

CORMORANT: COVARIANT MOLECULAR NEURAL NETWORKS

Brandon M. Anderson^{*‡}, Truong Son Hy^{*}, Risi Kondor^{*†‡}

^{*}Department of Computer Science, [†] Department of Statistics, The University of Chicago

[‡]Center for Computational Mathematics, Flatiron Institute

[‡]Atomwise



Abstract

We propose *Cormorant*, a rotationally covariant neural network architecture for learning the behavior and properties of complex many-body physical systems. We apply these networks to molecular systems with two goals: learning atomic potential energy surfaces for use in Molecular Dynamics simulations, and learning ground state properties of molecules calculated by Density Functional Theory. Some of the key features of our network are that (a) each neuron explicitly corresponds to a subset of atoms; (b) the activation of each neuron is covariant to rotations, ensuring that overall the network is fully rotationally invariant. Furthermore, the non-linearity in our network is based upon tensor products and the Clebsch-Gordan decomposition, allowing the network to operate entirely in Fourier space. *Cormorant* significantly outperforms competing algorithms in learning molecular Potential Energy Surfaces from conformational geometries in the MD-17 dataset, and is competitive with other methods at learning geometric, energetic, electronic, and thermodynamic properties of molecules on the GDB-9 dataset.

Learning on Molecules

We want to learn on molecular data specified by a set of charge-position pairs (Z_i, \mathbf{r}_i) for each atom.

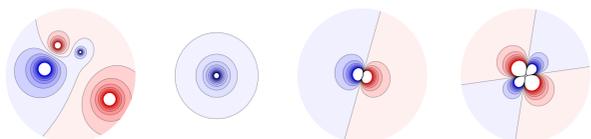


This problem is invariant to rotations and translations. We use covariant activations to “bake-in” these symmetries, while retaining local geometric information.

The multipole expansion

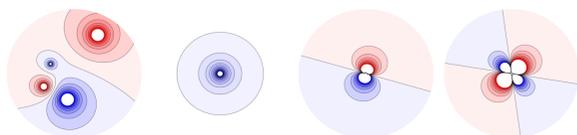
Let’s take inspiration from the multipole expansion in physics. Expanding the potential $V(\mathbf{r})$ around a point \mathbf{r} gives:

$$\sum_i \frac{Z_i}{|\mathbf{r}-\mathbf{r}_i|} = \text{monopole: } Q_0 Y^0(\hat{\mathbf{r}})/r + \text{dipole: } Q_1 Y^1(\hat{\mathbf{r}})/r^2 + \text{quadrupole: } Q_2 Y^2(\hat{\mathbf{r}})/r^3 + \dots$$



Here Q_ℓ is the ℓ -th multipole moment, and $Y^\ell(\hat{\mathbf{r}})$ is a spherical harmonic.

Consider the effects of a 90° CCW-rotation on the input charges and corresponding moments.



All moments rotate **covariantly**. More precisely, given a rotation $R \in \text{SO}(3)$, each moment rotates as $Q_\ell \rightarrow D^\ell(R) \cdot Q_\ell$, where $D^\ell: \text{SO}(3) \rightarrow \mathbb{C}^{(2\ell+1) \times (2\ell+1)}$ is a Wigner- D matrix or irreducible representation (irrep) of the group $\text{SO}(3)$ labeled by positive integer $\ell = 0, 1, 2, \dots$

Spherical tensors and representation theory

Physical quantities transform under rotation R :

$$q \mapsto q \quad \mu \mapsto R\mu \quad \Theta \mapsto R\Theta R^T \quad \mathbf{r}_{AB} \mapsto R\mathbf{r}_{AB}$$

In general, a k ’th order Cartesian moment tensor $T^{(k)} \in \mathbb{R}^{3 \times 3 \times \dots \times 3}$ (or its flattened $\bar{T}^{(k)} \in \mathbb{R}^{3^k}$) transforms as:

$$\bar{T}^{(k)} \mapsto (R \otimes R \otimes \dots \otimes R) \bar{T}^{(k)}$$

In the specific case of $\text{SO}(3)$, there is a fixed unitary transformation matrix $C^{(k)}$ which reduces the k ’th order rotation operator into a direct sum of irreps:

$$\underbrace{R \otimes R \otimes \dots \otimes R}_k = C^{(k)} \left[\bigoplus_{\ell} \bigoplus_{i=1}^{\tau_\ell} D^\ell(R) \right] C^{(k)\dagger}$$

