. Lawrence Zitnick. "Reducing SO (3) convolutions to SO (2) for efficient equivariant GNNs." International conference on machine learning. PMLR, 2023.

Dataset

- QM9: 134k small organic molecules, on average 18 atoms and 666K grid points for charge density voxels[1].
- MP: 123K crystal structures with charge density[2].

[4]Ramakrishnan, Raghunathan, et al. "Quantum chemistry structures and properties of 134 kilo molecules." Scientific data 1.1 (2014): 1-7.

[5]Shen, Jimmy-Xuan, et al. "A representation-independent electronic charge density database for crystalline materials." Scientific data 9.1 (2022): 661.

E3NN: Density fitting

- A consistent basis for all the charge density, not the case for DFT.
- **Density fitting**: fit the density to a predefined basis set, def2-universal-JFIT[3].

$$\rho(\mathbf{r}) = \sum_b C_b \phi_b^{\text{basis}}(\mathbf{r}),$$

The objective function is to minimize: $\mathcal{L} = \frac{1}{N} \sum_i \left(C_b - \widehat{C_b} \right)^2$.

- 3 hidden layers.
- Input reps = 2x0e (2 channels l=0) with hydrogen ([1, 0]) and oxygen ([0, 1]).
- Hidden reps = 125x0o + 125x0e + 40x1o + 40x1e + 25x2o + 25x2e + 15x3o + 15x3e.
- Output reps = 7x0e + 4x1e + 2x2e + 1x3e.

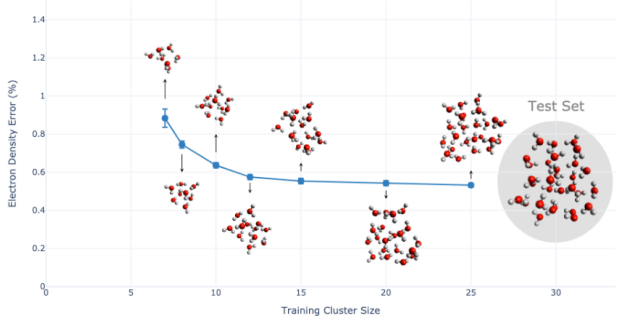
[6] Rackers, Joshua A., et al. "Cracking the quantum scaling limit with machine learned electron densities." arXiv preprint arXiv:2201.03726 (2022).

The effect of cluster size on density prediction

This is one of the first lines of work:

- Does not train on large dataset for density prediction.
- Training on specific systems to push the frontier of quantum chemistry calculations.

The y-axis is the relative L^1-error.



SCDP Architecture Overview

- Basis-based approach with following ingredients:
  - Virtual nodes for non-local electronic structures
  - Even-tempered Gaussian basis sets
  - High-capacity equivariant spherical channel network (eSCN).
- Charge density representation with trainable scale parameters:

$$\rho(\mathbf{r}) = \sum_{a}^{N_a} \sum_{l}^{l} \sum_{m=-l}^{l} c_{alm} \Phi_{\alpha,l,m,\mathbf{r}_a}(\mathbf{r}, s_{al})$$

$$\Phi_{\alpha,l,m,\mathbf{r}_i}(\mathbf{r}, s) = z_{\alpha,l,s} \exp\left(-s \cdot \alpha |\mathbf{r} - \mathbf{r}_a|^2\right) |\mathbf{r} - \mathbf{r}_a|^l Y_{lm}(\widehat{\mathbf{r} - \mathbf{r}_a})$$

- Model prediction, scaling factors $s_{al} \in (0.5, 2.0)$:

$$\{c_{alm}, s_{al}\} = F(\{(\mathbf{r}_a, Z_a)\}).$$

Virtual Nodes and Basis Sets

- Virtual nodes:
  - Placed at bond midpoints.
  - Use oxygen basis functions
  - SE(3)-equivariant placement
- Even-tempered Gaussian basis for better accuracy:

$$\alpha_k = \alpha \cdot \beta^k \text{ for } k = 0, 1, 2, ..., N_l$$

- Reducing SO(3) convolution to SO(2): $O(L^6) \rightarrow O(L^3)$.
- The basis set is not orthonormal, the coefficients depend on the cutoff radius.
- Two-stage training for stability caused by the scale factors on the exponent:
  - pre-train the model with fixed basis set exponents
  - fine-tune the prediction model with a small learning rate with the learning for scaling factors enabled.

