Principled Data Augmentation for Learning to Solve Quadratic Programming Problems

Chendi Qian Christopher Morris

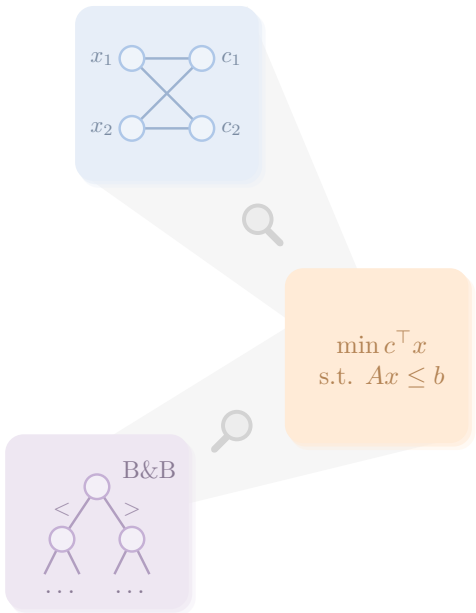
Department of Computer Science
RWTH Aachen University

NeurIPS 2025

Background

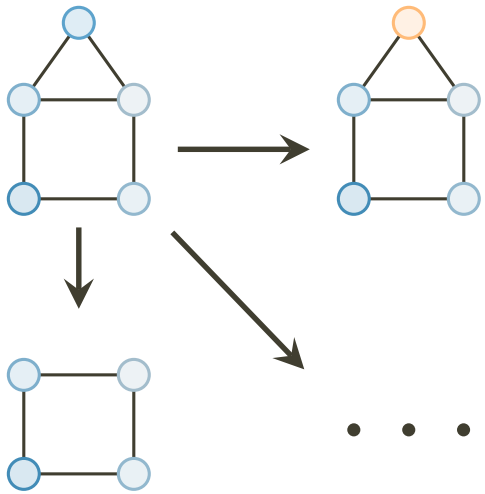
Learning to optimize (L2O)

- Represent problems
- Predict solutions
- Guide solvers
- ...



Graph data augmentation

- Feature perturbation
- Structure perturbation
- Learning-based augmentation
- ...



Background

Linearly-constrained quadratic programming (LCQP)

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & \frac{1}{2}\mathbf{x}^\intercal \mathbf{Q}\mathbf{x} + \mathbf{c}^\intercal \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{aligned}$$

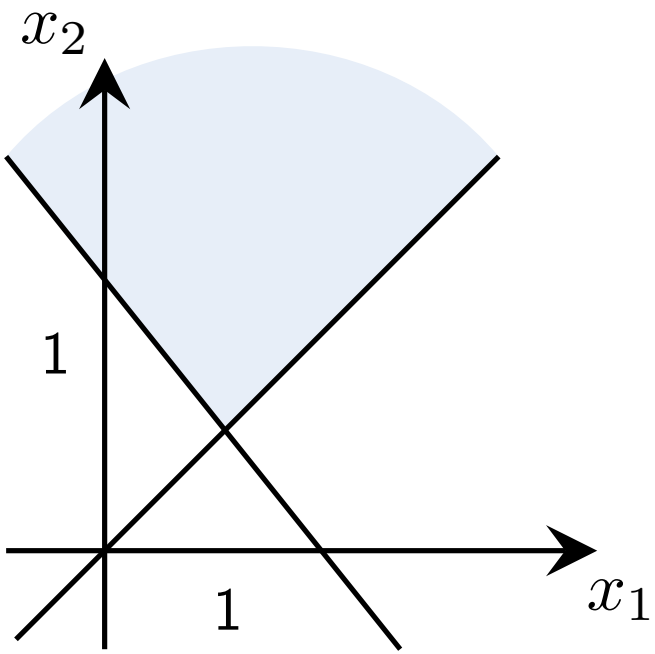
Linear programming (LP)

