Question: Can we model long-range interactions with a single jump in molecular graphs? 🤔

ICLR

**Neural Atoms: Propagating Long-Range Interaction In Molecular Graphs through Efficient Communication Channel**

Xuan Li*, Zhanke Zhou*, Jiangchao Yao, Yu Rong, Lu Zhang, Bo Han

TMLR · TRUSTWORTHY MACHINE LEARNING AND REASONING · 香港浸會大學 HONG KONG BAPTIST UNIVERSITY · 上海交通大学 SHANGHAI JIAO TONG UNIVERSITY · Tencent AI Lab

Paper   Code   Slide

Background: Long-range Interactions

Molecular graphs consist of different types of atom interaction with different properties and functions.

○ H ● O ● N ● C
○—○ SRI ○ - - - ○ LRI
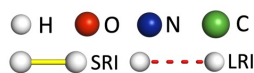
- The short-range interaction (SRI) forms the structure of the molecular graph.

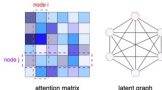- The long-range interaction (LRI) could determine physical and chemical properties.

(E)-N-(2-methylstyryl)-N-allylacetamide
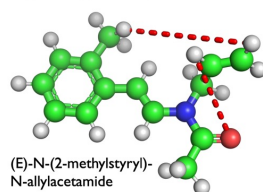
Limitations of existing approaches

Graph Neural Network

Stacking multiple GNN layers to capture LRI?
- **Over-smoothing:** representations become indistinguishable;
- **Over-squashing:** overwhelming information be squashed.

Graph Transformer

Using a fully connected graph to capture LRI?
- **Irrelevant interactions:** the LRIs are naturally sparse;
- **Additional computation:** unnecessary node-pair attentions.

Motivation

Modeling **distant interactions** with **a single jump** in molecular graphs in **another space.**

Advantages

- **Learnable projection** from atoms to neural atoms;
- **Reducing** the multi-hop long-range interaction to single-hop;
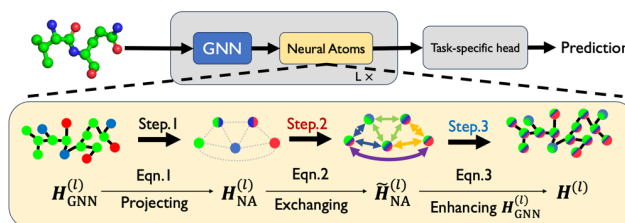- GNN-agnostic and **plug-in-and-play.**
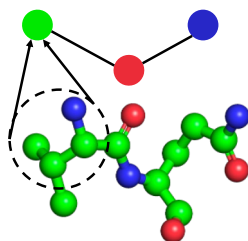
Neural Atoms

Definition 1. *Neural atoms encompass a collection of virtual, parameterized atoms that symbolize a cluster of atoms within a designated molecular graph. The process entails the acquisition of knowledge that enables the transformation of conventional atoms into neural atoms, along with their interactions. This transformation can be technically executed with model-agnostic methodologies.*

7-hops

Mapping f

Long-range communication channels

1-hop

Original Atoms

Retrieving f^{-1}
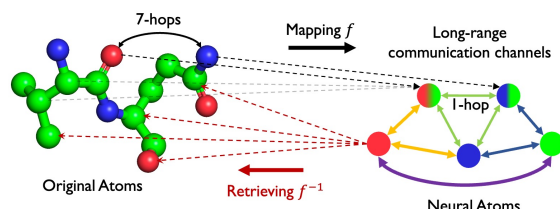
Neural Atoms

Learning to **project** all the original atoms into a few **neural atoms** that **abstract** the collective information of atomic groups in a molecule.

Implementation

GNN → Neural Atoms → Task-specific head → Prediction
L ×

Step.1   Step.2   Step.3

$H_{\text{GNN}}^{(l)}$ → Eqn.1 Projecting → $H_{\text{NA}}^{(l)}$ → Eqn.2 Exchanging → $\tilde{H}_{\text{NA}}^{(l)}$ → Eqn.3 Enhancing $H_{\text{GNN}}^{(l)}$ → $H^{(l)}$

Step-1. Project atom representations $H_{GNN}^{(\ell)}$ to neural atom representations $H_{NA}^{(\ell)}$.

$$H_{\text{NA}}^{(\ell)} = \text{LayerNorm}\left(Q_{\text{NA}}^{(\ell)} \oplus \text{MultiHead}(Q_{\text{NA}}^{(\ell)}, H_{\text{GNN}}^{(\ell)}, H_{\text{GNN}}^{(\ell)})\right)$$

Step-2. Exchange information among neural atoms $H_{NA}^{(\ell)} \mapsto \tilde{H}_{NA}^{(\ell)}$.

$$\tilde{H}_{\text{NA}}^{(\ell)} = \text{LayerNorm}\left(H_{\text{NA}}^{(\ell)} \oplus \text{MultiHead}(H_{\text{NA}}^{(\ell)}, H_{\text{NA}}^{(\ell)}, H_{\text{NA}}^{(\ell)})\right)$$