Model Architecture

- Backbone: eSCN (equivariant spherical channel network)
  - $\{x_a\} = \text{eSCN}(\{(\mathbf{r}_a, Z_a)\})$
  - Complexity: $O(L^3)$ vs $O(L^6)$ for tensor products
  - Features: Multi-channel spherical harmonics
  - Example: $128 \times 0e + 128 \times 1o + 128 \times 2e + 128 \times 3o$
- Prediction layers:

$$\{c_{alm}, h_i\} = \text{FullyConnectedTensorProduct}(x_i, x_l)$$

$$s_{al} = \frac{C_1}{1 + \exp(-\text{Linear}(h_i) + \ln C_2)} + C_3 \in [C_1, C_3].$$

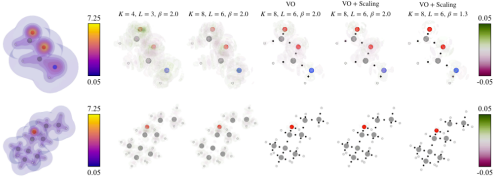
- Training objective:

$$\mathcal{L} = \mathbb{E}_{\mathbf{r} \in \text{Data}}[|\rho(\mathbf{r}) - \hat{\rho}(\mathbf{r})|]$$

Performance Analysis

- NMAE: 0.178% on QM9 test set
- 31.7x faster than ChargE3Net
- Flexible accuracy-efficiency trade-off

	NMAE [%] ↓	NMAE, Split 2 [%] ↓	Mol. per min. [min^{-1}] ↑
i-DeepDFT [7]	0.357 ± 0.001	-	-
e-DeepDFT [7]	0.284 ± 0.001	-	-
ChargE3Net [15]	0.196 ± 0.001	0.203 ± 0.003	3.95
InfGCN [12]	0.869 ± 0.002	0.93	72.00
InfGCN, GTO only [12]	-	3.72	-
GPWNO [22]	-	0.73	-
SCDP models (Ours)			
eSCN, $K = 4, L = 3, \beta = 2.0$	0.504 ± 0.001	0.514 ± 0.003	675.47
eSCN, $K = 8, L = 6, \beta = 2.0$	0.434 ± 0.006	0.452 ± 0.017	567.19
eSCN, $K = 8, L = 6, \beta = 1.5$	0.381 ± 0.001	0.391 ± 0.002	442.25
eSCN + VO, $K = 8, L = 6, \beta = 2.0$	0.237 ± 0.001	0.250 ± 0.002	231.21
eSCN + VO, $K = 8, L = 6, \beta = 1.5$	0.206 ± 0.001	0.220 ± 0.002	177.14
eSCN + VO, $K = 8, L = 6, \beta = 1.3$	0.196 ± 0.001	0.209 ± 0.002	136.92
SCDP models fine-tuned with scaling factors (Ours)			
eSCN, $K = 4, L = 3, \beta = 2.0$	0.432 ± 0.001	0.438 ± 0.003	644.00
eSCN, $K = 8, L = 6, \beta = 2.0$	0.369 ± 0.007	0.386 ± 0.018	544.56
eSCN, $K = 8, L = 6, \beta = 1.5$	0.346 ± 0.001	0.354 ± 0.002	419.57
eSCN + VO, $K = 8, L = 6, \beta = 2.0$	0.207 ± 0.001	0.220 ± 0.002	221.19
eSCN + VO, $K = 8, L = 6, \beta = 1.5$	0.187 ± 0.001	0.200 ± 0.002	164.94
eSCN + VO, $K = 8, L = 6, \beta = 1.3$	$\mathbf{0.178 \pm 0.001}$	$\mathbf{0.191 \pm 0.002}$	125.29



ChargE3Net Architecture Overview

- Based on the E3NN backbone.
- Construct two k-d trees to partition the atoms and probes.
- Two types of convolutions:
  - $\text{Conv}_{\text{atom}}$: Bidirectional between atoms

$$\text{Conv}^n_{\text{atom}}(\mathbf{r}_i, X^n_i) = W^n_1\Big(\sum_{j\in\partial(i)} W^n_2 X^n_j \otimes R(r_{ij})Y(\mathbf{r}_{ij})\Big) + W^n_3 X^n_i.$$

  - $\text{Conv}_{\text{probe}}$: From atoms to probes, only contains neighboring atoms, no probe-probe interactions.
- Training objective:

$$\mathcal{L} = \frac{\sum_{\mathbf{r}\in G}|\rho(\mathbf{r}) - \hat{\rho}(\mathbf{r})|}{\sum_{\mathbf{r}\in G}|\rho(\mathbf{r})|}$$