The vectorized form of the Cartesian moment tensor $\bar{T}^{(k)}$ has a corresponding decomposition:

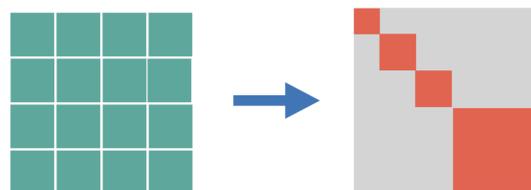
$$\bar{T}^{(k)} = C^{(k)} \left[\bigoplus_{\ell} \bigoplus_{i=1}^{\tau_\ell} Q_{\ell,i} \right]$$

Under rotations, the individual $Q_{\ell,i}$ components transform independently as $Q_{\ell,i} \mapsto D^\ell(R) Q_{\ell,i}$. For example, in the dipole/dipole case, we need terms of the form $Q_{\ell_1}^A \otimes Q_{\ell_2}^B$ that transform according to the tensor product of the corresponding irreps:

$$Q_{\ell_1}^A \otimes Q_{\ell_2}^B \mapsto (D^{\ell_1}(R) \otimes D^{\ell_2}(R)) (Q_{\ell_1}^A \otimes Q_{\ell_2}^B)$$

in which $D^{\ell_1}(R) \otimes D^{\ell_2}(R)$ is not an irreducible representation, but can be decomposed into irreducibles by **Clebsch - Gordan** decomposition:

$$D^{\ell_1}(R) \otimes D^{\ell_2}(R) = C_{\ell_1, \ell_2}^\dagger \left[\bigoplus_{\ell=|\ell_1-\ell_2|}^{\ell_1+\ell_2} D^\ell(R) \right] C_{\ell_1, \ell_2}$$



Covariant molecular neural network

Definition 1. We say that F is an $\text{SO}(3)$ - covariant vector of type $\tau = (\tau_0, \tau_1, \tau_2, \dots, \tau_L)$ if it can be written as a collection of complex matrices F_0, F_1, \dots, F_L , called its isotypic parts, where each F_ℓ is a matrix of size $(2\ell+1) \times \tau_\ell$ and transforms under rotations as $F_\ell \mapsto D^\ell(R) F_\ell$.

Definition 2. Let S be a molecule or other physical system consisting of N atoms. A **Cormorant** covariant molecular neural network for S is a feed forward neural network consisting of m neurons $\mathbf{n}_1, \dots, \mathbf{n}_m$ such that:

- Every neuron \mathbf{n}_i corresponds to some subset S_i of the atoms. In particular, each input neuron corresponds to a single atom. Each output neuron corresponds to the entire system S .
- The activation of each \mathbf{n}_i is an $\text{SO}(3)$ - vector of a fixed type τ_i .
- The type of each output neuron is $\tau_{out} = (1)$, i.e., a scalar.

Clebsch-Gordan Layers

Cormorant is based upon the Clebsch-Gordan layer. We use covariant $\text{SO}(3)$ -vectors $F_i^s = \bigoplus_{\ell=0}^L \bigoplus_{\ell} [F_{\ell, c, i}^s]$ as activations for each atom i at layer s . The Clebsch-Gordan decomposition is the central operation in our network:

$$[F_{\ell_1} \otimes_{\text{CG}} G_{\ell_2}]_{\ell, *; i} = \bigoplus_{\ell=\ell_1-\ell_2}^{\ell_1+\ell_2} C_{\ell_1, \ell_2, \ell} [F_{\ell_1}]_{*; i} \otimes [G_{\ell_2}]_{*; i}.$$

Using this operation, we update our activations in a two-step process. Atom activations are updated using:

$$F_i^s = \left[\underbrace{F_i^{s-1} \oplus (F_i^{s-1} \otimes_{\text{CG}} F_i^{s-1})}_{\text{one-body part}} \oplus \left(\underbrace{\sum_j G_{i,j}^{s,\ell} \otimes_{\text{CG}} F_j^{s-1}}_j \right) \right] \cdot W_{s,\ell}^{\text{vertex}},$$

where $W_{s,\ell}^{\text{vertex}}$ is a set of learnable weights. Note that the form of the CG non-linearity is enforced by group theory. Any other covariant non-linearity must be either composed of CG operations, or scalars.

