Optimising (Graph) Neural Networks Using Exact Solvers

Tom Mc Donald, Calvin Tsay, Artur M. Schweidtmann, **Neil Yorke-Smith**
n.yorke-smith@tudelft.nl

Computers & Chemical Engineering
Volume 185, June 2024, 108660

Mixed-integer optimisation of graph neural networks for computer-aided molecular design

Tom McDonald [a d], Calvin Tsay [b], Artur M. Schweidtmann [c], Neil Yorke-Smith [d]

Show more

+ Add to Mendeley Share Cite

https://doi.org/10.1016/j.compchemeng.2024.108660 Get rights and content

Highlights

- Mixed-integer linear programming formulation of graph neural networks.
- Formulations for Graph Convolutional Network and GraphSAGE.
- Showcase that such models can learn from non-euclidean data structures efficiently.
- Sample application to molecule discovery with optimal boiling points.

Training NN with MIP

Fischetti and Jo
2018

Grimstad and Andersson
2019

Anderson et al
2020

Thorbjarnarson and
Yorke-Smith, 2021

Maragno et al
2021

Schweidtmann et al
2022

Can we formulate Graph Convolutional Networks as Mixed Integer (Non-) Linear Problems?

1

MINLP formulation of GCN

MILP formulation of GraphSAGE

2

genetic algorithm approach to optimise GNNs

3

experiments on molecular boiling point optimisation

1

MINLP formulation of GCN

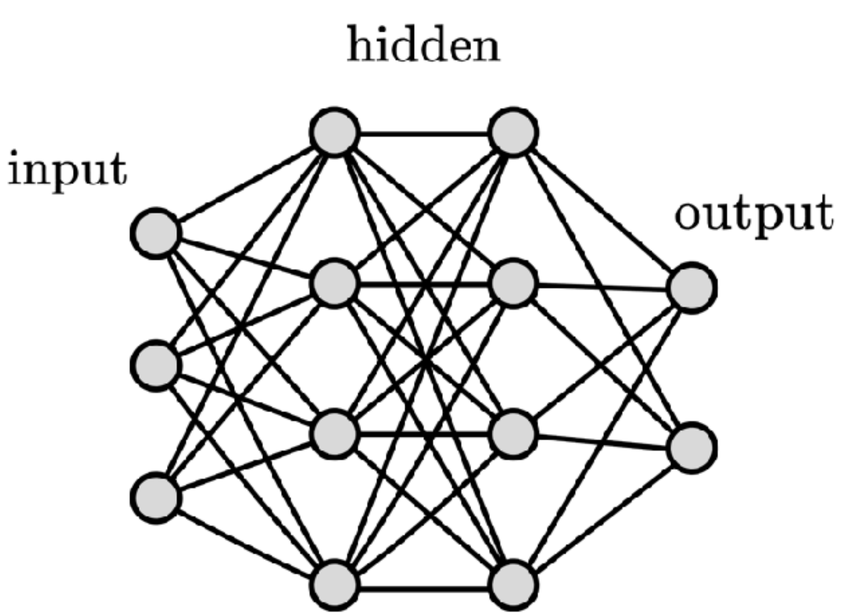
MILP formulation of GraphSAGE

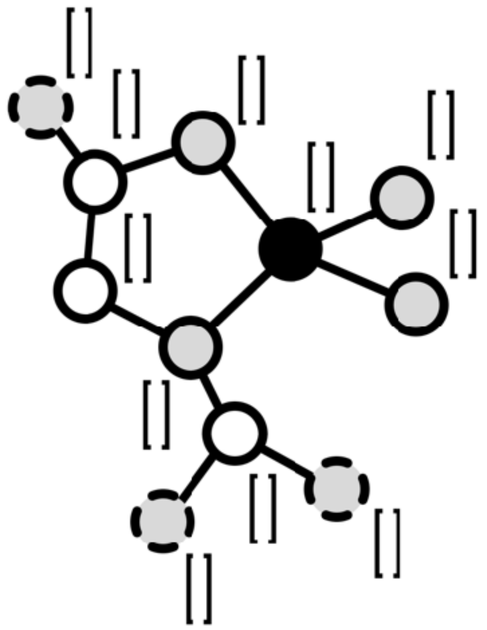
2

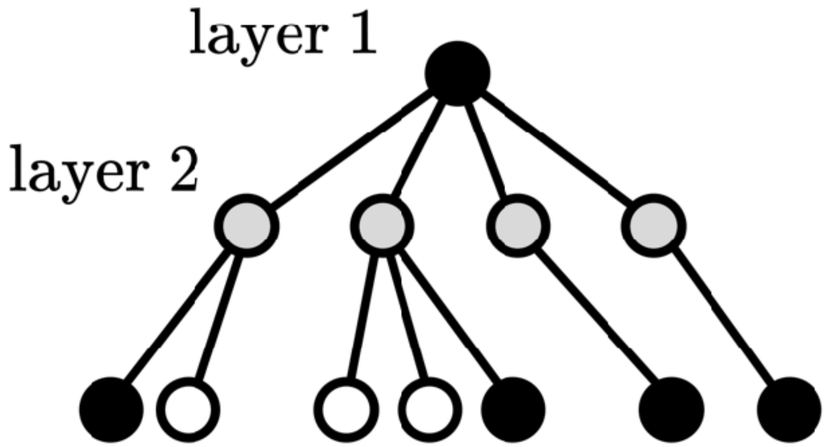
genetic algorithm approach to optimise GNNs

