Optimizing over trained GNNs via symmetry breaking

Shiqiang Zhang[a], Juan S Campos[a], Christian Feldmann[b], David Walz[b], Frederik Sandfort[b], Miriam Mathea[b], Calvin Tsay[a], Ruth Misener[a]

[a] Imperial College London, South Kensington, SW7 2AZ, UK
[b] BASF SE, Ludwigshafen am Rhein, Germany

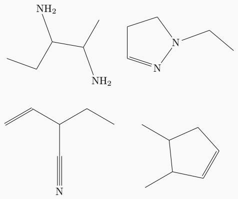
Funding EPSRC EP/W003317/1 & EP/T001577/1

37th Conference on Neural Information Processing Systems
New Orleans, United States
December 10th - 16th, 2023

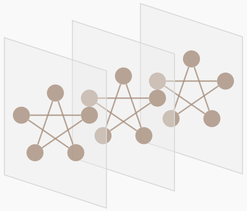
Motivation: Forward & Backward problems over GNNs

Prediction (Forward): What are the properties for a given molecule?

molecules



GNN



properties

quantum mechanics
physico-chemical
bioactivity
(eco)-toxicity
. . .

Optimization (Backward): What is the optimal molecule with desired properties?

Problem definition

Given a trained GNN, we aim to find the input with optimal property [1]:

$$\begin{aligned}(X^*, A^*) = \underset{(X,A)}{\arg\min}\ & GNN(X, A) \\ s.t.\ & f_j(X, A) \le 0, j \in \mathcal{J} \\ & g_k(X, A) = 0, k \in \mathcal{K}\end{aligned}$$

where X denotes features, A is the adjacency matrix of input graph, f_j, g_k are problem-specific constraints, and $\mathcal{J}, K$ are index sets.

[1]Optimality is defined on this given GNN instead of true properties.

Symmetry issue

Observation

GNN is permutation invariant[2]: isomorphic graphs have the same output.

Good for training

Different indexing of a graph data will not influence its output.

[2] In this work, we only consider GNNs that are permutation invariant.

Symmetry issue

Observation

GNN is permutation invariant[2]: isomorphic graphs have the same output.

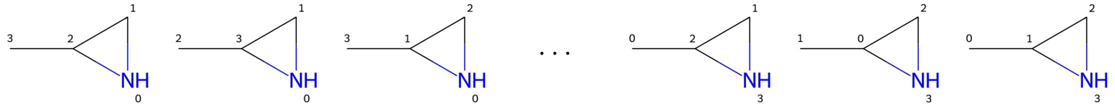
Good for training

Different indexing of a graph data will not influence its output.

Bad for optimization

Each graph indexing corresponds to a solution, which significantly enlarges the searching space.

For example, there are $4! = 24$ different indexing for this molecule:



[2] In this work, we only consider GNNs that are permutation invariant.

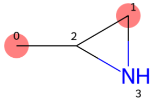
Symmetry-breaking constraints I[3]

Each node (except 0) should be linked with a node with smaller index:

$$\forall v \in [N] \backslash \{0\},\ \exists u < v,\ s.t.\ A_{u,v} = 1 \tag{S1}$$

i.e., the subgraph induced by nodes $\{0, 1, \dots, v\}$ is connected.

10 out of 24 solutions violate (S1), for example:



$\Longrightarrow$ Node 1 is not linked with node 0.

[3] N: number of nodes.

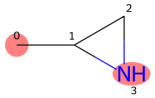
Symmetry-breaking constraints II[4]

Node 0 has the minimal function value under a designed hierarchical function $h : \mathbb{R}^F \to \mathbb{R}$ defined over features:

$$h(X_0) \le h(X_v),\ \forall v \in [N]\backslash\{0\} \tag{S2}$$

i.e., node 0 has the most "special" features under the action of h.

11 out of 14 solutions violate (S2), for example:



$\Longrightarrow$ Construct h such that $h(N) < h(C)$, then the nitrogen atom should be indexed 0.

[4] F: number of features.

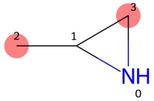
Symmetry-breaking constraints III[5]

The neighbor set of a node with smaller index has smaller lexicographical order:

$$LO(\mathcal{N}(v)\backslash\{v+1\}) \leq LO(\mathcal{N}(v+1)\backslash\{v\}),\ \forall v \in [N-1]\backslash\{0\} \tag{S3}$$

i.e., node v has "stronger" neighbors comparing to node $v+1$.

2 out of 3 solutions violate (S3), for example:



$$\Longrightarrow \quad \mathcal{N}(2) = \{1\}, \mathcal{N}(3) = \{0,1\}$$

$$LO(\mathcal{N}(2)\backslash\{3\}) > LO(\mathcal{N}(3)\backslash\{2\})$$

[5] $\mathcal{N}(\cdot)$: neighbor set. $LO(\cdot)$: lexicographical order.

Do these constraints reduce the diversity of the feasible set?

Algorithm 1 yields at least one feasible indexing for any graph (see proofs in the paper).

