# Covariant Compositional Neural Networks For Learning Graphs

Truong Son Hy†, Shubhendu Trivedi‡, Horace Pan†, Brandon M. Anderson†, Risi Kondor†

†Department of Computer Science, University of Chicago ‡Toyota Technological Institute at Chicago

## Abstract

We propose Covariant Compositional Networks (CCNs), a novel neural network architecture for learning on graphs. CCNs use tensor representations for vertex features which can then be manipulated with permutation covariant tensor operations as opposed to the standard symmetric operations used in other graph neural network models. These permutation covariant operations allow us to build more expressive graph representations while still maintaining permutation invariance.

For learning small-scale molecular graphs, we investigate the efficacy of CCNs in estimating Density Functional Theory (DFT), a widely used but expensive approach to compute the electronic structure of matter. We obtain promising results in for this task and outperform other graph learning models on the Harvard Clean Energy Project (**HCEP**) and QM9 (**QM9**) molecular datasets.

## Compositional Scheme

Let $\mathcal{G}$ be an object with $n$ elementary parts (atoms) $\mathcal{E} = \{e_1, .., e_n\}$. A **compositional scheme** for $\mathcal{G}$ is a directed acyclic graph (DAG) $\mathcal{M}$ in which each node $\nu$ is associated with some subset $\mathcal{P}_\nu$ of $\mathcal{E}$ (these subsets are called **parts** of $\mathcal{G}$) in such a way that:

1. In the bottom level, there are exactly $n$ leaf nodes in which each leaf node $\nu$ is associated with an elementary atom $e$. Then $\mathcal{P}_\nu$ contains a single atom $e$.
2. $\mathcal{M}$ has a unique root node $\nu_r$ that corresponds to the entire set $\{e_1, .., e_n\}$.
3. For any two nodes $\nu$ and $\nu'$, if $\nu$ is a descendant of $\nu'$, then $\mathcal{P}_\nu$ is a subset of $\mathcal{P}_{\nu'}$.

## Covariance

For a graph $G$ with the comp-net $\mathcal{N}$, and an isomorphic graph $G'$ with comp-net $\mathcal{N}'$, let $\nu$ be any neuron of $\mathcal{N}$ and $\nu'$ be the corresponding neuron of $\mathcal{N}'$. Assume that $\mathcal{P}_\nu = (e_{p_1}, .., e_{p_m})$ while $\mathcal{P}_{\nu'} = (e_{q_1}, .., e_{q_m})$, and let $\pi \in \mathbb{S}_m$ be the permutation that aligns the orderings of the two receptive fields, i.e., for which $e_{q_{\pi(a)}} = e_{p_a}$. We say that $\mathcal{N}$ is **covariant to permutations** if for any $\pi$, there is a corresponding function $R_\pi$ such that $f_{\nu'} = R_\pi(f_\nu)$.

## First-order Message Passing

We call standard message passing **zero'th order message passing** where each vertex is represented by a feature vector of length $c$ (or $c$ channels). When we sum together vertex features of this form, we lose identity information on where certain vertex features originated from. Hence, we propose **first order message passing** by instead representing each vertex $v$ by a matrix: $f_v^\ell \in \mathbb{R}^{|\mathcal{P}_v^\ell| \times c}$. Each row of this feature matrix corresponds to a vertex in the neighborhood of $v$.

We say that $\nu$ is a **first order covariant node** in a comp-net if under the permutation of its receptive field $\mathcal{P}_\nu$ by any $\pi \in \mathbb{S}_{|\mathcal{P}_\nu|}$, its activation transforms as $f_\nu \mapsto P_\pi f_\nu$, where $P_\pi$ is the permutation matrix:

$$[P_\pi]_{i,j} \triangleq \begin{cases} 1, & \pi(j) = i \\ 0, & otherwise \end{cases} \tag{1}$$

The transformed activation $f_{\nu'}$ will be: $[f_{\nu'}]_{a,s} = [f_\nu]_{\pi^{-1}(a),s}$ where $s$ is the channel.

## Second-order Message Passing

Instead of representing a vertex with a feature matrix (a 2nd order tensor) as done in first order message passing, we can represent it by a 3rd order tensor $f_v^\ell \in \mathbb{R}^{|\mathcal{P}_v^\ell| \times |\mathcal{P}_v^\ell| \times c}$ and require these feature tensors to transform covariantly.

We say that $\nu$ is a **second order covariant node** in a comp-net if under the permutation of its receptive field $\mathcal{P}_\nu$ by an $\pi \in \mathbb{S}_{|\mathcal{P}_\nu|}$, its activation transforms as $f_\nu \mapsto P_\pi f_\nu P_\pi^T$. The transformed activation $f_{\nu'}$ will be:

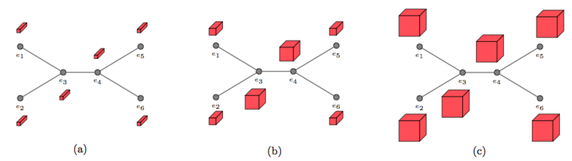
$$[f_{\nu'}]_{a,b,s} = [f_\nu]_{\pi^{-1}(a),\pi^{-1}(b),s}$$

where $s$ is the channel index.