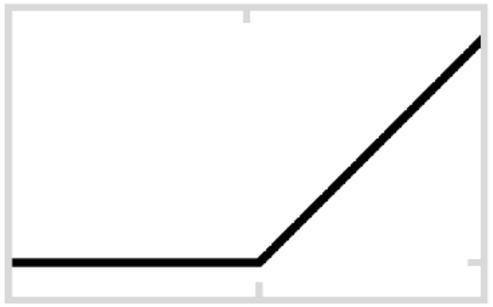
3

experiments on molecular boiling point optimisation



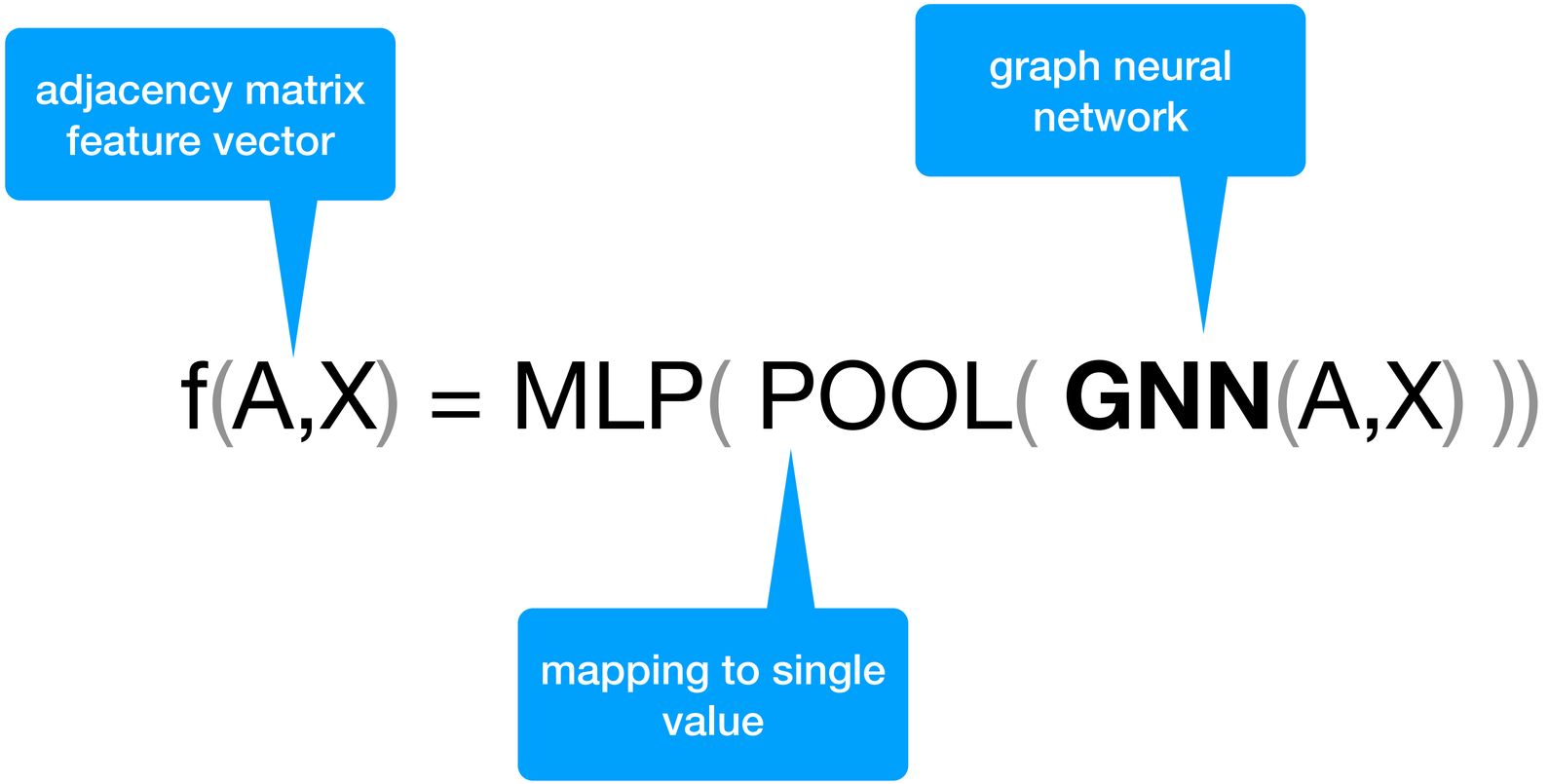






$$
\begin{aligned}
\text{maximise} \quad & x_1^K \\
\text{subject to} \quad & \boldsymbol{W}^K \boldsymbol{x}^{K-1} + b^K = x_1^K \\
& \boldsymbol{W}_j^k \boldsymbol{x}^{k-1} + b_j^k = x_j^k - s_j^k && \forall k \in \{1, \ldots, K-1\}, \forall j \in \{1, \ldots, n_k\} \\
& x_j^k \leq u_j^k z_j^k && \forall k \in \{1, \ldots, K-1\}, \forall j \in \{1, \ldots, n_k\} \\
& s_j^k \leq -l_j^k (1 - z_j^k) && \forall k \in \{1, \ldots, K-1\}, \forall j \in \{1, \ldots, n_k\} \\
& x_j^k, s_j^k \geq 0 && \forall k \in \{1, \ldots, K-1\}, \forall j \in \{1, \ldots, n_k\} \\
& z_j^k \in \{0, 1\} && \forall k \in \{1, \ldots, K-1\}, \forall j \in \{1, \ldots, n_k\} \\
& x^0 \in \Omega
\end{aligned}
$$

Matteo Fischetti and Jason Jo: Deep neural networks and mixed integer linear optimization. Constraints, 23(3):296–309, 2018



$$H^{(k+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(k)}W^{(k)}\right)$$

GCN structure

$$\sum_{l|\tilde{A}_{il}=1} \frac{1}{\sqrt{d_i^+ d_l^+}} \boldsymbol{W}_j^{k^T} \boldsymbol{H}_l^{(k-1)} = H_{ij}^k - S_{ij}^k$$

$$H_{ij}^k \leq U_{ij}^k Z_{ij}^k$$

$$S_{ij}^k \leq -L_{ij}^k (1 - Z_{ij}^k)$$

$$0 \leq H_{ij}^k, S_{ij}^k \in \mathbb{R}$$

$$Z_{ij}^k \in \{0, 1\}$$

$$A, H^0 \in \Omega$$

MINLP model

$$x_j^0 = \sum_{i=1}^{N} H_{ij}^K \quad \forall j \in \{1, \ldots, n_K\}$$

pooling

Unknown graph structure

$$\sum_{l|\tilde{A}_{il}=1} \frac{1}{\sqrt{d_i^+ d_l^+}} \boldsymbol{W}_j^{k\,T} \boldsymbol{H}_l^{(k-1)} = H_{ij}^k - S_{ij}^k$$

can make bilinear

add a boolean var

GUROBI OPTIMIZATION

$$H_{ij}^{(k)} = \sigma\left(\hat{\boldsymbol{W}}_j^{kT}\boldsymbol{H}_i^{(k-1)} + \bar{\boldsymbol{W}}_j^{kT}\sum_{l|A_{il}=1}\boldsymbol{H}_l^{(k-1)}\right)$$

GraphSAGE structure

$$(\hat{\boldsymbol{W}}_j^k)^T\boldsymbol{H}_i^{(k-1)} + (\bar{\boldsymbol{W}}_j^k)^T\sum_l \boldsymbol{b}_{il}^{(k-1)} = H_{ij}^k - S_{ij}^k$$

$$H_{ij}^k \leq U_{ij}^k Z_{ij}^k$$

$$S_{ij}^k \leq -L_{ij}^k(1 - Z_{ij}^k)$$

$$\boldsymbol{H}_l^k - \boldsymbol{M}(1 - A_{il}) \leq \boldsymbol{b}_{il}^k$$

$$\boldsymbol{b}_{il}^k \leq \boldsymbol{H}_l^k + \boldsymbol{M}(1 - A_{il})$$

$$-\boldsymbol{M}(A_{il}) \leq \boldsymbol{b}_{il}^k \leq \boldsymbol{M}(A_{il})$$

$$\boldsymbol{H}_i^0 = \boldsymbol{x}_i$$

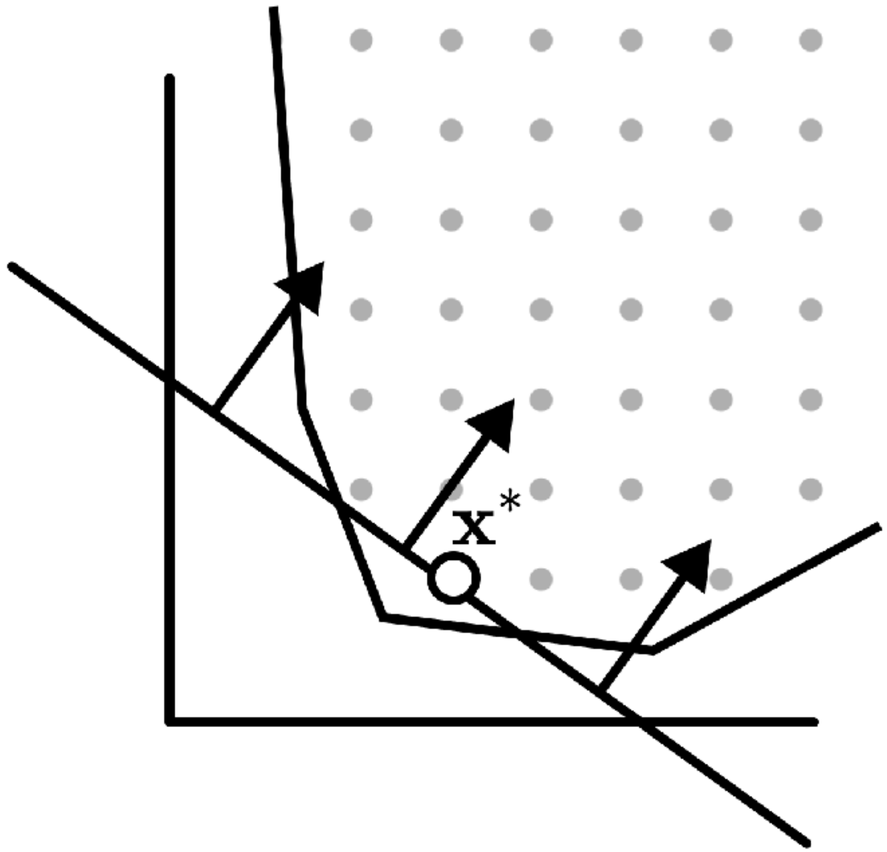
$$H_j^{*K} = \sum_i H_{ij}^K$$

$$0 \leq H_{ij}^k, S_{ij}^k \in \mathbb{R}$$

$$Z_{ij}^k \in \{0, 1\}$$

details in the paper :)

