Research Meeting - June 2, 2021

Graph Representation Learning & Deep Generative Models On Graphs

Truong Son Hy *

*Department of Computer Science The University of Chicago

Ryerson Physical Lab



Content

Content:

- Graph representation learning:
 - Message passing neural networks
 - Permutation equivariance
 - Covariant compositional networks
- Deep generative models on graphs:
 - Variational Autoencoder
 - Equivariant graph/molecule generation





Message passing neural networks

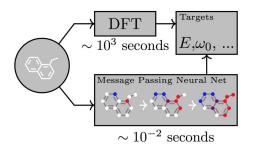
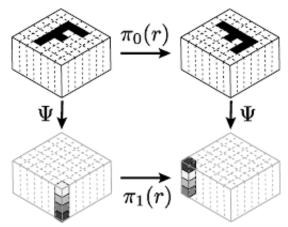


Figure 1. A Message Passing Neural Network predicts quantum properties of an organic molecule by modeling a computationally expensive DFT calculation.

Neural Message Passing for Quantum Chemistry, ICML 2017



Equivariance



Steerable CNNs, Taco S. Cohen, Max Welling, https://arxiv.org/abs/1612.08498



Permutation equivariance on graphs (1)

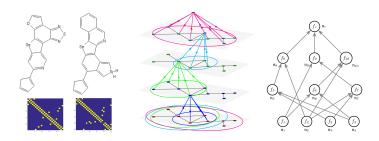
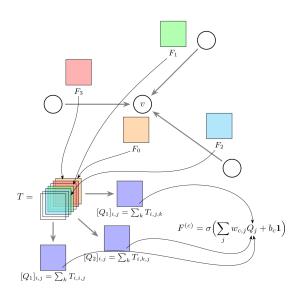


FIG. 1. Left: Molecular graphs for C₁₈H₉N₃OSSe and C₂₂H₁₅NSeSi from the Harvard Clean Enery Project (HCEP)¹⁰ dataset with corresponding adjacency matrices. Center and right: The comp-net of a graph G is constructed by decomposing G into a hierarchy of subgraphs {P_t} and forming a neural network N in which each "neuron" norresponds to one of the P_t subgraphs, and receives inputs from other neurons that correspond to smaller subgraphs contained in P_t. The center pane shows how this can equivalently be thought of as an algorithm in which each vertex of G receives and aggregates messages from its neighbors. To keep the figure simple, we only marked aggregation at a single vertex in each round (layer).

Predicting molecular properties with covariant compositional networks, https://aip.scitation.org/doi/10.1063/1.5024797

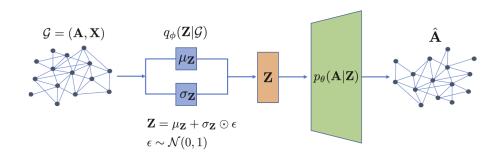
Permutation equivariance on graphs (2)







Variational Autoencoder



Graph Representation Learning, William L. Hamilton (McGill University 2020) https://www.cs.mcgill.ca/~wlh/grl_book/

Equivariant molecule generation

Generated examples (model trained on ZINC):

Interpolation on the latent:

