Using Reinforcement Learning to solve Quadratic Assignment Problems

Final Talk

Tim Göttlicher

Supervisors

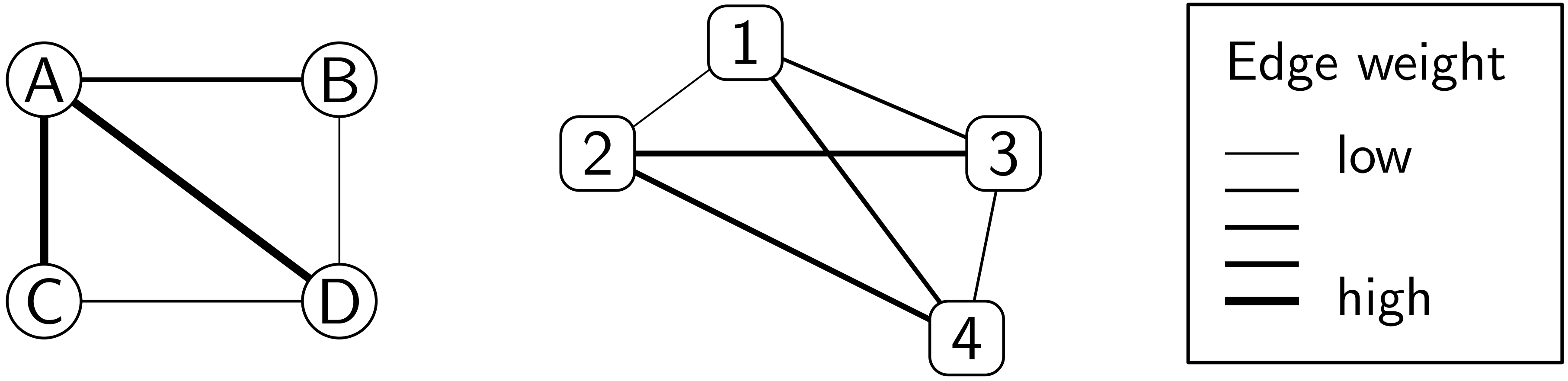
Prof Dr. Verena Wolf

Dr. Andreas Karrenbauer

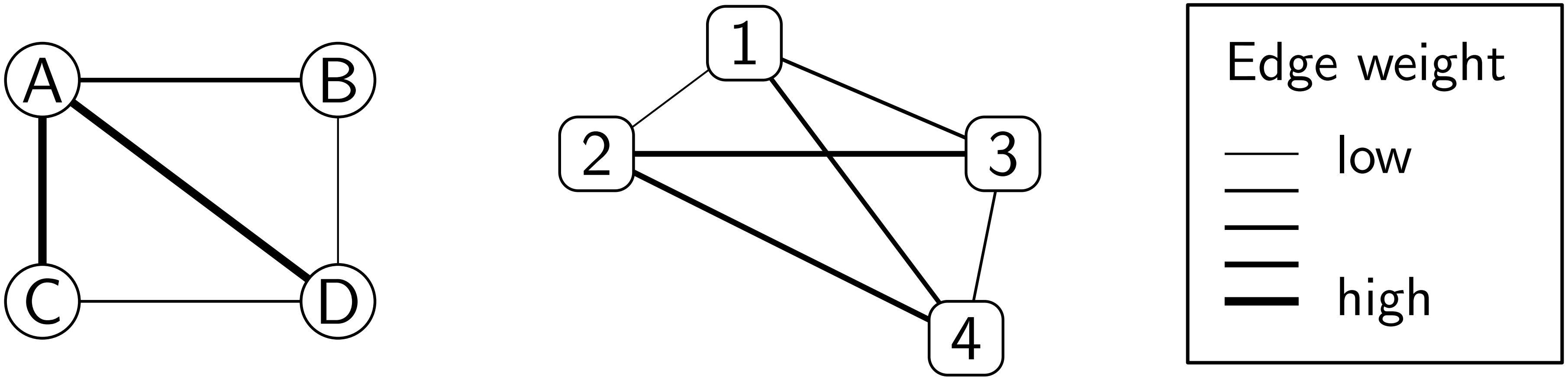
Joschka Groß

15.07.2022

The Quadratic Assignment Problem



The Quadratic Assignment Problem

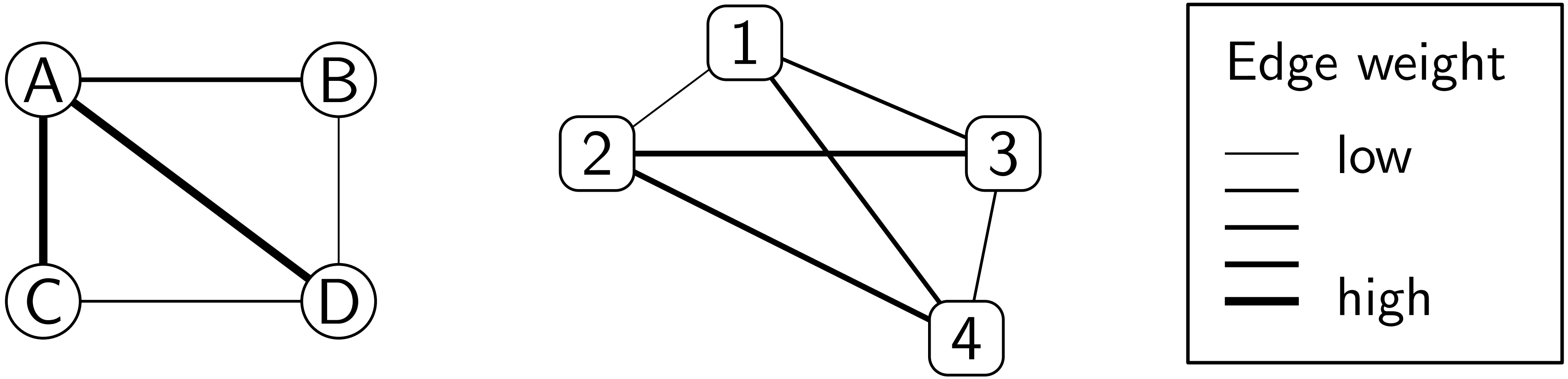


Example: Economics

Transport volume between facilities

Cost per unit between locations

The Quadratic Assignment Problem



Example: Economics

Transport volume
between facilities

Cost per unit
between locations

Example: Keyboard layout

Letter pair frequency

Travel time between keys

Mathematical formulation

$$\sum_{i,j} a_{i,j} b_{f(i),f(j)}$$

Mathematical formulation

edge weights

$$\sum_{i,j} a_{i,j} b_{f(i),f(j)}$$

Mathematical formulation

edge weights

$$\sum_{i,j} a_{i,j} b_{f(i),f(j)}$$

edges in left graph

Mathematical formulation

edge weights

$$\sum_{i,j} a_{i,j} b_{f(i),f(j)}$$

edges in left graph

assigned edge in right graph

Mathematical formulation

edge weights

$$\sum_{i,j} a_{i,j} b_{f(i),f(j)} + \sum_i c_{i,f(i)}$$

edges in left graph

assigned edge in right graph

Mathematical formulation

edge weights

$$\sum_{i,j} a_{i,j} b_{\underline{f(i),f(j)}} + \sum_i c_{i,f(i)}$$

edges in left graph

assigned edge in right graph

Goal

Find assignment f that minimizes cost

Mathematical formulation

edge weights

$$\sum_{i,j} a_{i,j} b_{\underline{f(i),f(j)}} + \sum_i c_{i,f(i)}$$

edges in left graph

assigned edge in right graph

Goal

Find assignment f that minimizes cost

$\rightarrow$ How can RL be applied?

Transition function

$$(\text{state}, \text{action}) \rightarrow (\text{new state}, \text{reward})$$

Transition function

$$(\text{state}, \text{action}) \to (\text{new state}, \text{reward})$$

$$T : (\mathrm{QAP}_n, \{1, \dots, n\}^2) \to$$

Transition function

$$(\text{state}, \text{action}) \to (\text{new state}, \text{reward})$$

$$T : (\mathrm{QAP}_n, \underbrace{\{1, \ldots, n\}^2}_{\text{pair of nodes}}) \to$$

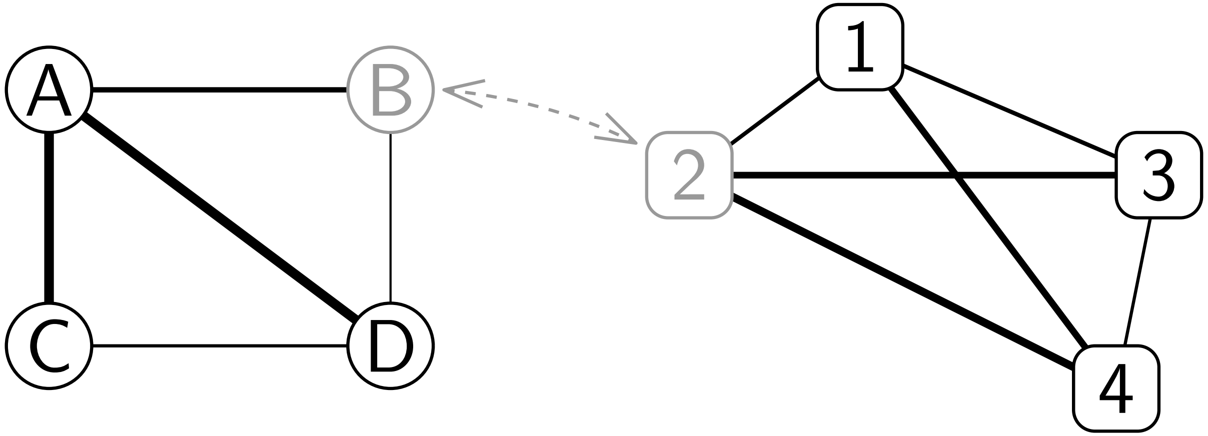
Transition function

$$(\text{state}, \text{action}) \rightarrow (\text{new state}, \text{reward})$$

$$T : (\mathrm{QAP}_n, \{1, \ldots, n\}^2) \rightarrow (\mathrm{QAP}_{n-1}, \mathbb{R})$$

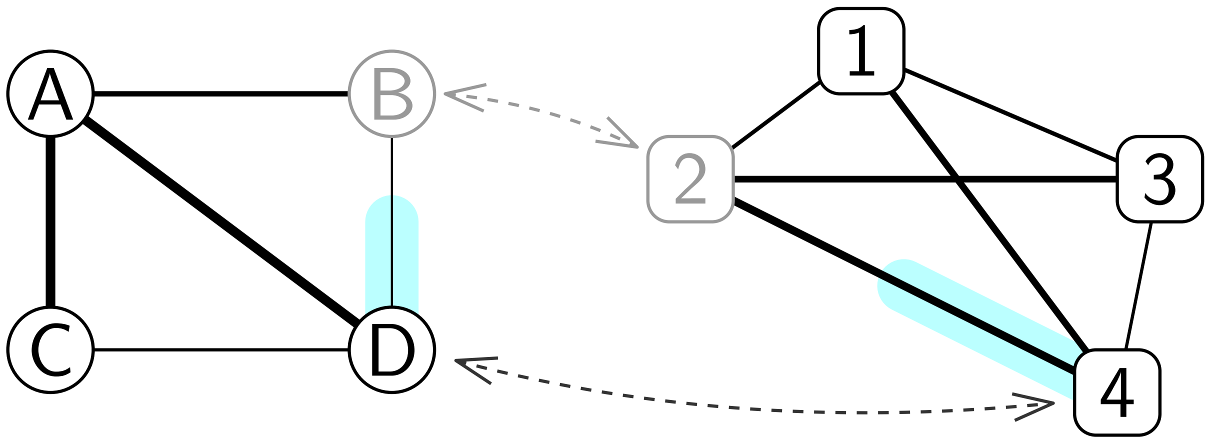
Reduced subproblem representation

Can we remove the assigned nodes from the graph?



Reduced subproblem representation

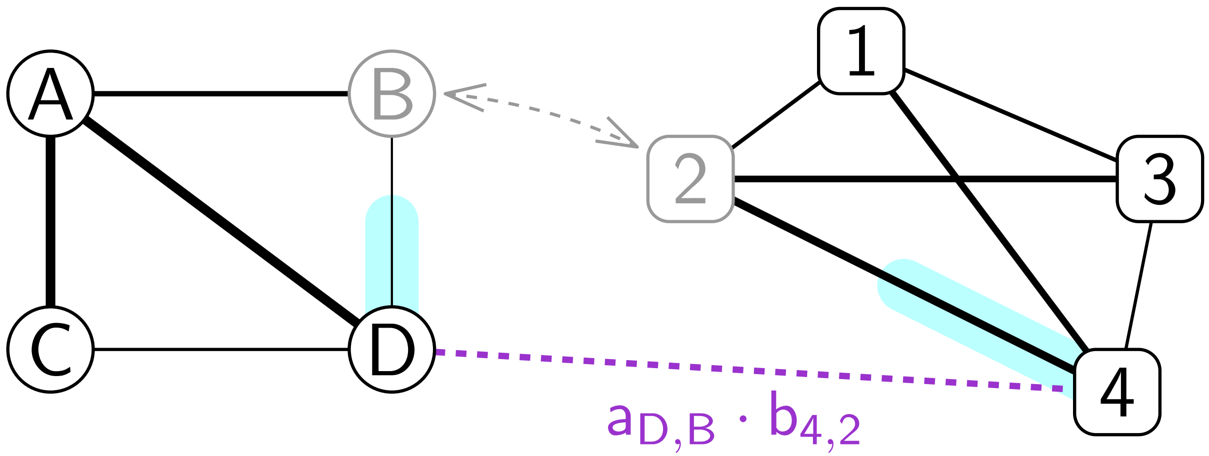
Can we remove the assigned nodes from the graph?



Assignment cost based on half assigned edges

Reduced subproblem representation

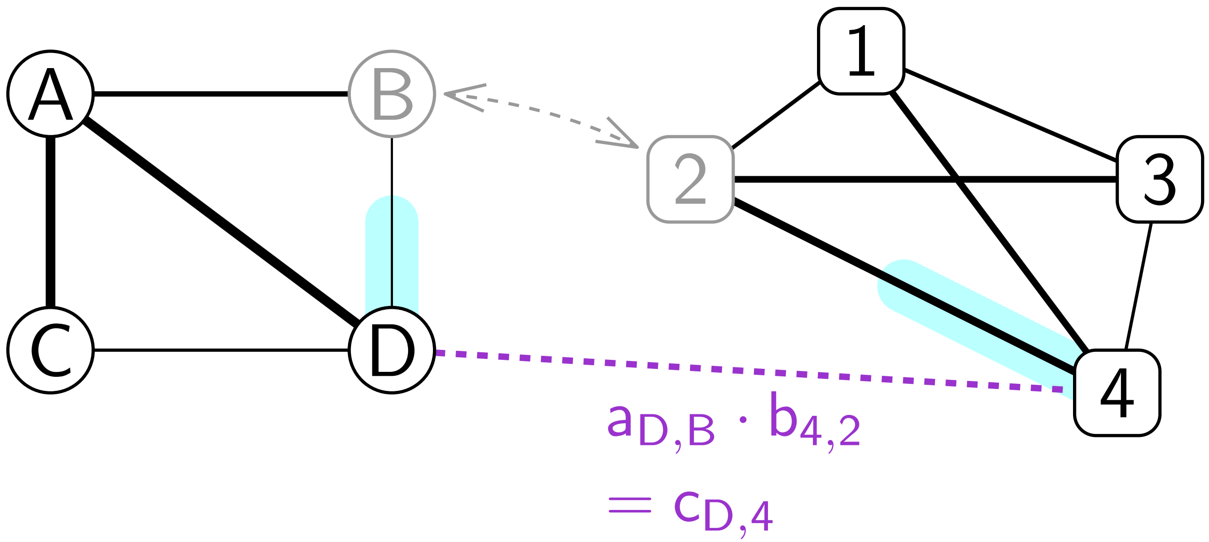
Can we remove the assigned nodes from the graph?



Assignment cost based on half assigned edges

Reduced subproblem representation

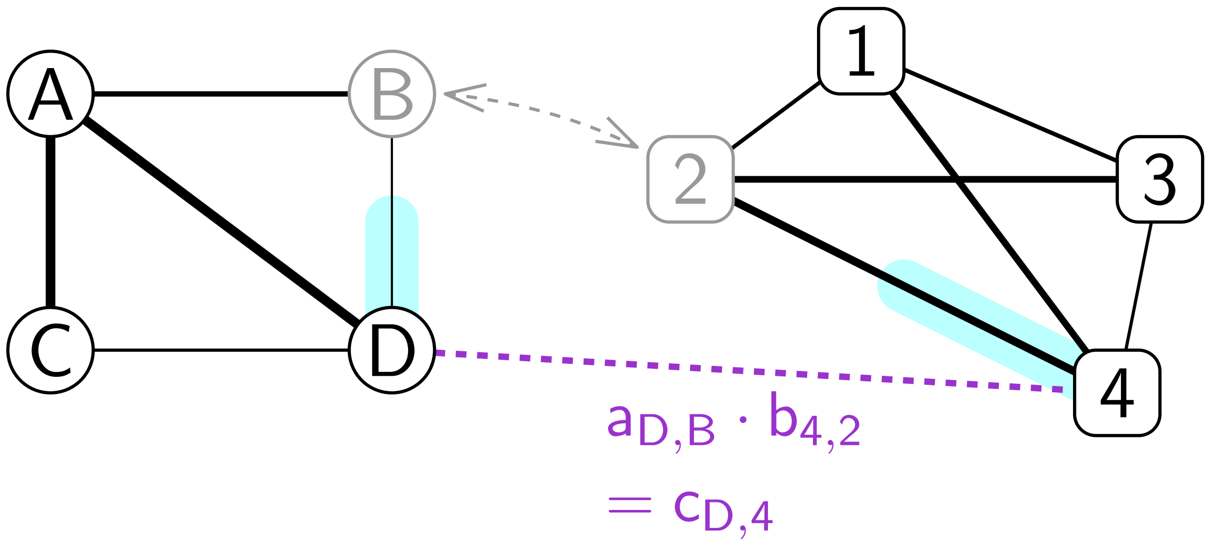
Can we remove the assigned nodes from the graph?



Assignment cost based on half assigned edges

Reduced subproblem representation

Can we remove the assigned nodes from the graph?

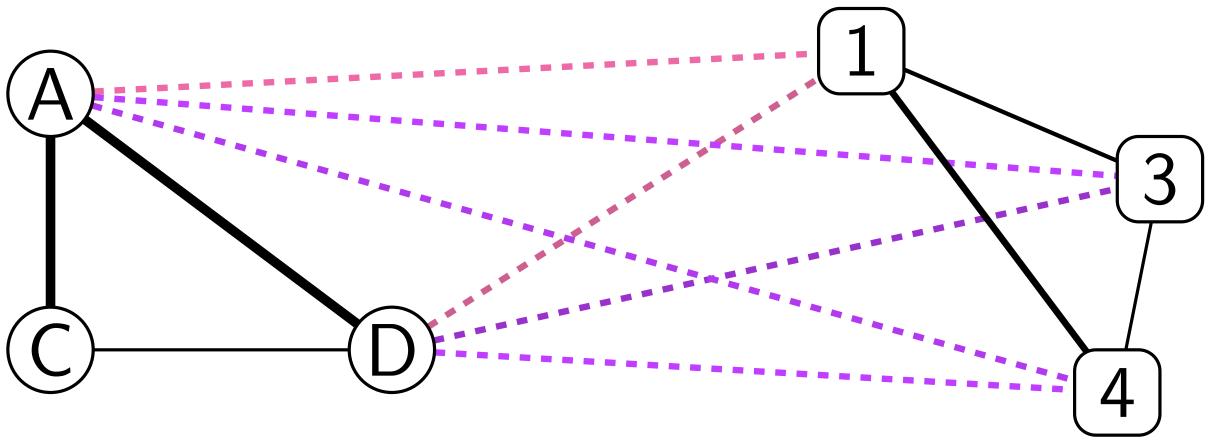


Assignment cost based on half assigned edges

$\rightarrow$ Encode in new edge type between graphs

Reduced subproblem representation

Can we remove the assigned nodes from the graph?



Assignment cost based on half assigned edges

$\rightarrow$ Encode in new edge type between graphs

Advantages of closed transitions

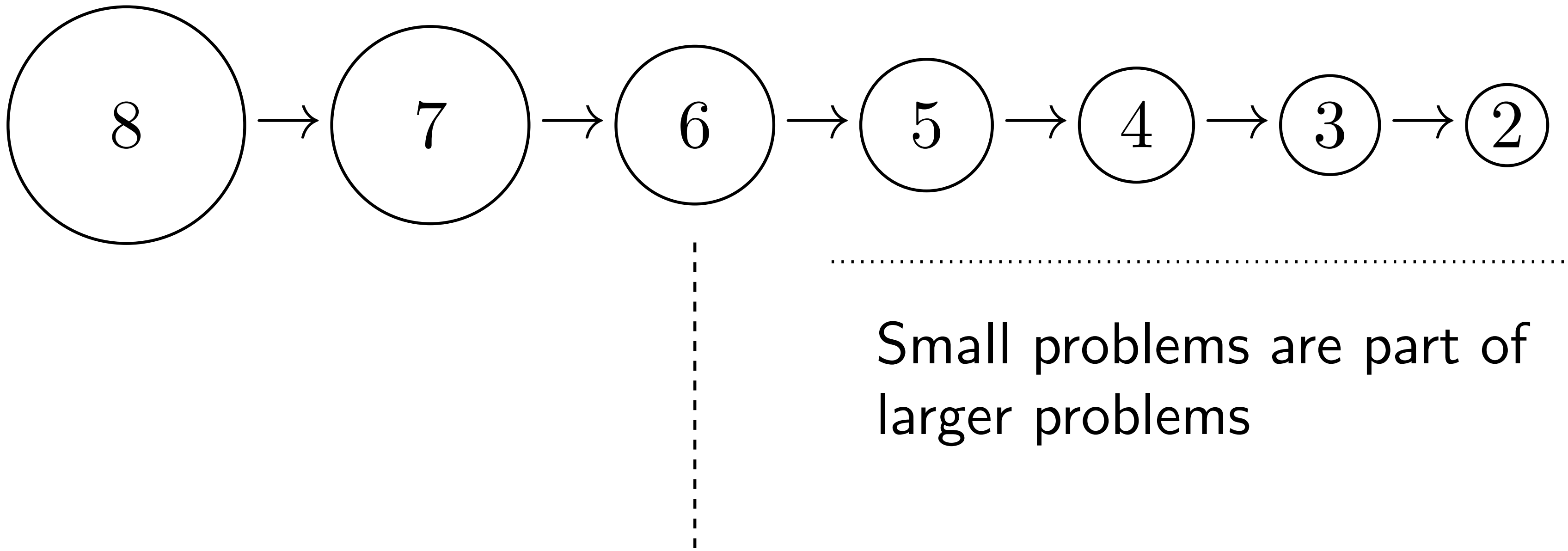
$$8 \to 7 \to 6 \to 5 \to 4 \to 3 \to 2$$

Advantages of closed transitions

$$8 \to 7 \to 6 \to 5 \to 4 \to 3 \to 2$$

Small problems are part of larger problems

Advantages of closed transitions



Small problems are part of larger problems

No artificial states

Advantages of closed transitions

$$8 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2$$

Small problems are part of larger problems

No artificial states

$\longleftarrow$

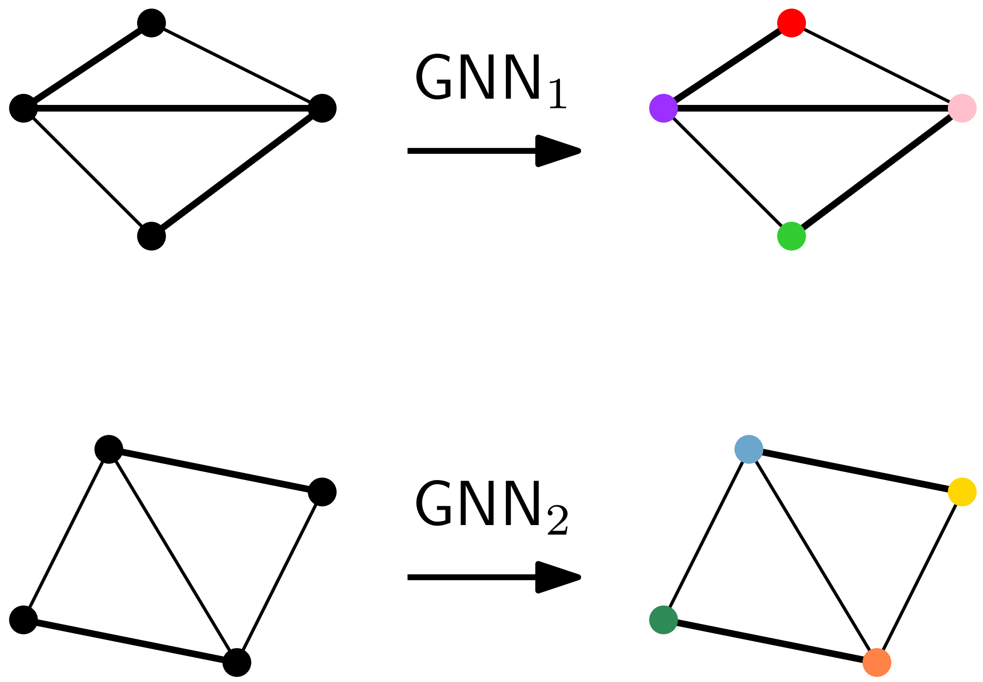
incremental learning

Policy network architecture

Compute value for each action

Policy network architecture

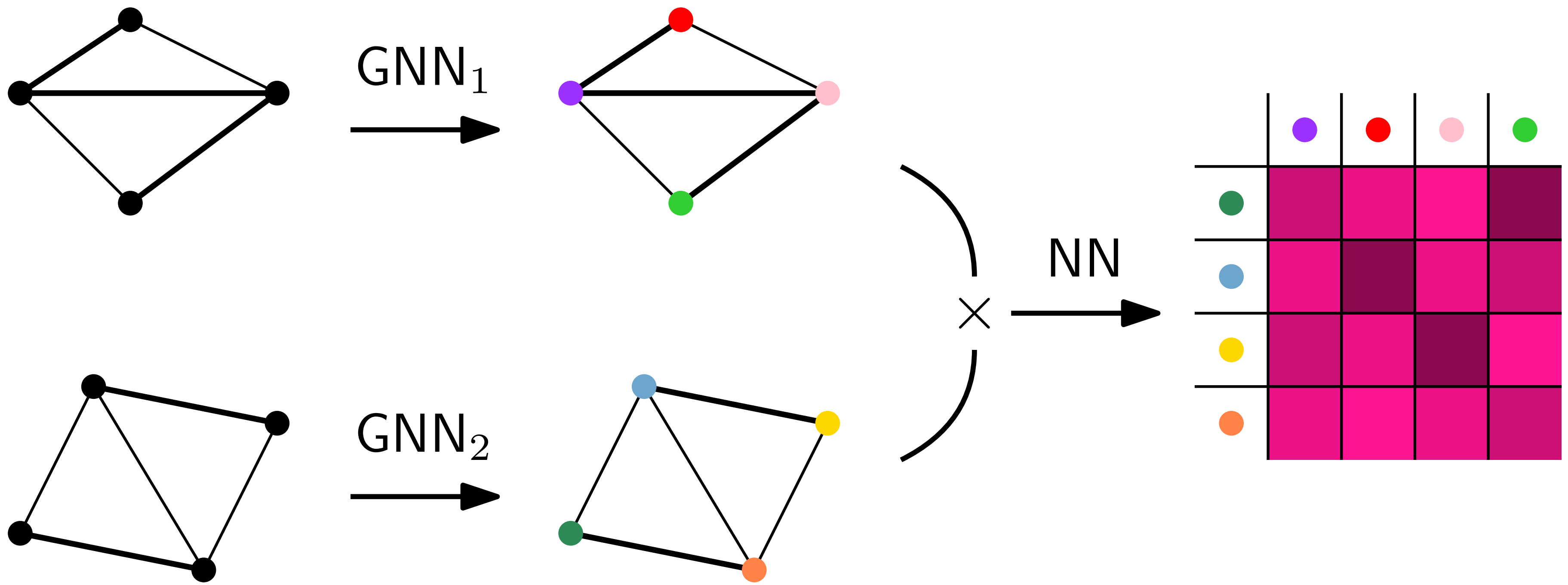
Compute value for each action



Encode graph structure

Policy network architecture

Compute value for each action

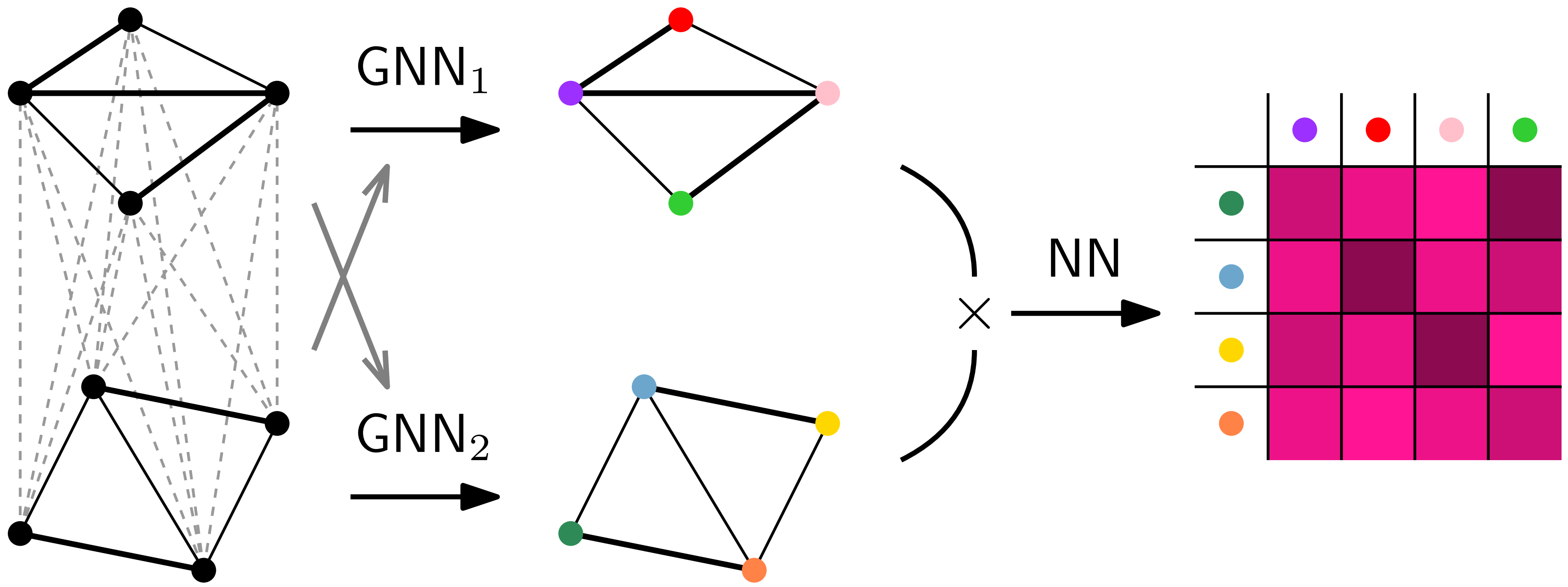


Encode graph structure

Predict pair values

Policy network architecture

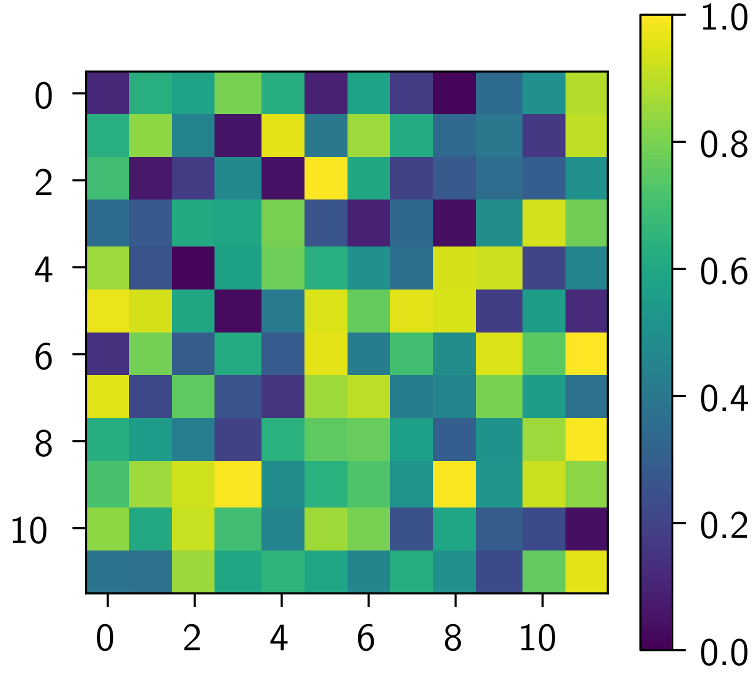
Compute value for each action



Encode graph structure

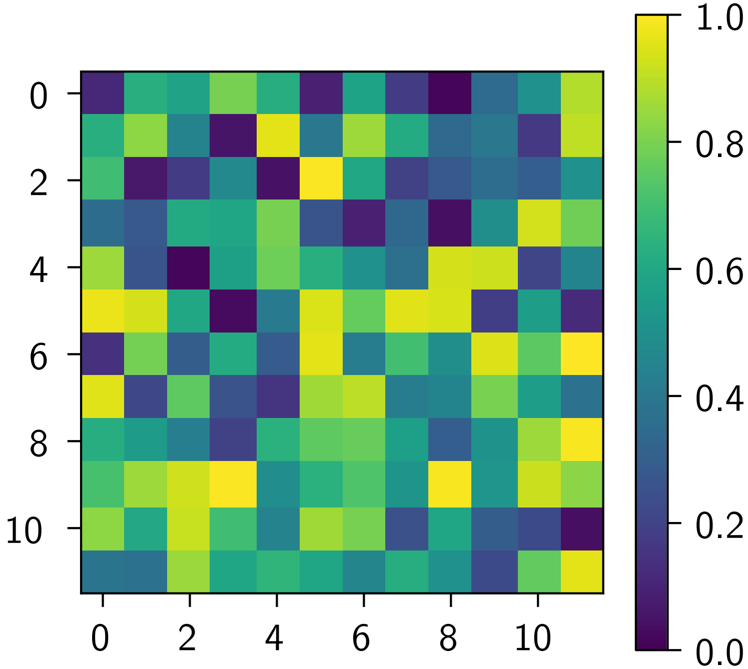
Predict pair values

Separation experiment: Setup

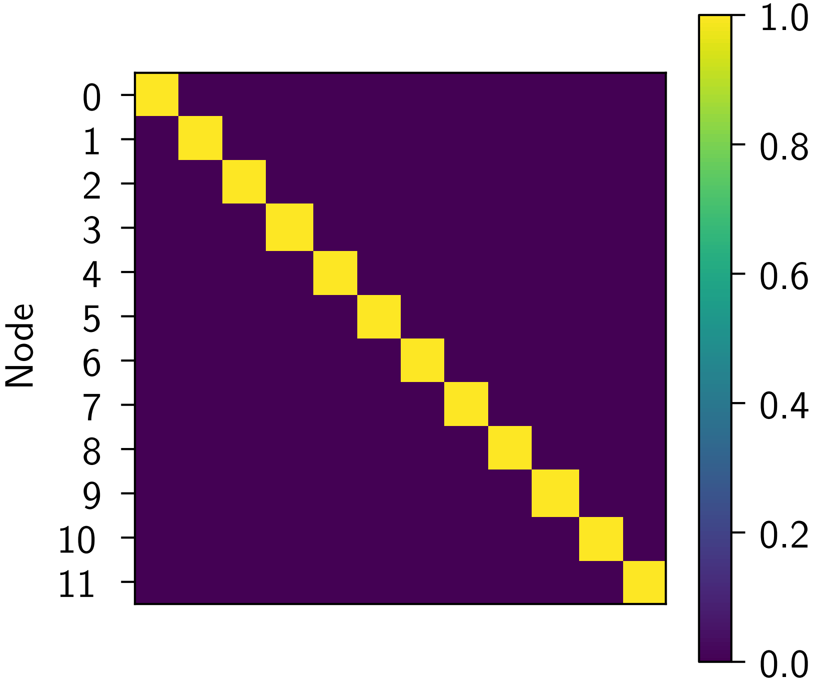


Input connectivity matrix

Separation experiment: Setup

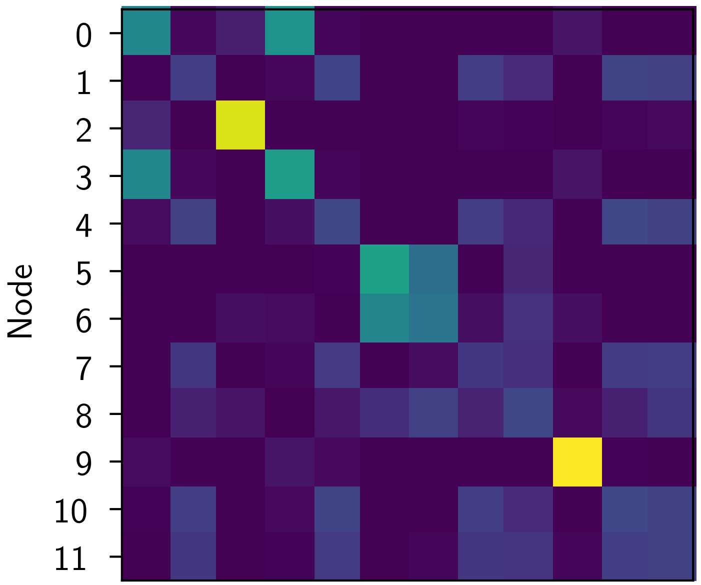


Input connectivity matrix



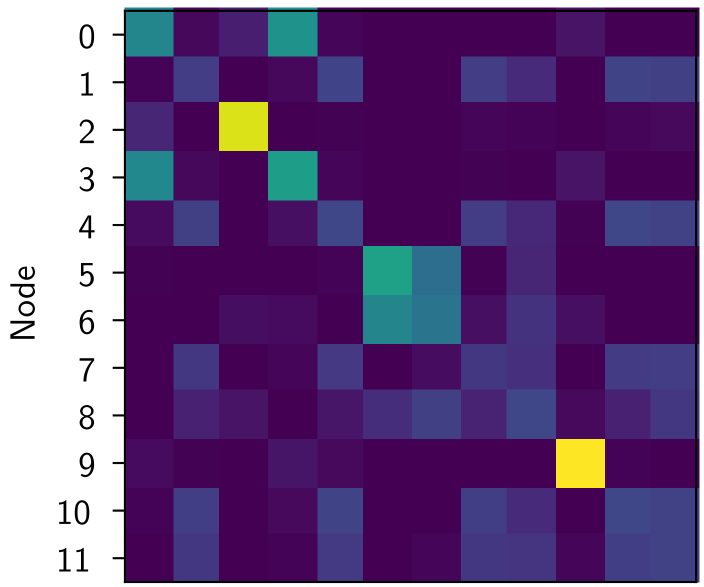
Target node embeddings

Separation experiment: Results

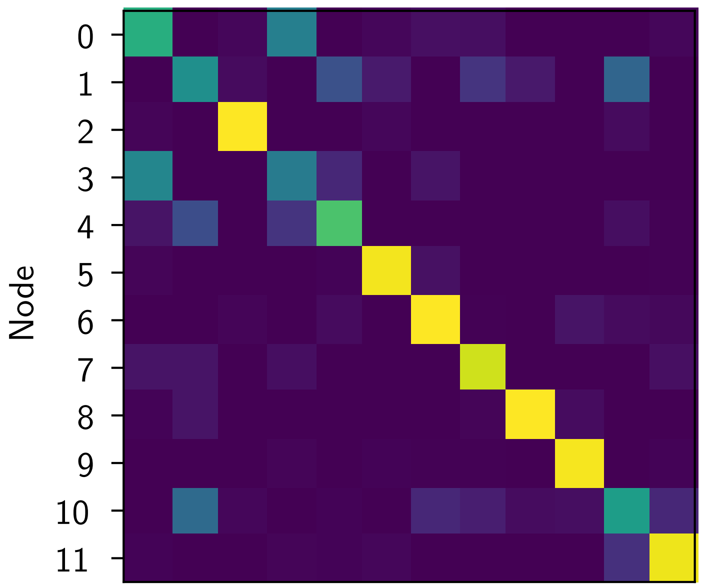


Output of trained GNN

Separation experiment: Results

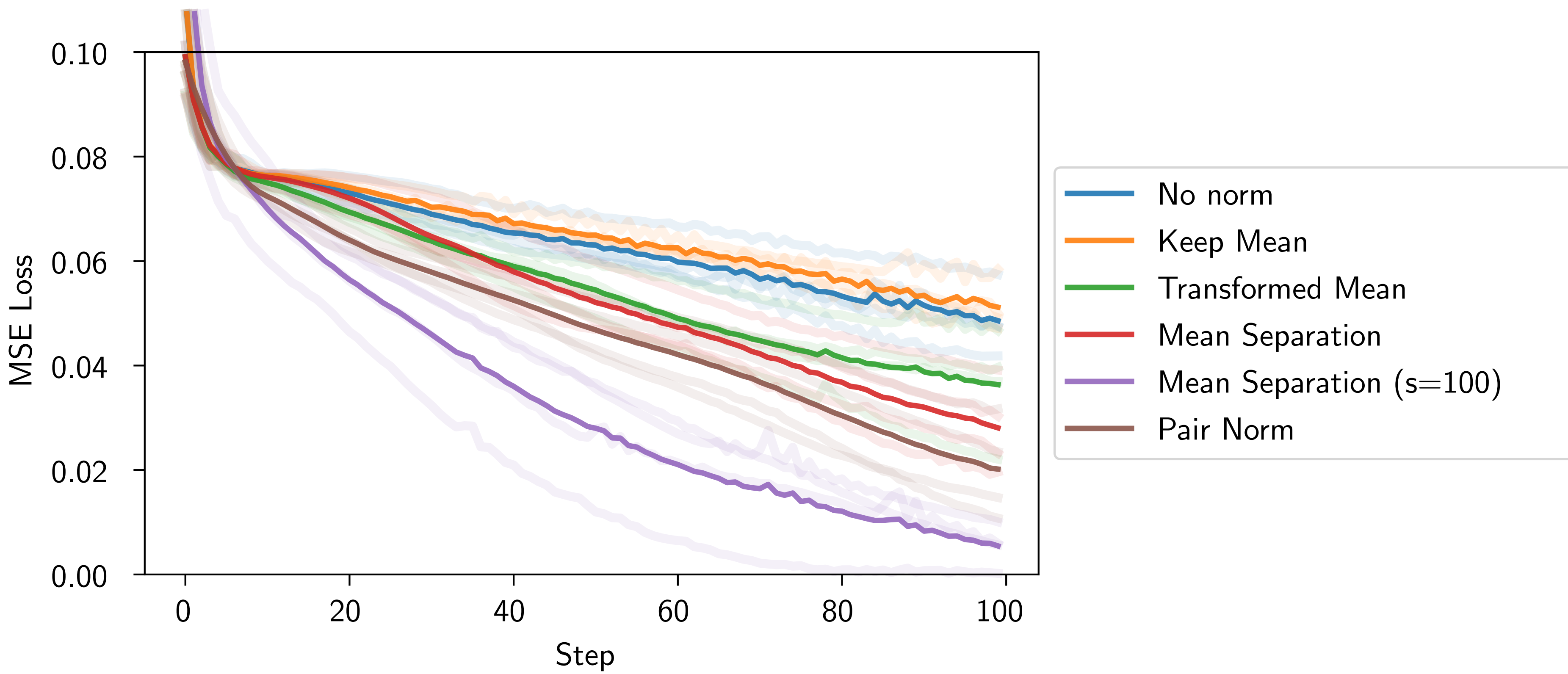


Output of trained GNN



PairNorm

Separation experiment: Training progress comparison



Experiment setup

Experiment setup

Problem generator

Undirected 8 node graphs with random weights

Experiment setup

Problem generator

Undirected 8 node graphs with random weights

$8! = 40320$ possible assignments

Experiment setup

Problem generator

Undirected 8 node graphs with random weights

$8! = 40320$ possible assignments

RL algorithms

	Value-based	Policy gradient
Monte Carlo		
Temporal difference		

Experiment setup

Problem generator

Undirected 8 node graphs with random weights

$8! = 40320$ possible assignments

RL algorithms

	Value-based	Policy gradient
Monte Carlo	MCQ	REINFORCE
Temporal difference	DQN	A2C