Model performance

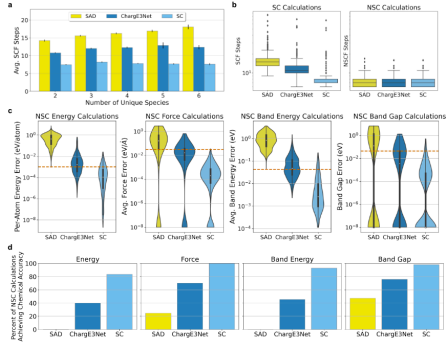
- Vertices:
  - Atoms: One-hot encoding of atomic number; reps = Nx0o.
  - Probe points: Initialized as zero scalar; reps = 1x0o.
- Edges:
  - Atom-atom: Unidirectional, cutoff 4Å
  - Atom-probe: Directed from atoms to probes
- Periodic boundary conditions supported

Dataset	invDeepDFT	equiDeepDFT	OrbNet-Equi	ChargE3Net
NMC	0.089 ± 0.001	$\mathbf{0.061} \pm 0.001$	-	$\mathbf{0.060} \pm 0.001$
QM9	0.357 ± 0.001	0.284 ± 0.001	0.206 ± 0.001	$\mathbf{0.196} \pm 0.001$
MP	0.859 ± 0.011	0.799 ± 0.010	-	$\mathbf{0.523} \pm 0.010$

All values are presented as an average ϵ_{mae} (%) and $\pm$ one standard error. Bold values represent the best ϵ_{mae} (%) scores. ϵ_{mae} reported for MP include both magnetic and non-magnetic materials.

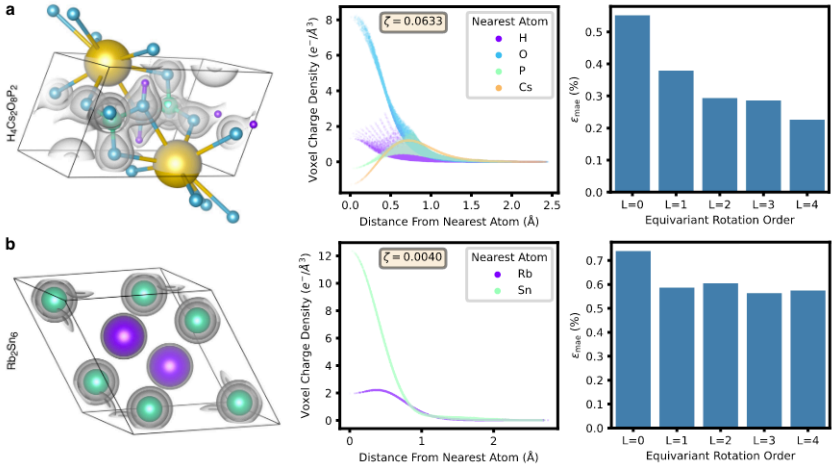
Performance

- Accelerate DFT calculations: MP materials: 26.7% reduction; GNoME materials: 28.6% reduction.
- Non-self-consistent property prediction:
  - 40% of materials: energy errors < 1 meV/atom
  - 70% of materials: forces < 0.03 eV/Å
  - 76% of materials: band gaps within chemical accuracy
- Linear scaling O(N) with system size.



Effect of higher-order features

- The total dimension of features is N.
- With highest order L, The dimension of each order is $N/(L+1)$.
- The channel of order l is $N/(L+1)/(2l+1)$.



Angular Variance Analysis

- Performance improvement:
  - 44.6% median improvement for materials with non-metals/metalloids
  - 23.0% median improvement for materials with only metals
- Metric ζ for angular variance:

$$\zeta(G) = 1 - \frac{\sum_{\vec{g}_k \in G} |\nabla \rho(\vec{g}_k) \cdot \hat{r}_{ki}|}{\sum_{\vec{g}_k \in G} ||\nabla \rho(\vec{g}_k)||}$$

- High angular variance materials, e.g. Cs(H2PO4)
  - Strong covalent bonding
  - Significant L=4 improvement
- Low angular variance materials, e.g. Rb2Sn6:
  - Primarily ionic interactions
  - Similar L=0 and L=4 performance

Discussions

- Incorperate more symmetries: space-group symmetry, point-group symmetry of crystal.
- Comparison with approaches of canonical transformations. This type of method is is restricted to be probe-based.
- Different approaches to equivariance in robot community: equivariant vision.
- **Sobolev training for downstream tasks where the gradient information of the density is required.**
- Does it worth to pursue the equivariance for the model? Seems not necessary as the AlphaFold evolves.