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & \mathbf{c}^\intercal \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{aligned}$$

Background

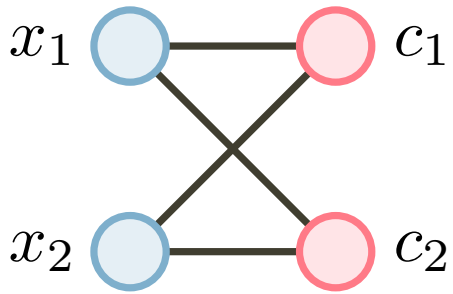
LP example

$$\begin{aligned} \min \quad & x_2 \\ s.t. \quad & c_1 : x_1 + x_2 \geq 1 \\ & c_2 : x_1 - x_2 \leq 0 \end{aligned}$$



Background

Message passing neural networks (MPNN)



$$\mathbf{h}_c^{(t)} := \mathsf{UPD}_{\mathsf{c}}^{(t)}\left(\mathbf{h}_c^{(t-1)}, \sum_{v \in N(c) \cap V(I)} A_{cv}\mathbf{h}_v^{(t-1)}\right)$$

$$\mathbf{h}_v^{(t)} := \mathsf{UPD}_{\mathsf{v}}^{(t)}\left(\mathbf{h}_v^{(t-1)}, \sum_{c \in N(v) \cap C(I)} A_{cv}\mathbf{h}_c^{(t-1)}\right)$$

Perturbation examples

1. Remove a variable: infeasible

$$\begin{aligned} \min \quad & \cancel{x_2} \\ s.t. \quad & c_1 : x_1 + \cancel{x_2} \geq 1 \\ & c_2 : x_1 - \cancel{x_2} \leq 0 \end{aligned}$$

2. Remove a constraint: unbounded

$$\begin{aligned} \min \quad & x_2 \\ s.t. \quad & c_1 : x_1 + x_2 \geq 1 \\ & c_2 : \cancel{x_1 - x_2 \leq 0} \end{aligned}$$

Motivation

Insight

Classical graph augmentation can invalidate problems!

Challenge

Can we design principled data augmentation for optimization problems?

Design principles

Expressivity

Spanning the space of feasible problem instances

Validity

Ensuring feasibility and optimality of new instances

Efficiency

Solver-free, linear time transformations

Derivation

Karush–Kuhn–Tucker (KKT) conditions

$$\begin{bmatrix} \mathbf{Q} & \mathbf{A}^\intercal \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \boldsymbol{\lambda}^* \end{bmatrix} = \begin{bmatrix} -\mathbf{c} \\ \mathbf{b} - \mathbf{s}^* \end{bmatrix}$$

Linear transformation

$$\mathbf{M} \begin{bmatrix} \mathbf{Q} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \mathbf{M}^\top \quad \mathbf{M}^{\top\dagger} \begin{bmatrix} \mathbf{x}^* \\ \lambda^* \end{bmatrix} = \mathbf{M} \begin{bmatrix} -\mathbf{c} \\ \mathbf{b} - \mathbf{s}^* \end{bmatrix}$$

Example augmentations

1

$$c_1x_1 + c_2x_2 + c_3x_3 + \cdots + c_nx_n$$
$$A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + \cdots + A_{1n}x_n \leq b_1$$
$$A_{21}x_1 + A_{22}x_2 + A_{23}x_3 + \cdots + A_{2n}x_n \leq b_2$$
$$\vdots$$
$$A_{m1}x_1 + A_{m2}x_2 + A_{m3}x_3 + \cdots + A_{mn}x_n \leq b_m$$
$$A_{m+1,1}x_1 + A_{m+1,2}x_2 + A_{m+1,3}x_3 + \cdots \leq b_{m+1}\ \alpha + \beta$$

Adding redundant constraints

2

$$c_1x_1 + c_2x_2 + c_3x_3 + \cdots + c_nx_n$$
$$A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + \cdots + A_{1n}x_n \leq b_1$$
$$\cancel{A_{21}x_1 + A_{22}x_2 + A_{23}x_3 + \cdots + A_{2n}x_n \leq b_2}$$
$$\vdots$$
$$\cancel{A_{m1}x_1 + A_{m2}x_2 + A_{m3}x_3 + \cdots + A_{mn}x_n \leq b_m}$$

Removing inactive constraints

3

$$\begin{aligned} c_1x_1 + c_2x_2 + \ &\ldots + c_nx_n + 0\,x_{n+1} \\ A_{11}x_1 + A_{12}x_2 + \ &\ldots + A_{1n}x_n + A_{1,n+1}x_{n+1} \leq b_1 \\ A_{21}x_1 + A_{22}x_2 + \ &\ldots + A_{2n}x_n + A_{2,n+1}x_{n+1} \leq b_2 \\ &\vdots \\ A_{m1}x_1 + A_{m2}x_2 + &\ldots + A_{mn}x_n + A_{m,n+1}x_{n+1} \leq b_m \end{aligned}$$

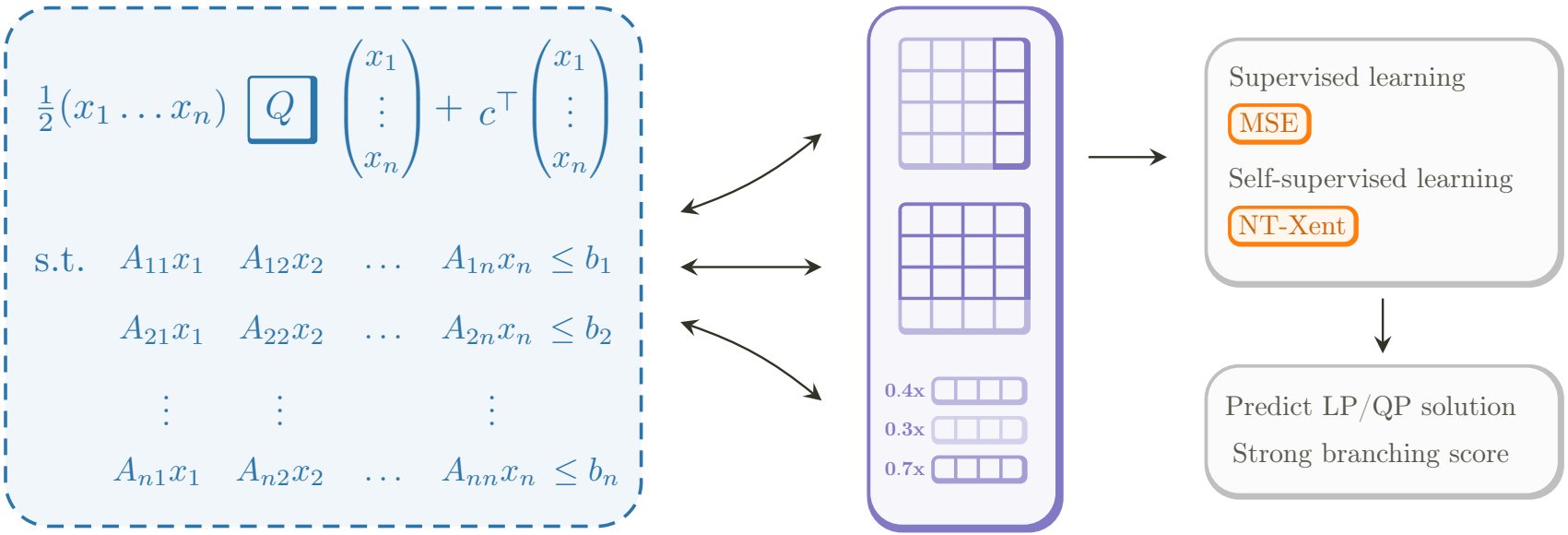
Adding dummy variables

4

$$c_1x_1 + c_2x_2 + c_3x_3 + \cdots + c_nx_n$$
$$\alpha_1 \cdot (A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + \cdots + A_{1n}x_n) \leq \alpha_1 b_1$$
$$\alpha_2 \cdot (A_{21}x_1 + A_{22}x_2 + A_{23}x_3 + \cdots + A_{2n}x_n) \leq \alpha_2 b_2$$
$$\vdots$$
$$\alpha_m \cdot (A_{m1}x_1 + A_{m2}x_2 + A_{m3}x_3 + \cdots + A_{mn}x_n) \leq \alpha_m b_m$$

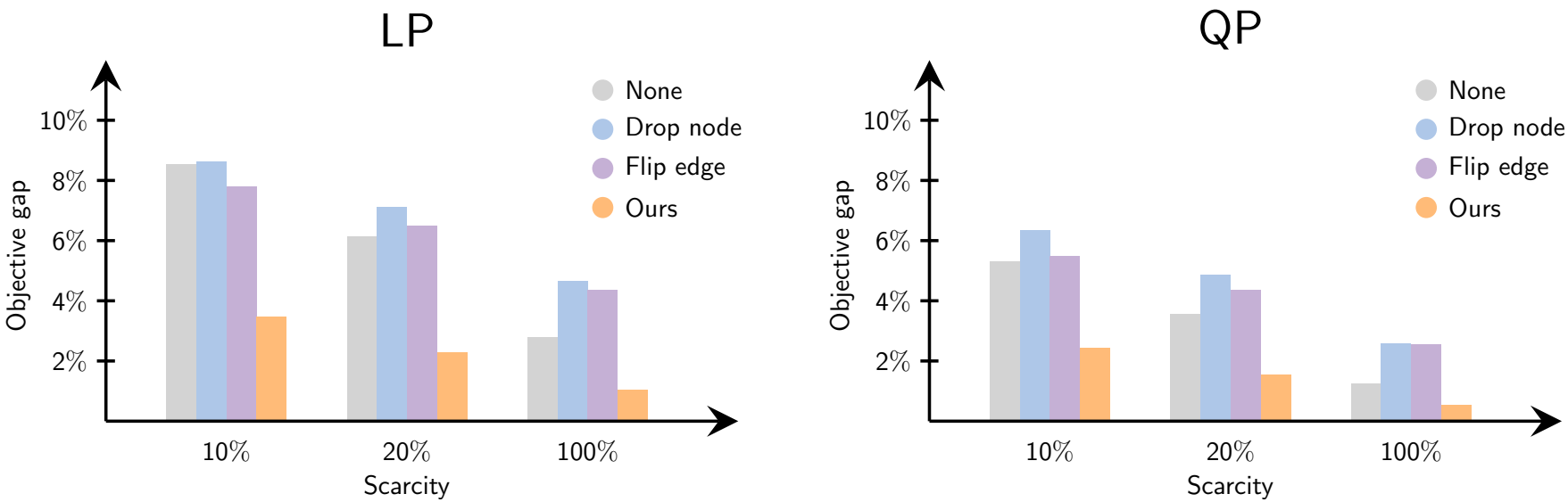
Scaling constraints

Deployment


$\frac{1}{2}(x_1 \dots x_n)$ Q $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} + c^\top \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$
s.t. $A_{11}x_1$ $A_{12}x_2$ … $A_{1n}x_n \le b_1$
$A_{21}x_1$ $A_{22}x_2$ … $A_{2n}x_n \le b_2$
$A_{n1}x_1$ $A_{n2}x_2$ … $A_{nn}x_n \le b_n$
0.4x
0.3x
0.7x
Supervised learning
MSE
Self-supervised learning
NT-Xent
Predict LP/QP solution
Strong branching score

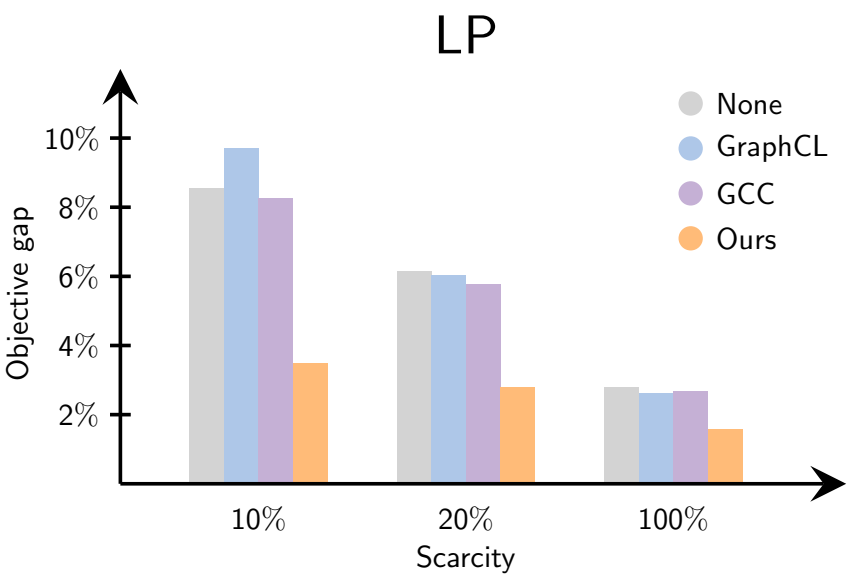
Supervised learning

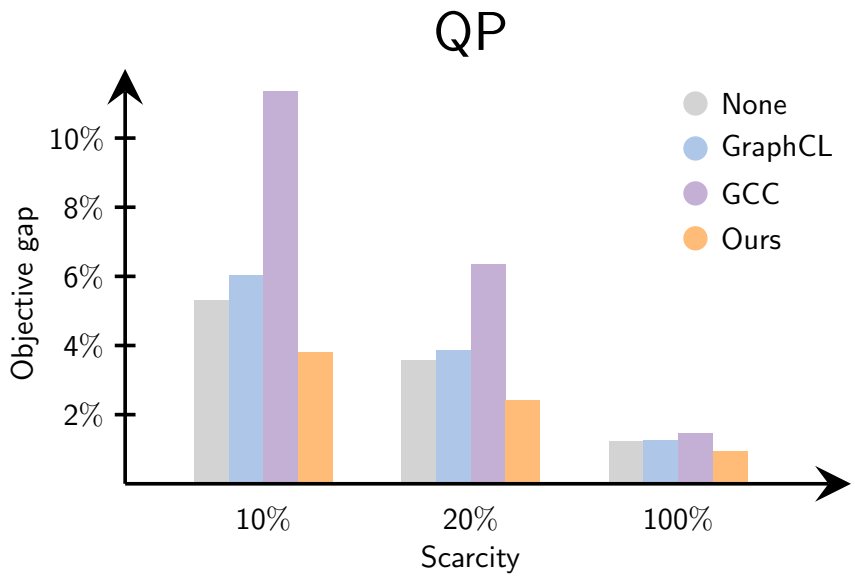
- 10000 **labeled** LP / QP instances, $8:1:1$ split
- Different level of data scarcity: $x\%$ of all training data
- Compared with classical data augmentation methods



Contrastive learning

- 10000 **unlabeled** LP / QP instances
- Contrastive learning using NT-Xent loss
- Supervised finetuning on $x\%$ of labeled training data
- Compared with graph representation learning methods





Conclusion

Contribution

- Novel Connection between data augmentation and L2O
- Principled, theory-guided data augmentation methods
- Strong empirical results & generalization performance

Limitation

- Theoretical generalization gain unknown
- Trade-off between diversity and efficiency