Performance indicators

Performance indicators

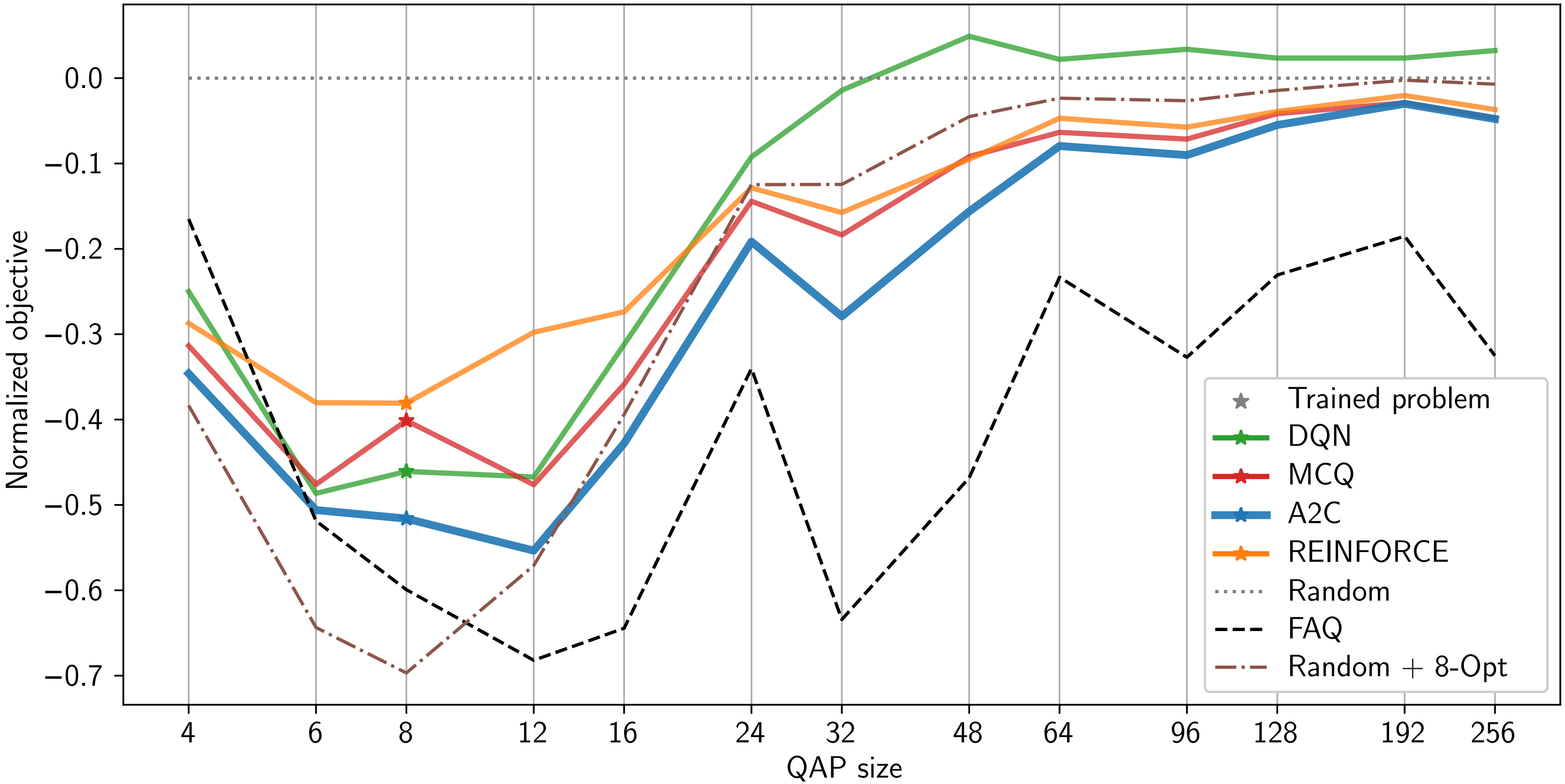
Gap to best known value

$$\frac{v - v_{\text{known}}}{v_{\text{known}}}$$

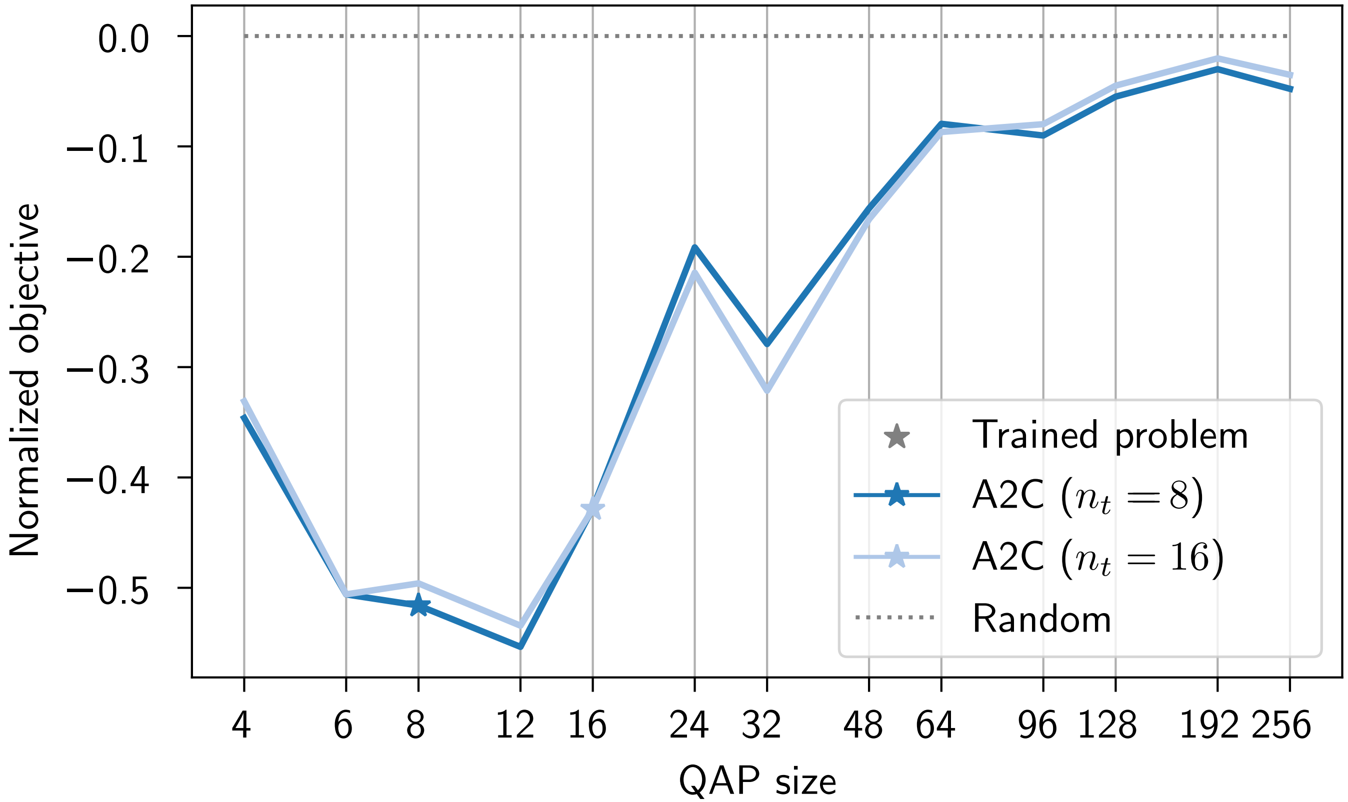
Performance indicators

Gap to best known value	$\frac{v - v_{\text{known}}}{v_{\text{known}}}$
Normalization to randomness	$\frac{\text{mean}(v) - \text{mean}(v_{\text{random}})}{\text{variance}(v_{\text{random}})}$

Generalization results



Larger training size does not improve results



QAPLIB results

	FAQ	MCQ	DQN	RF	A2C	NGM	10k rand
bur (26-26)	**0.2**	<u>3.0</u>	14.4	4.0	3.8	3.4	3.1
chr (12-25)	**54.9**	125.3	<u>70.4</u>	87.9	74.5	121.3	130.5
els (19-19)	23.7	**<u>9.7</u>**	132.2	132.2	98.9	57.0	56.5
esc (16-128)	32.0	22.1	**<u>12.6</u>**	19.8	17.8	32.0	28.8
had (12-20)	**0.8**	8.0	9.4	4.0	<u>2.9</u>	4.4	4.5
kra (30-32)	**5.7**	<u>20.5</u>	28.9	21.7	22.0	31.4	31.4
lipa (20-90)	**2.5**	14.4	14.2	13.4	<u>12.0</u>	14.1	14.1
nug (12-30)	**3.0**	18.9	20.6	12.3	<u>12.2</u>	16.3	15.4
rou (12-20)	**3.8**	18.7	14.0	14.3	14.2	13.1	<u>11.2</u>
scr (12-20)	17.4	36.9	21.1	28.1	**<u>16.9</u>**	30.2	30.2
sko (42-100)	**1.3**	15.2	21.5	14.0	<u>11.8</u>	18.1	17.3
ste (36-36)	**7.0**	61.0	141.1	<u>51.8</u>	66.6	102.2	108.0
tai (12-150)	**7.0**	<u>18.8</u>	63.4	45.5	40.9	22.2	22.9
tho (30-150)	**2.4**	24.7	32.8	22.8	<u>19.0</u>	25.3	26.0
wil (50-100)	**0.8**	10.0	9.9	<u>6.6</u>	6.9	9.4	8.8
mean	**10.8**	<u>27.2</u>	40.4	31.9	28.0	33.4	33.9

References

[1] E. L. Lawler, "The quadratic assignment problem," *Management Science*, vol. 9, pp. 586–599, 1963.

[2] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*, 2018.

[3] E. Loiola, N. Abreu, P. Boaventura-Netto, P. Hahn, and T. Querido, "A survey of the quadratic assignment problem," *European Journal of Operational Research*, vol. 176, pp. 657–690, 01 2007.

[4] A. Nowak, S. Villar, A. S. Bandeira, and J. Bruna, "Revised note on learning algorithms for quadratic assignment with graph neural networks," 2018.

[5] C. Liu, R. Wang, Z. Jiang, J. Yan, L. Huang, and P. Lu, "Revocable deep reinforcement learning with affinity regularization for outlier-robust graph matching," 2021.

[6] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica*, vol. 25, no. 1, pp. 53–76, 1957.

[7] R. Sato, M. Yamada, and H. Kashima, "Random features strengthen graph neural networks," in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 333–341, SIAM, 2021.

[8] R. Sato, "A survey on the expressive power of graph neural networks," *ArXiv*, vol. abs/2003.04078, 2020.