Helping the solver



Helping the solver

$$(\hat{\boldsymbol{W}}_j^k)^T \boldsymbol{H}_i^{(k-1)} + (\bar{\boldsymbol{W}}_j^k)^T \sum_l \boldsymbol{b}_{il}^{(k-1)} = H_{ij}^k - S_{ij}^k$$

$$H_{ij}^k \leq U_{ij}^k Z_{ij}^k$$

$$S_{ij}^k \leq -L_{ij}^k (1 - Z_{ij}^k)$$

$$\boldsymbol{H}_l^k - \boldsymbol{M}(1 - A_{il}) \leq \boldsymbol{b}_{il}^k$$

$$\boldsymbol{b}_{il}^k \leq \boldsymbol{H}_l^k + \boldsymbol{M}(1 - A_{il})$$

$$-\boldsymbol{M}(A_{il}) \leq \boldsymbol{b}_{il}^k \leq \boldsymbol{M}(A_{il})$$

$$\boldsymbol{H}_i^0 = \boldsymbol{x}_i$$

$$H_j^{*K} = \sum_i H_{ij}^K$$

$$0 \leq H_{ij}^k, S_{ij}^k \in \mathbb{R}$$

$$Z_{ij}^k \in \{0, 1\}$$

feasibility-based bounds tightening

domain-specific side constraints

1

MINLP formulation of GCN

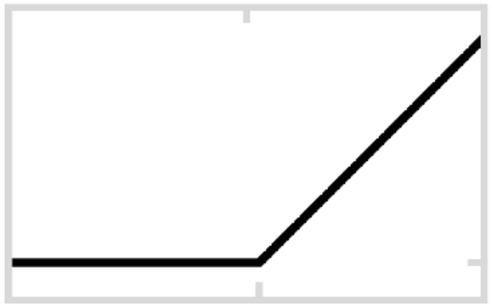
MILP formulation of GraphSAGE

2

genetic algorithm approach to optimise GNNs

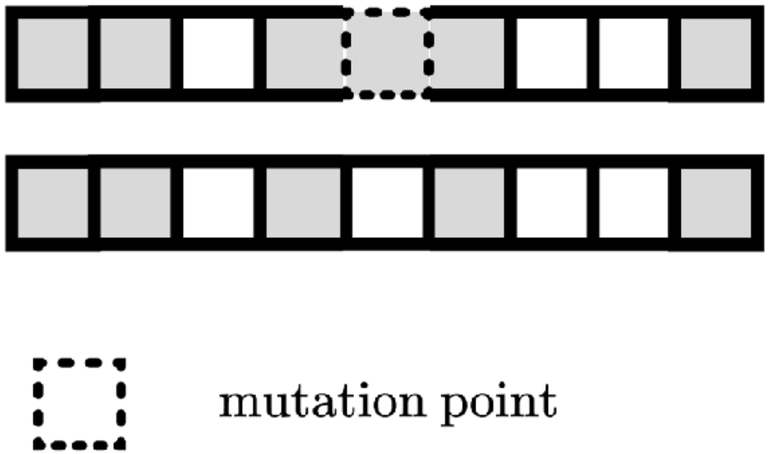
3

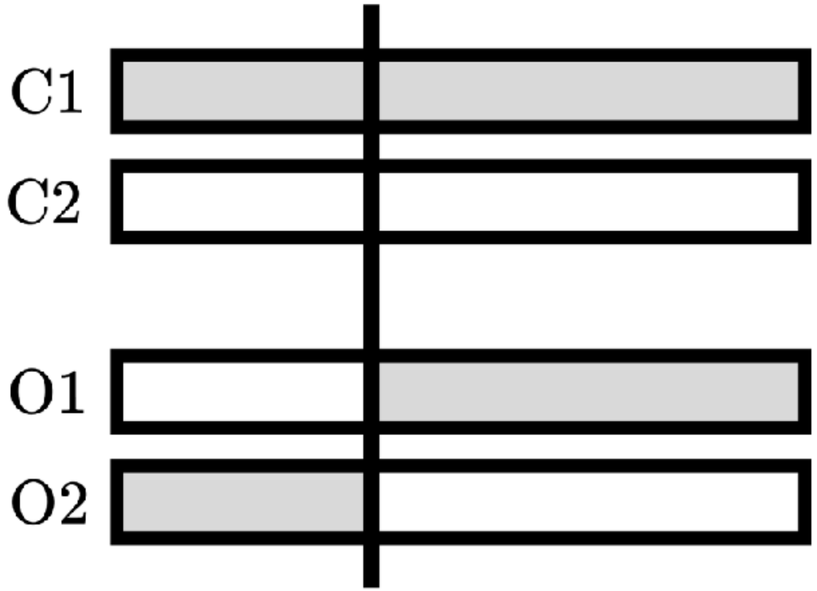
experiments on molecular boiling point optimisation

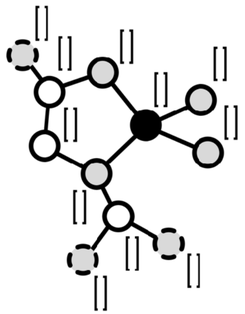


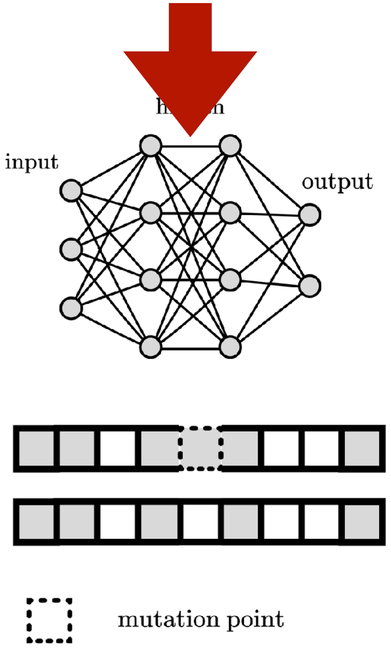
$$
\begin{aligned}
\text{maximise} \quad & x_1^K \\
\text{subject to} \quad & \boldsymbol{W}^K \boldsymbol{x}^{K-1} + b^K = x_1^K \\
& \boldsymbol{W}_j^k \boldsymbol{x}^{k-1} + b_j^k = x_j^k - s_j^k && \forall k \in \{1, \ldots, K-1\}, \forall j \in \{1, \ldots, n_k\} \\
& x_j^k \leq u_j^k z_j^k && \forall k \in \{1, \ldots, K-1\}, \forall j \in \{1, \ldots, n_k\} \\
& s_j^k \leq -l_j^k (1 - z_j^k) && \forall k \in \{1, \ldots, K-1\}, \forall j \in \{1, \ldots, n_k\} \\
& x_j^k, s_j^k \geq 0 && \forall k \in \{1, \ldots, K-1\}, \forall j \in \{1, \ldots, n_k\} \\
& z_j^k \in \{0, 1\} && \forall k \in \{1, \ldots, K-1\}, \forall j \in \{1, \ldots, n_k\} \\
& x^0 \in \Omega
\end{aligned}
$$

Matteo Fischetti and Jason Jo: Deep neural networks and mixed integer linear optimization. Constraints, 23(3):296–309, 2018

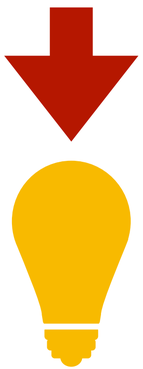




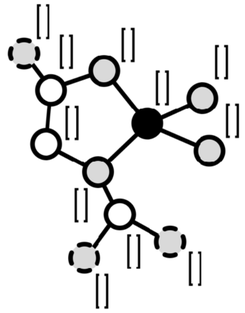


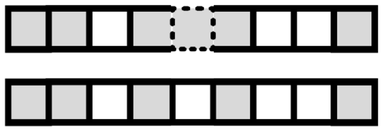


mutation point

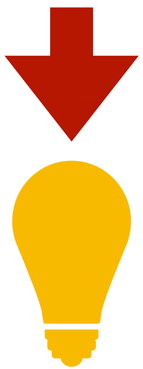


Directly optimise Graph Neural Networks with a (simple) genetic algorithm





mutation point



1

MINLP formulation of GCN

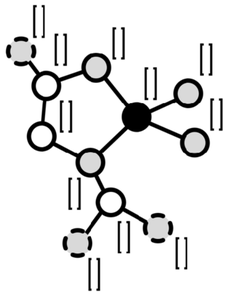
MILP formulation of GraphSAGE

2

genetic algorithm approach to optimise GNNs

3

experiments on molecular boiling point optimisation

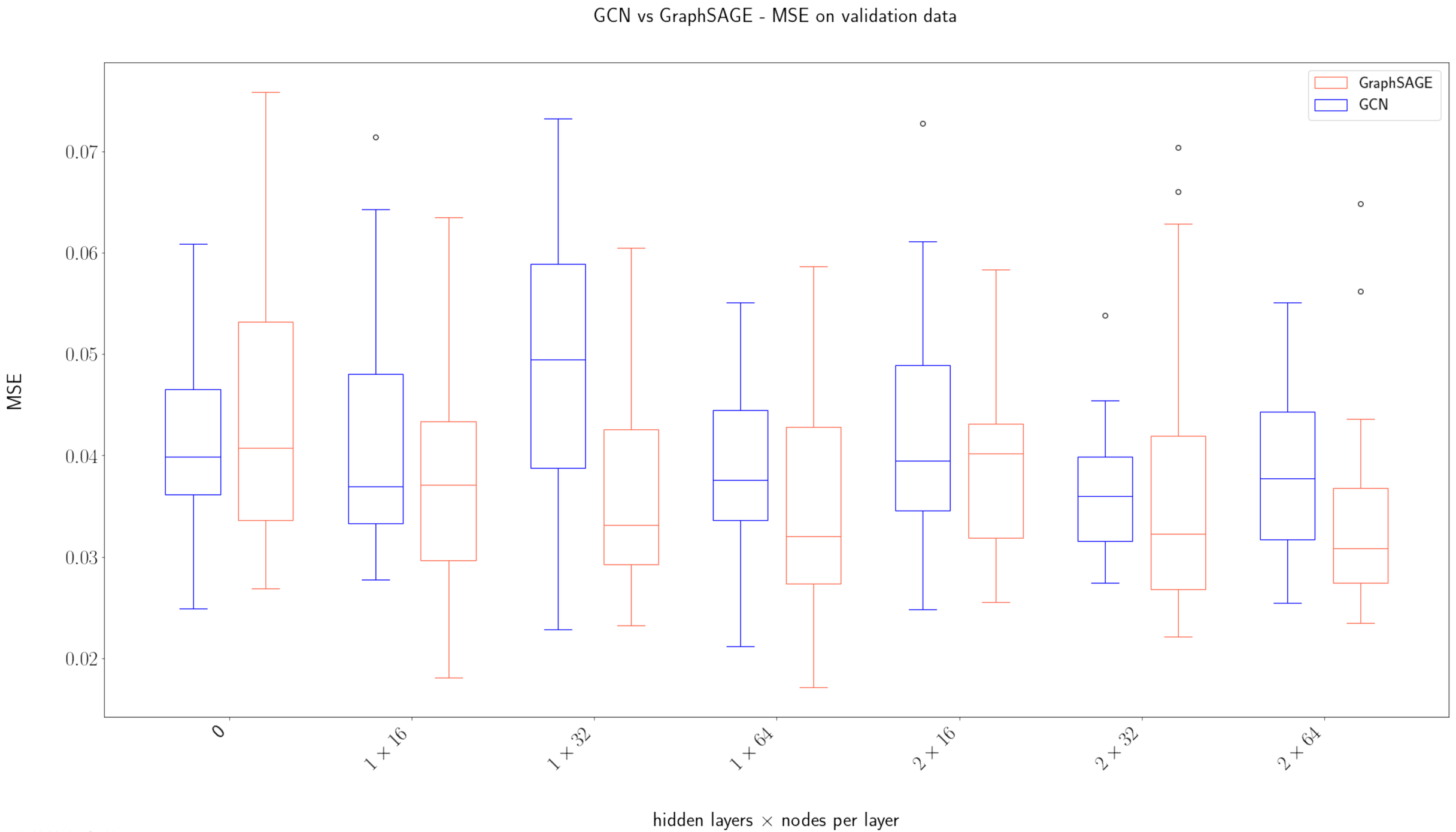


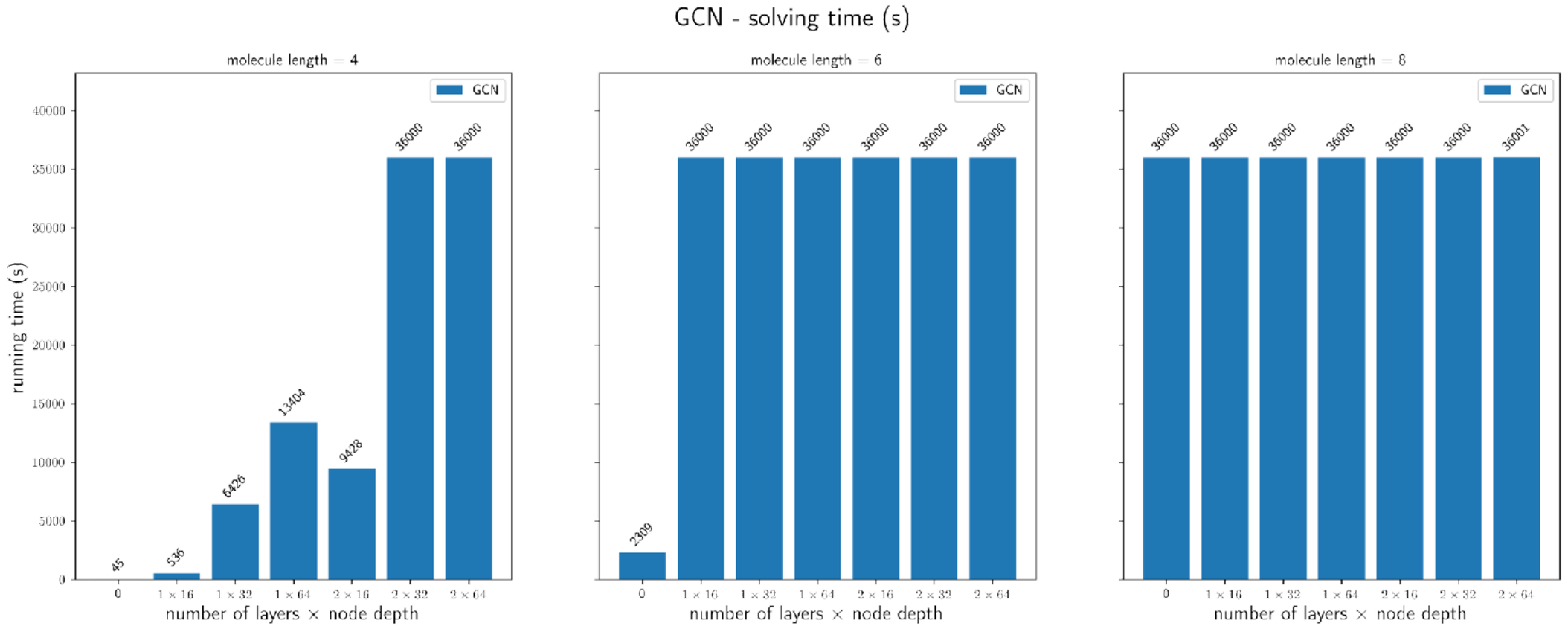
train GNN (using standard training mechanism)