Step-3. Project neural atoms back and **enhance** the atoms' representation $(H_{GNN}^{(\ell)}, \tilde{H}_{NA}^{(\ell)}) \mapsto H^{(\ell)}$.

$$H^{(\ell)} = H_{\text{GNN}}^{(\ell)} \oplus \tilde{A}_{\text{NA}}^{(\ell)} \tilde{H}_{\text{NA}}^{(\ell)}, \text{ s.t. } \tilde{A}_{\text{NA}}^{(\ell)} = \text{Aggregate}\left(\{\hat{A}_m\}_{m=1}^{M}\right)^{\top} \in \mathbb{R}^{N \times K}$$

Empirical Study

→ Neural atoms can **boost** the performance of various GNNs up to **27.32%.**

Table 1: Test performance on three LRGB datasets. Shown is the mean ± s.d. of 4 runs.

Model	Peptides-func	Peptides-struct	PCQM-Contact
	AP ↑	MAE ↓	MRR ↑
Transformer+LapPE	0.6326 ± 0.0126	0.2529 ± 0.0016	0.3174 ± 0.0020
SAN+LapPE	0.6384 ± 0.0121	0.2683 ± 0.0043	0.3350 ± 0.0003
GraphGPS	0.6535 ± 0.0041	0.2500 ± 0.0005	0.3337 ± 0.0006
GCN	0.5930 ± 0.0023	0.3496 ± 0.0013	0.2329 ± 0.0009
+ Neural Atoms	**0.6220 ± 0.0046**	**0.2606 ± 0.0027**	**0.2534 ± 0.0200**
GINE	0.5498 ± 0.0079	0.3547 ± 0.0045	0.3180 ± 0.0027
+ Neural Atoms	**0.6154 ± 0.0157**	**0.2553 ± 0.0005**	**0.3126 ± 0.0021**
GCNII	0.5543 ± 0.0078	0.3471 ± 0.0010	0.3161 ± 0.0004
+ Neural Atoms	**0.5996 ± 0.0033**	**0.2563 ± 0.0020**	**0.3049 ± 0.0006**
GatedGCN	0.5864 ± 0.0077	0.3420 ± 0.0013	0.3218 ± 0.0011
+ Neural Atoms	**0.6562 ± 0.0075**	**0.2585 ± 0.0017**	**0.3258 ± 0.0003**
GatedGCN+RWSE	0.6069 ± 0.0035	0.3357 ± 0.0006	0.3242 ± 0.0008
+ Neural Atoms	**0.6591 ± 0.0050**	**0.2568 ± 0.0005**	**0.3262 ± 0.0010**

→ Neural atoms **without 3D information** and **half the #params.** achieve competitiveness or outperform the Ewald-based approach (previous SOTA).

Table 2: Validation energy MAE and MSE comparison on OE62 dataset.

	Energy MAE ↓	Energy MSE ↓	Number of Params.
SchNet (Schütt et al., 2017)	0.1351	0.0658	2.75 M
+ Ewald Block	**0.0811**	**0.0301**	12.21 M
+ Neural Atoms	0.0834	0.0309	2.63 M
PaiNN (Schütt et al., 2021)	0.6049	0.0133	12.52 M
+ Ewald Block	0.0590	0.0134	15.68 M
+ Neural Atoms	0.0558	0.0122	6.05 M
DimeNet++ (Gasteiger et al., 2020)	0.0501	0.0117	2.76 M
+ Ewald Block	**0.0479**	**0.0107**	4.75 M
+ Neural Atoms	0.0551	0.0129	1.97 M

→ Neural Atoms are **better** than multiple fully connected Virtual Nodes.

Table 14: Performance for virtual nodes (VNs) and neural atoms (NAs) in Peptide-Func, evaluated by AP (the higher, the better).

Model	Method	#VNs /#NAs = 5	#VNs /#NAs = 15	#VNs /#NAs = 75	#VNs /#NAs = 135
GCN	VNs	0.5566	0.5543	0.5568	0.5588
	NAs	**0.5962**	**0.5859**	**0.5903**	**0.6220**
GINE	VNs	0.5437	0.5500	0.5426	0.5426
	NAs	**0.6107**	**0.6128**	**0.6147**	**0.6154**
GCNII	VNs	0.5086	0.5106	0.5077	0.5083
	NAs	**0.6061**	**0.5862**	**0.5909**	**0.5996**
GatedGCN	VNs	0.5810	0.5868	0.5761	0.5810
	NAs	**0.6660**	**0.6533**	**0.6562**	**0.6562**

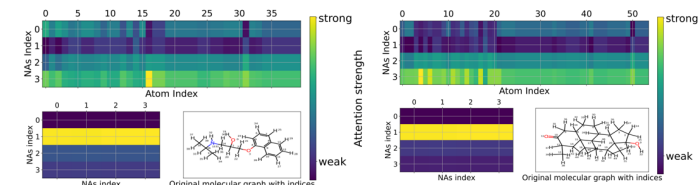→ Neural Atoms have **sparse** attention and establish **inter-communications.**

Figure 17: Mutagenicity test set index-18

Figure 19: Mutagenicity test set index-29