The $\text{SO}(3)$ -vector edge activations [4] $G_{i,j}^{s,\ell} = g_{i,j}^{s,\ell} Y^\ell(\hat{\mathbf{r}}_{ij})$ are constructed from the spherical harmonics $Y^\ell(\hat{\mathbf{r}}_{ij})$ of the relative position vector $\mathbf{r}_{i,j} = \mathbf{r}_i - \mathbf{r}_j$ between atoms i and j , along with the scalar edge network $g_{i,j}^{s,\ell}$. Specifically,

$$g_{i,j}^{s,\ell} = \mu^s(r_{i,j}) \left[\left(g_{i,j}^{s-1,\ell} \oplus (F_i^{s-1} \cdot F_j^{s-1}) \oplus \eta^{s,\ell}(r_{i,j}) \right) \cdot W_{s,\ell}^{\text{edge}} \right],$$

where $\mu^s(r_{i,j})$ is a learnable mask, $\eta^{s,\ell}(r_{i,j})$ is a set of learnable radial basis functions of the relative distance $r_{i,j}$, and $W_{s,\ell}^{\text{edge}}$ are learnable weights.

Experiments

We present experimental results on two datasets of interest to the computational chemistry community: QM-9 [1] for learning the ground state properties of a set of molecules, and MD-17 [2] for learning molecular force fields and potential energy surfaces.

Our code is available at <https://github.com/risilab/cormorant>.

Table 1. GDB-9 results

	Cormorant	SchNet [3]	NMP [4]	WaveScatt [5]
α (bohr ³)	0.085	0.235	0.092	0.160
$\Delta\epsilon$ (eV)	0.061	0.063	0.069	0.118
ϵ_{HOMO} (eV)	0.034	0.041	0.043	0.085
ϵ_{LUMO} (eV)	0.038	0.034	0.038	0.076
μ (D)	0.038	0.033	0.030	0.340
C_v (cal/mol K)	0.026	0.033	0.040	0.049
G (eV)	0.020	0.014	0.019	0.022
H (eV)	0.021	0.014	0.017	0.022
R^2 (bohr ²)	0.961	0.073	0.180	0.410
U (eV)	0.021	0.019	0.020	0.022
U_0 (eV)	0.022	0.014	0.020	0.022
ZPVE (meV)	2.027	1.700	1.500	2.000

Table 2. MD-17 results

	Cormorant	DeepMD [6]	DTNN [7]	SchNet [3]	GDML [2]	GDML [8]
Aspirin	0.098	0.201	0.120	0.270	0.190	0.190
Benzene	0.023	0.065	0.040	0.070	0.070	0.100
Ethanol	0.027	0.055	0.050	0.150	0.150	0.070
Malonaldehyde	0.041	0.095	0.190	0.080	0.160	0.100
Naphthalene	0.029	0.095	–	0.110	0.120	0.120
Salicylic Acid	0.066	0.106	0.410	0.100	0.120	0.120
Toluene	0.034	0.085	0.180	0.080	0.120	0.100
Uracil	0.023	0.085	–	0.100	0.110	0.110

- [1] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld. Scientific Data, 1, 140022 (2014).
 [2] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller. Sci. Adv. 3, e1603015 (2017)
 [3] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. J. Chem. Phys. 148, 241722 (2018)
 [4] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and H. E. Dahl. PMLR 70, 1263, (2017).
 [5] M. Hirn, S. Mallat, and N. Poilvert. Multiscale Modeling Simulation, 15, 827 (2017).
 [6] L. Zhang, J. Han, H. Wang, R. Car, and W. E. Phys. Rev. Lett., 120, 143001 (2018).
 [7] K. T. Schütt, F. Arbabzadah, S. Chmiela, K.-R. Müller, and A. Tkatchenko. Nat. Comm. 8, 13890 (2017).
 [8] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko. Nat. Comm., 9, 3887 (2018).
 [9] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, P. Riley, arXiv:1802.08219