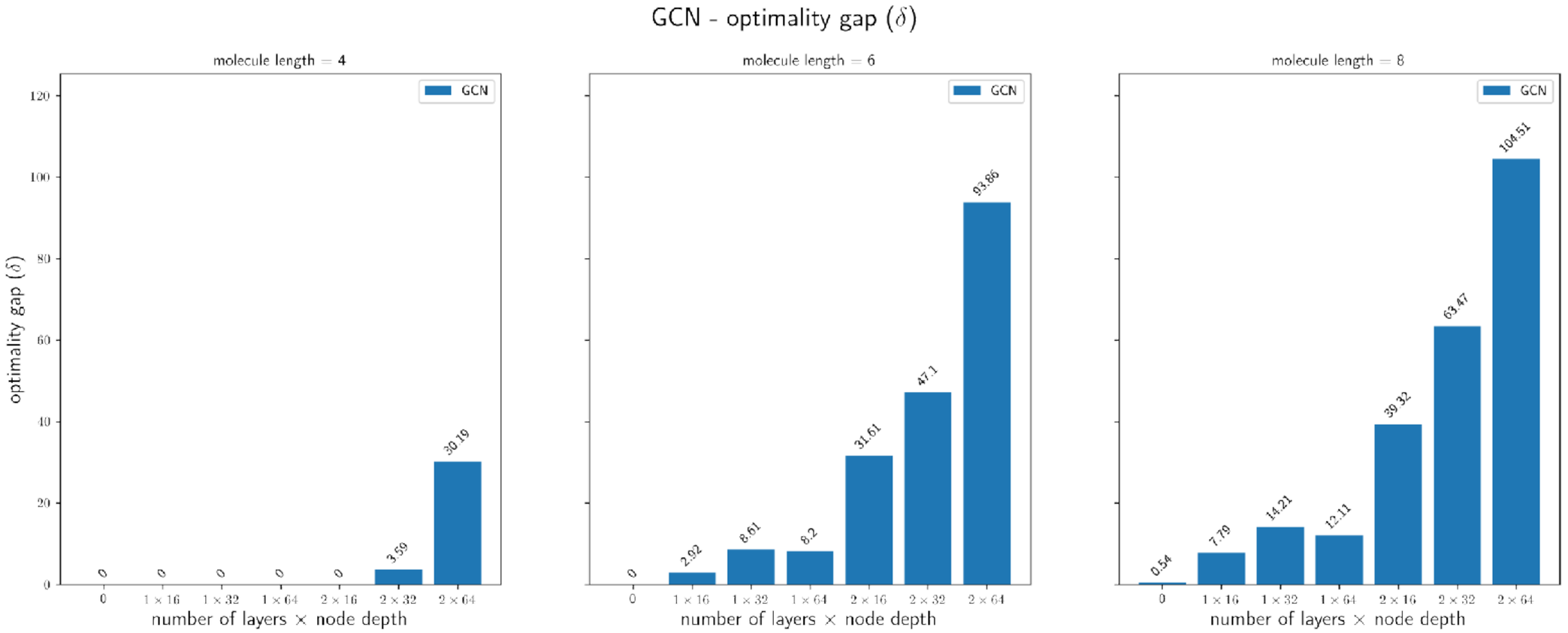
find molecular structures such that boiling point is optimised

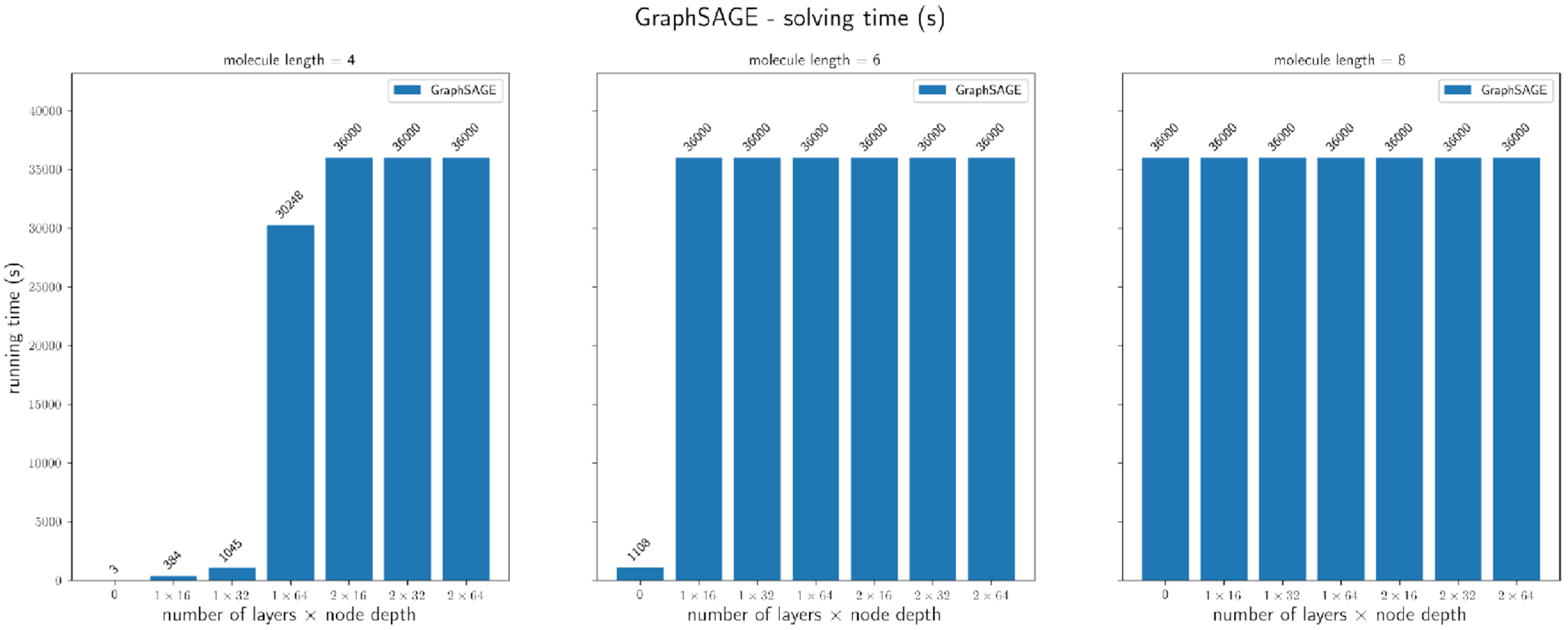
(using Gurobi solver with embedded trained GNN)