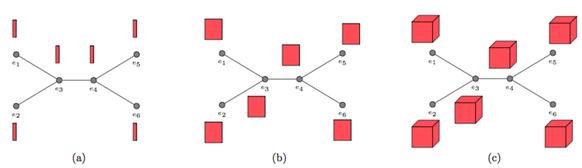
## Algorithm

**Input:** $G$, $l_\nu$, $L$
**Parameters:** Matrices $W_0 \in \mathbb{R}^{c \times c}$, $W_1, .., W_L \in \mathbb{R}^{(18c) \times c}$ and biases $b_0, .., b_L$.
$F_\nu^0 \leftarrow \Upsilon(W_0 l_\nu + b_0 1)$ $(\forall \nu \in V)$
Reshape $F_\nu^0$ to $1 \times 1 \times c$ $(\forall \nu \in V)$
**for** $\ell = 1, .., L$ **do**
  **for** $\nu \in V$ **do**
    $F_{w \to \nu}^\ell \leftarrow \chi \times F_w^{\ell-1} \times \chi^T$ where $\chi = \chi_{w \to \nu}^\ell$ $(\forall w \in \mathcal{P}_\nu^\ell)$
    Apply virtual tensor contraction algorithm with inputs $\{F_{w \to \nu}^\ell | w \in \mathcal{P}_\nu^\ell\}$ and the restricted adjacency matrix $A \downarrow_{\mathcal{P}_\nu^\ell}$ to compute $\overline{F}_\nu^\ell \in \mathbb{R}^{|\mathcal{P}_\nu^\ell| \times |\mathcal{P}_\nu^\ell| \times (18c)}$.
    $F_\nu^\ell \leftarrow \Upsilon(\overline{F}_\nu^\ell \times W_\ell + b_\ell 1)$
  **end**
**end**
$F^\ell \leftarrow \sum_{\nu \in V} \Theta(F_\nu^\ell)$ $(\forall \ell)$

**Output:** Graph feature $F \leftarrow \bigoplus_{\ell=0}^{L} F^\ell \in \mathbb{R}^{(L+1)c}$. Use $F$ for downstream tasks.




Tensor activations for our CCN-2D architecture applied to a $C_2H_4$ molecular graph. The tensor activations of each vertex in a CCN 2D model are shown after 0, 1, and 2 rounds of message passing in (a), (b) and (c). Here the rows and columns correspond to the size of the receptive field, whereas the depth of the tensor is determined by the number of channels.




Geometry of the tensor activations in zeroth (CCN 0D), first (CCN 1D), and second (CCN 2D) order message passing algorithms. Vertices have a vector (zeroth order), matrix (first order), and second order tensor representations corresponding as shown in (a), (b), and (c).

## Result

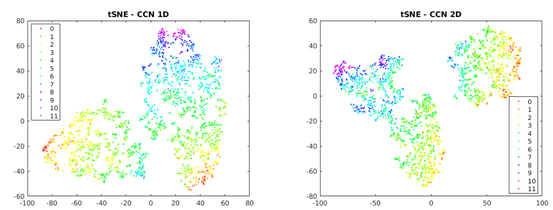
MAE and RMSE results of each model on predicting the Power Conversion Efficiency (PCE) for graphs on the test set of HCEP. Lower values are better.

| | Test MAE | Test RMSE |
|---|---|---|
| Lasso | 0.867 | 1.437 |
| Ridge regression | 0.854 | 1.376 |
| Random forest | 1.004 | 1.799 |
| Gradient boosted trees | 0.704 | 1.005 |
| WL graph kernel | 0.805 | 1.096 |
| Neural graph fingerprints | 0.851 | 1.177 |
| PSCN | 0.718 | 0.973 |
| CCN 1D | **0.216** | **0.291** |
| CCN 2D | **0.340** | **0.449** |

Regression results of CCN-1D architecture applied to QM9(b). A comparison between CCN prediction error and DFT error known as "chemical accuracy."

| Target | CCNs | DFT error | Physical unit |
|---|---|---|---|
| alpha | **0.19** | 0.4 | $\text{Bohr}^3$ |
| Cv | **0.06** | 0.34 | cal/mol/K |
| G | **0.05** | 0.1 | eV |
| gap | **0.11** | 1.2 | eV |
| H | **0.05** | 0.1 | eV |
| HOMO | **0.08** | 2.0 | eV |
| LUMO | **0.07** | 2.6 | eV |
| mu | 0.43 | **0.1** | Debye |
| omega1 | **2.54** | 28 | $\text{cm}^{-1}$ |
| R2 | 5.03 | - | $\text{Bohr}^2$ |
| U | **0.06** | 0.1 | eV |
| U0 | **0.05** | 0.1 | eV |
| ZPVE | **0.0043** | 0.0097 | eV |

2D t-SNE visualization of learned CCNs molecular features on HCEP dataset:




## Reference

T. S. Hy and S. Trivedi and H. Pan and B. M. Anderson and R. Kondor. 2018. Predicting molecular properties with covariant compositional networks. *Journal of Chemical Physics*. 148 (2018). Issue 24.

## Contact

School: hytruongson@uchicago.edu
Internship: hytruongson@google.com