Algorithm 1 Indexing algorithm

```
Input: $G = (V, E)$ with node set $V = \{v_0, v_1, \dots, v_{N-1}\}$ ($N := |V|$). Denote the neighbor set of node $v$ as $\mathcal{N}(v), \forall v \in V$.
$\mathcal{I}(v_0) \leftarrow 0$                                         ▷ Assume that $v_0$ is indexed with $0$
$s \leftarrow 1$                                                        ▷ Index for next node
$V_1^1 \leftarrow \{v_0\}$                                              ▷ Initialize set of indexed nodes
while $s < N$ do
    $V_2^s \leftarrow V \backslash V_1^s$                               ▷ Set of unindexed nodes
    $\mathcal{N}^s(v) \leftarrow \{\mathcal{I}(u) \mid u \in \mathcal{N}(v) \cap V_1^s\}, \ \forall v \in V_2^s$    ▷ Obtain all indexed neighbors
    $rank^s(v) \leftarrow \left|\{LO(\mathcal{N}^s(u)) < LO(\mathcal{N}^s(v)) \mid \forall u \in V_2^s\}\right|, \ \forall v \in V_2^s$
                                                                        ▷ Assign a rank to each unindexed node
    $\mathcal{I}^s(v) \leftarrow \begin{cases} \mathcal{I}(v), & \forall v \in V_1^s \\ rank^s(v) + s, & \forall v \in V_2^s \end{cases}$    ▷ Assign temporary indexes
    $\mathcal{N}_t^s(v) \leftarrow \{\mathcal{I}^s(u) \mid u \in \mathcal{N}(v)\}, \ \forall v \in V_2^s$    ▷ Define temporary neighbor sets based on $\mathcal{I}^s$
    $v^s \leftarrow \arg\min_{v \in V_2^s} LO(\mathcal{N}_t^s(v))$      ▷ Neighbors of $v^s$ has minimal order
                                ▷ If multiple nodes share the same minimal order, arbitrarily choose one
    $\mathcal{I}(v^s) = s$                                              ▷ Index $s$ to node $v^s$
    $V_1^{s+1} \leftarrow V_1^s \cup \{v^s\}$                           ▷ Add $v^s$ to set of indexed nodes
    $s \leftarrow s + 1$                                                ▷ Next index is $s + 1$
end while
Output: $\mathcal{I}(v), v \in V$                                       ▷ Result indexing
```

Mixed-integer formulation for GNNs

Since the input graph structure is not fixed, all elements in the adjacency matrix are variables:

$$\boldsymbol{x}_v^{(l)} = \sigma\left(\sum_{u\in V} e_{u\to v}\boldsymbol{w}_{u\to v}^{(l)}\boldsymbol{x}_u^{(l-1)} + \boldsymbol{b}_v^{(l)}\right)$$

where

- $\boldsymbol{x}_v^{(l)}$: (continuous or discrete) **variables**, the features of node v in l-th layer.
- $e_{u\to v}$: binary **variable**, the existence of edge $u \to v$.
- $\boldsymbol{w}_{u\to v}^{(l)}, \boldsymbol{b}_v^{(l)}$: **constants**, weights and biases of l-th layer.

Bilinear terms $e_{u\to v}\boldsymbol{x}_u^{(l-1)}$ result in a mixed-integer quadratically constrained optimization problem (MIQCP), which can be handled by state-of-the-art solvers such as Gurobi.

Alternatively, they can be reformulated in a linear way using big-M formulation.

Numerical results

Optimal molecular design:

- atom $\rightarrow$ node, bond $\rightarrow$ edge
- atom type, #neighbors, ... $\rightarrow$ features
- chemical requirements $\rightarrow$ constraints

Our numerical results show that:

- Symmetry-breaking constraints significantly reduce the searching space.
- After breaking symmetry, the solving time is largely decreased.

Table 1: Numbers of feasible solutions for QM7.

N	(S1)	(S1) - (S2)	(S1) - (S3)
4	$3,323$	726	416
5	$67,020$	$11,747$	$3,003$
6	$\mathbf{>2,500,000}$	$\mathbf{443,757}$	$\mathbf{50,951}$
7	$>2,500,000$	$>2,500,000$	$504,952$

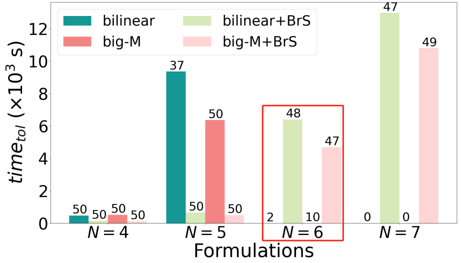


Figure 1: Average solving time over 50 runs.

Numerical results

Optimal molecular design:

- atom $\rightarrow$ node, bond $\rightarrow$ edge
- atom type, #neighbors, ... $\rightarrow$ features
- chemical requirements $\rightarrow$ constraints

Our numerical results show that:

- Symmetry-breaking constraints significantly reduce the searching space.
- After breaking symmetry, the solving time is largely decreased.

Thanks for your attention!

Table 1: Numbers of feasible solutions for QM7.

N	(S1)	(S1) - (S2)	(S1) - (S3)
4	$3,323$	726	416
5	$67,020$	$11,747$	$3,003$
6	$\mathbf{> 2,500,000}$	$\mathbf{443,757}$	$\mathbf{50,951}$
7	$> 2,500,000$	$> 2,500,000$	$504,952$

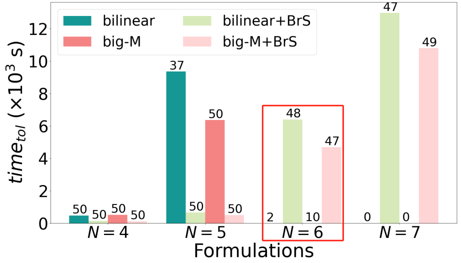


Figure 1: Average solving time over 50 runs.