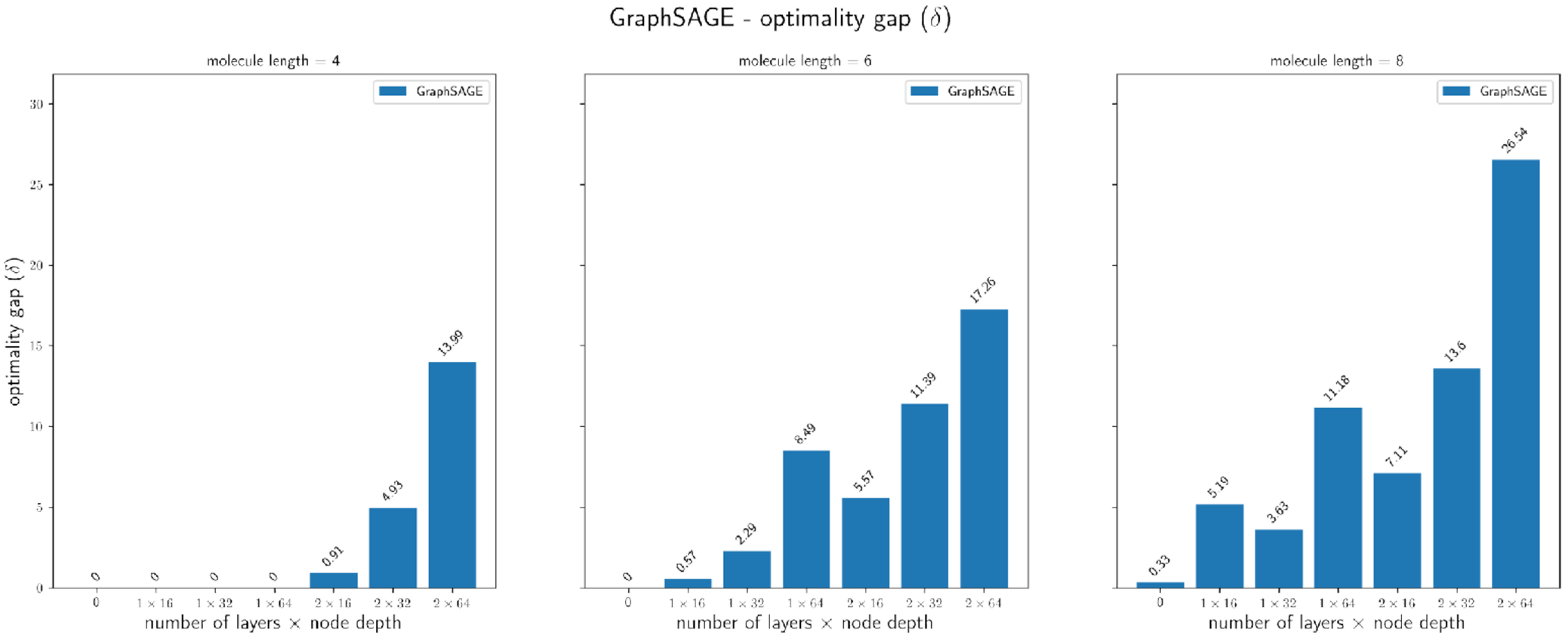
examine output molecules by human expert scientist

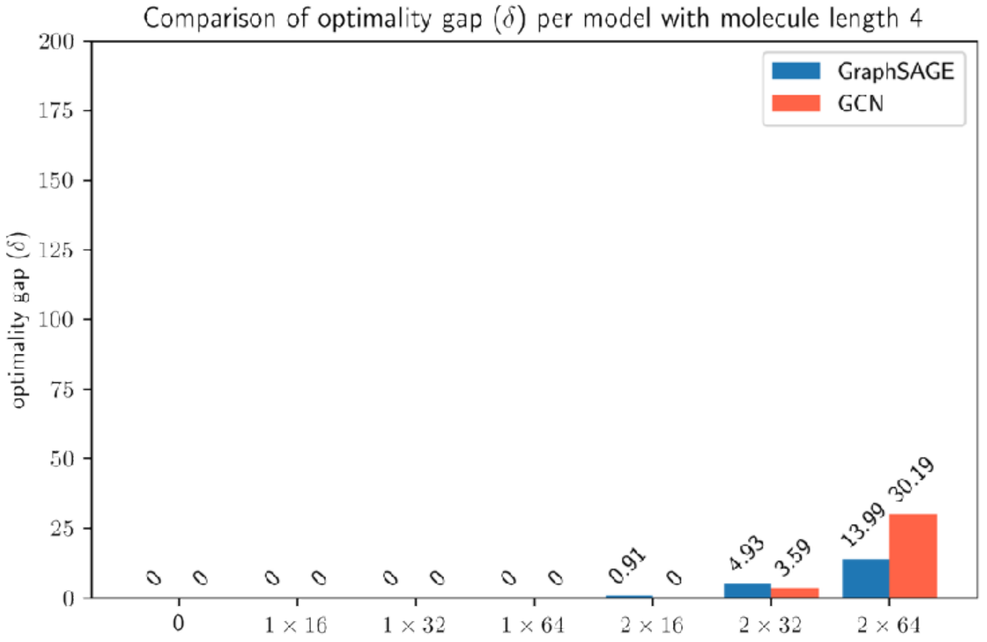


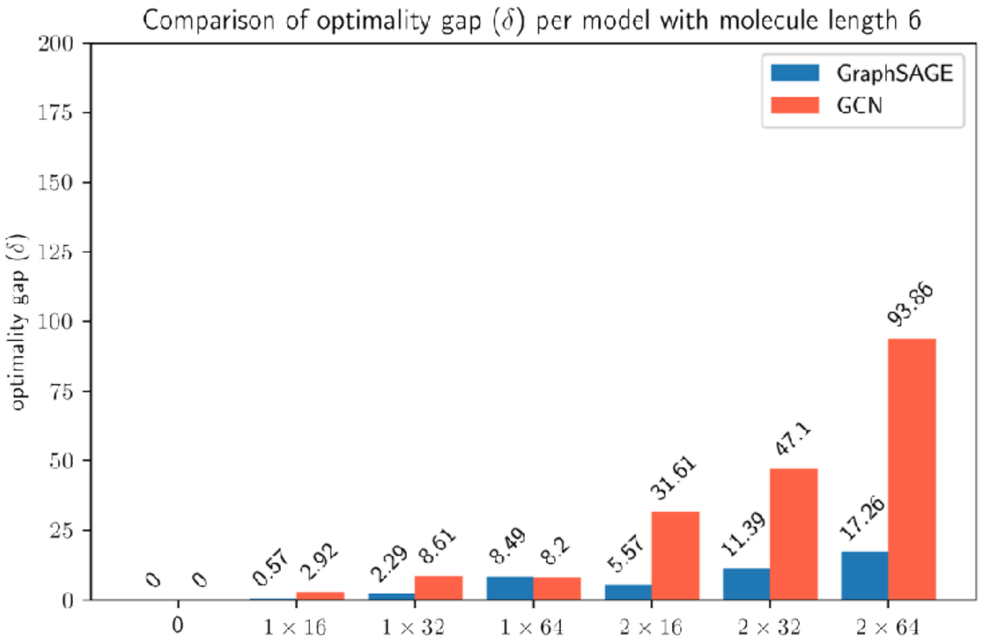


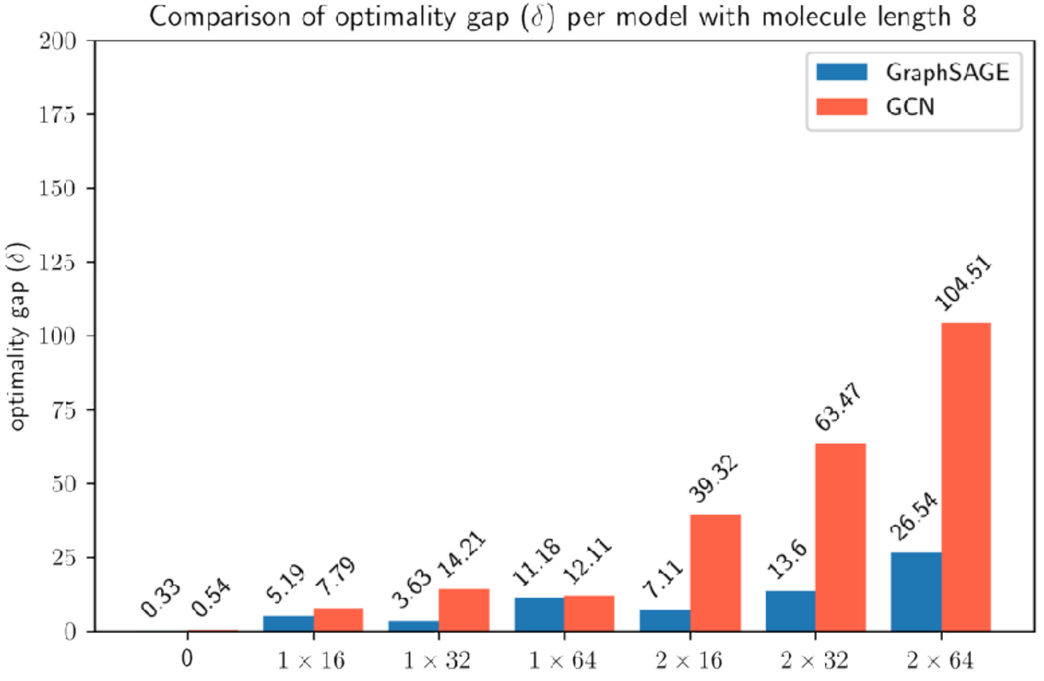


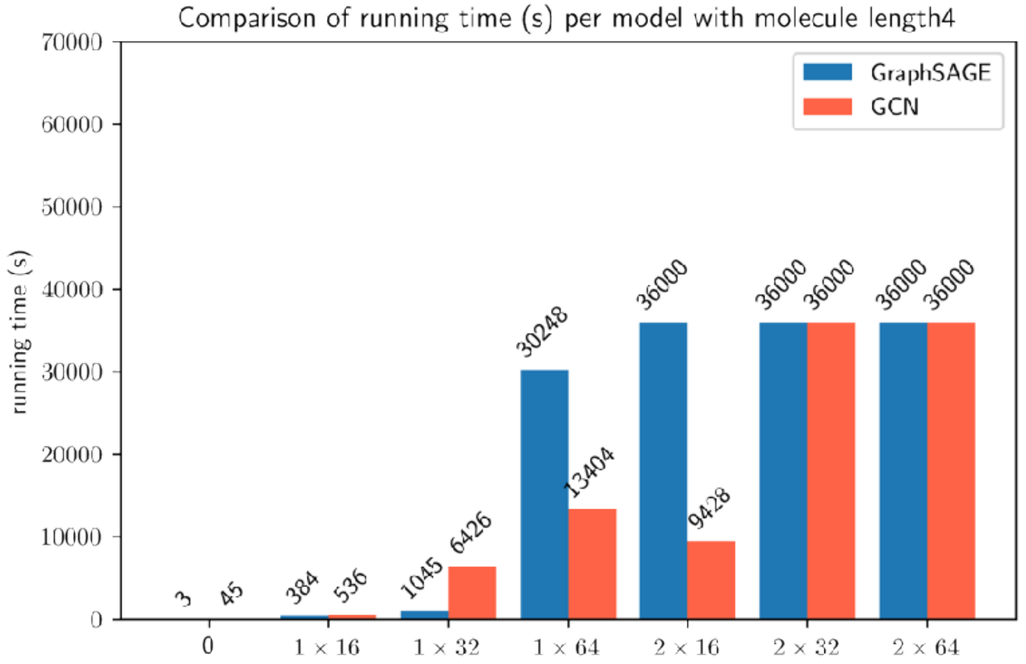


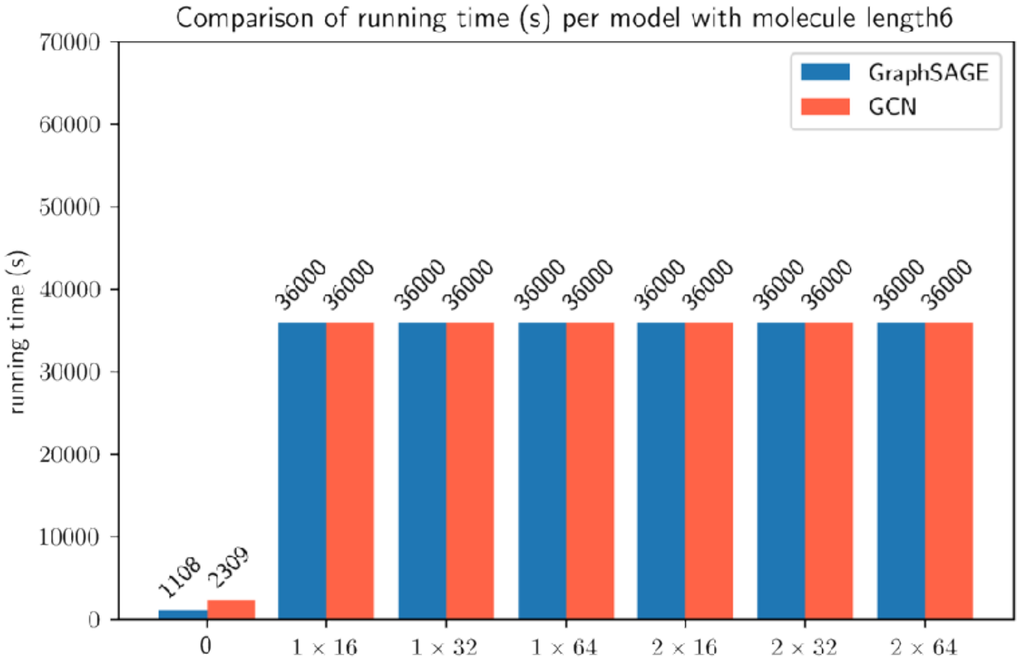


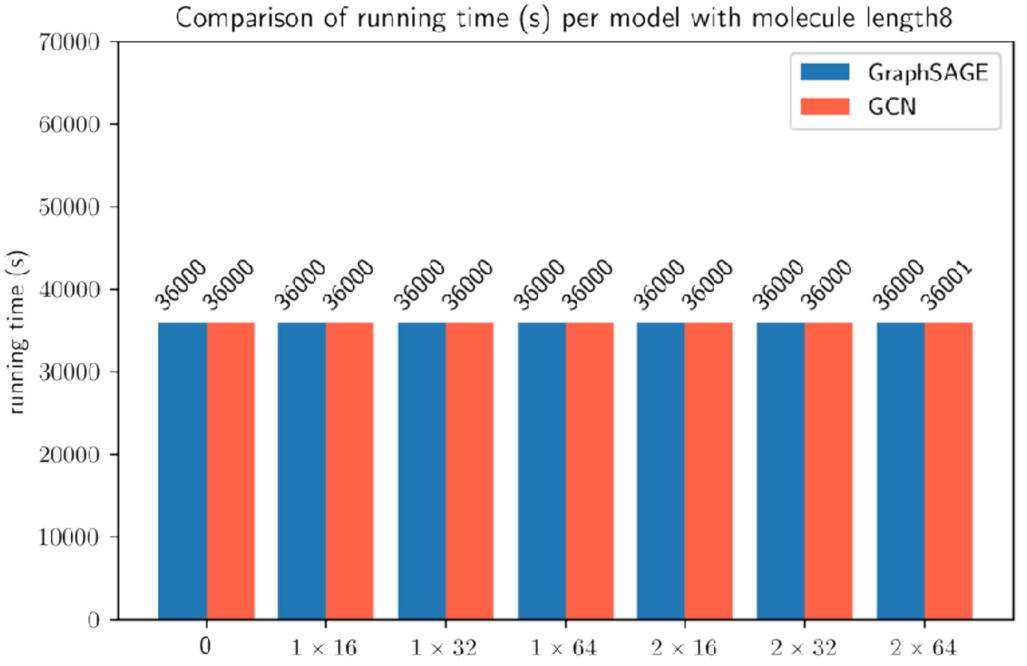


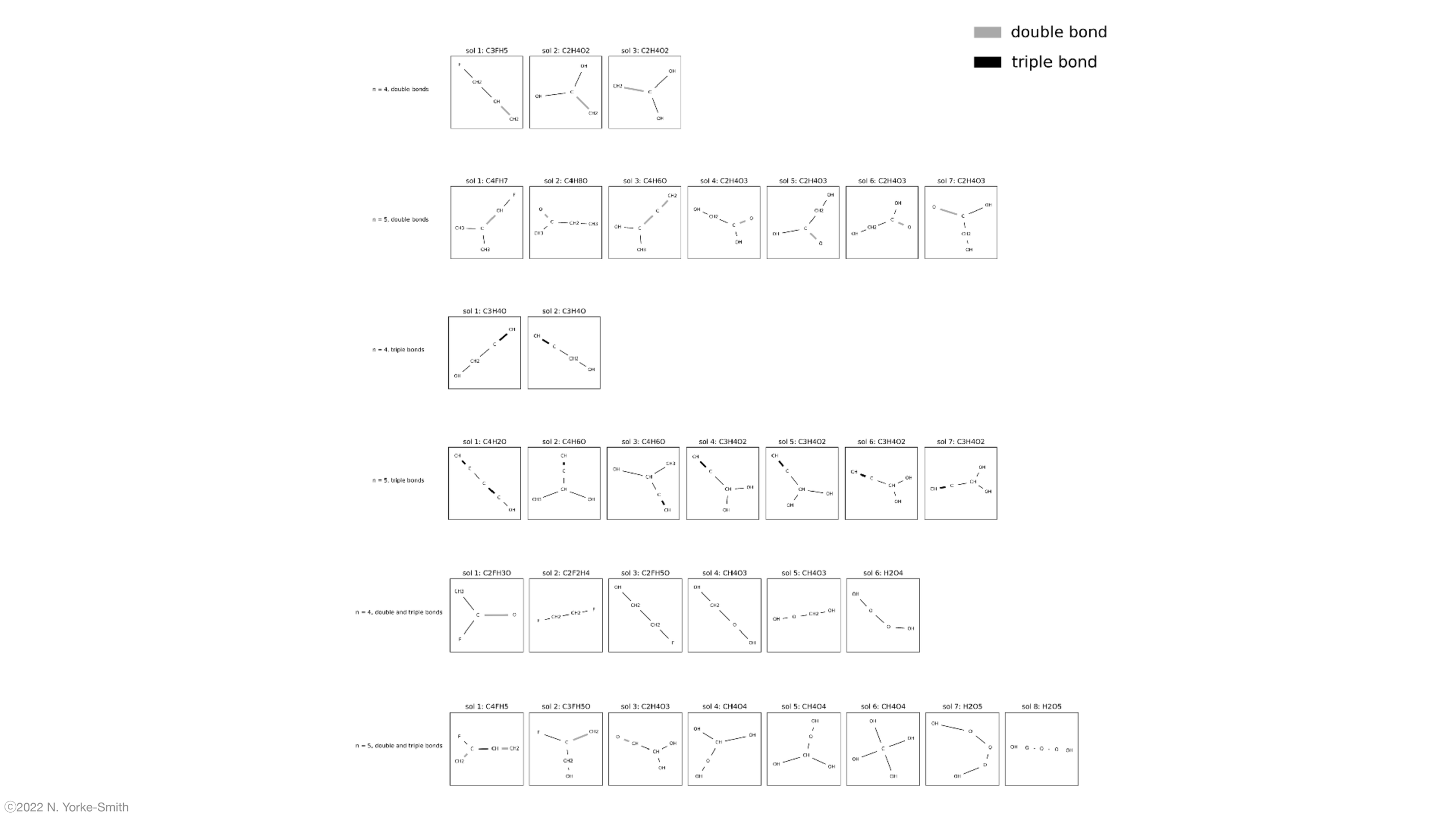












1

MINLP formulation of GCN

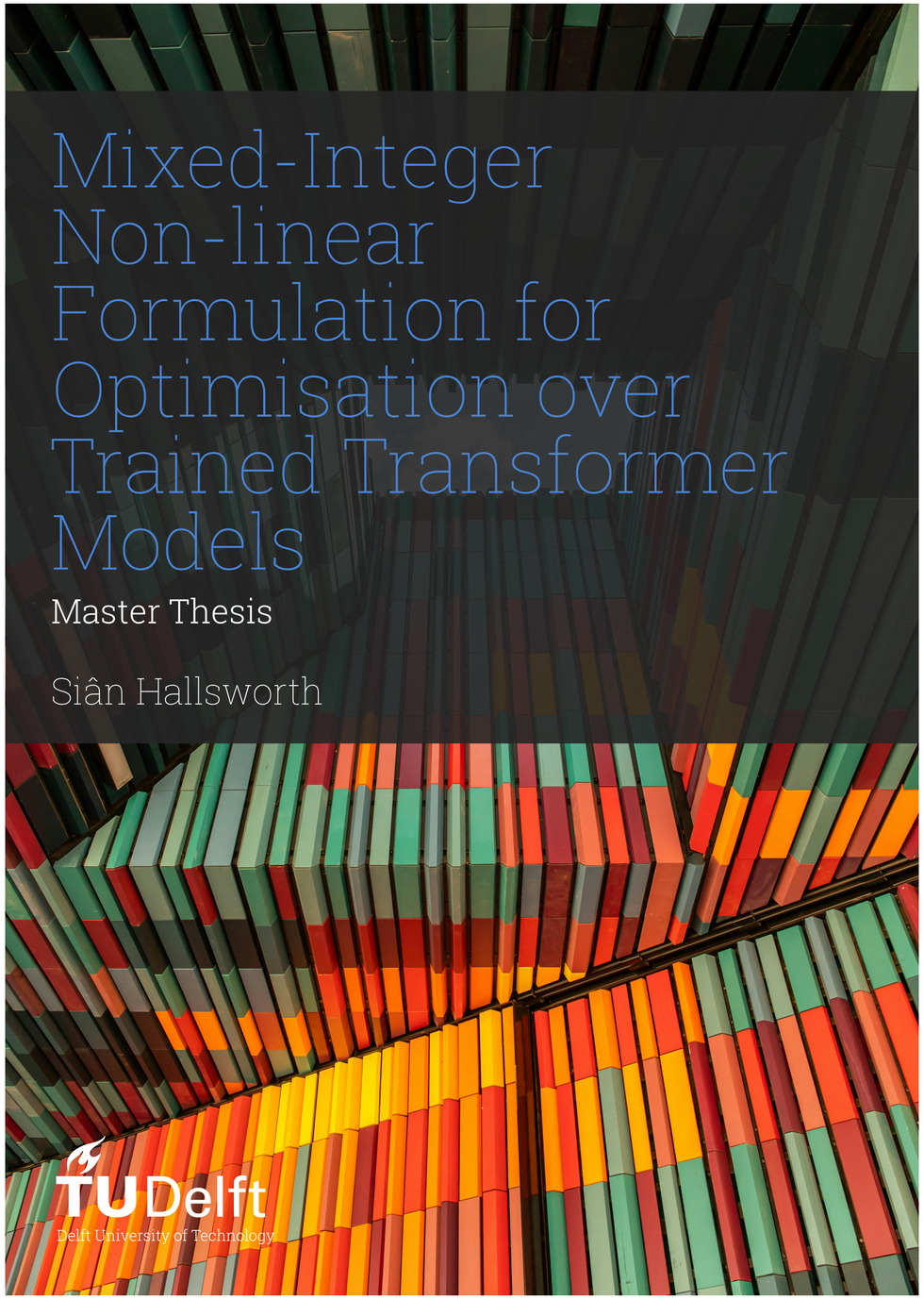
MILP formulation of GraphSAGE

2

genetic algorithm approach to optimise GNNs

3

experiments on molecular boiling point optimisation



Computers & Chemical Engineering
Volume 185, June 2024, 108660

Mixed-integer optimisation of graph neural networks for computer-aided molecular design

Tom McDonald [a d], Calvin Tsay [b], Artur M. Schweidtmann [c], Neil Yorke-Smith [d]

Show more

+ Add to Mendeley Share Cite

https://doi.org/10.1016/j.compchemeng.2024.108660 Get rights and content

Highlights

- Mixed-integer linear programming formulation of graph neural networks.
- Formulations for Graph Convolutional Network and GraphSAGE.
- Showcase that such models can learn from non-euclidean data structures efficiently.
- Sample application to molecule discovery with optimal boiling points.