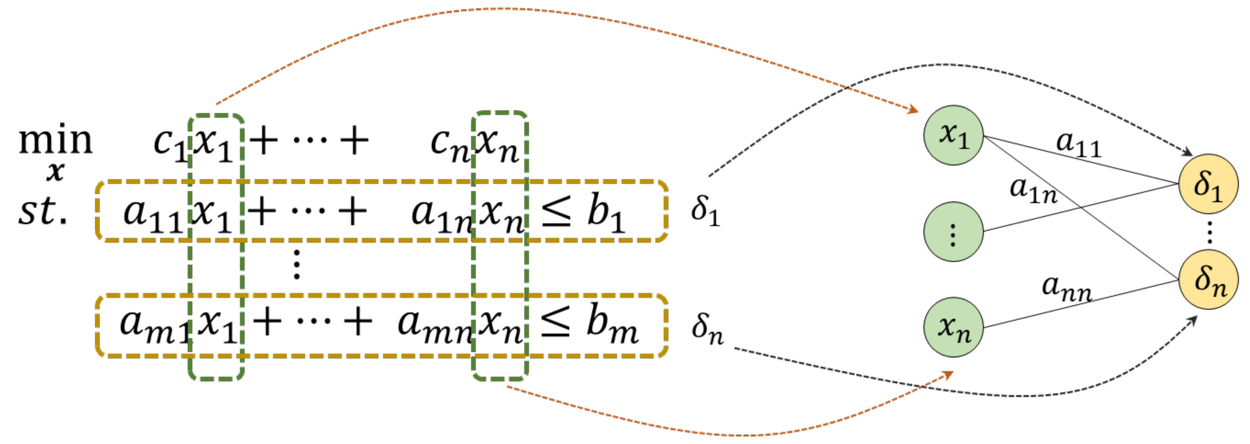
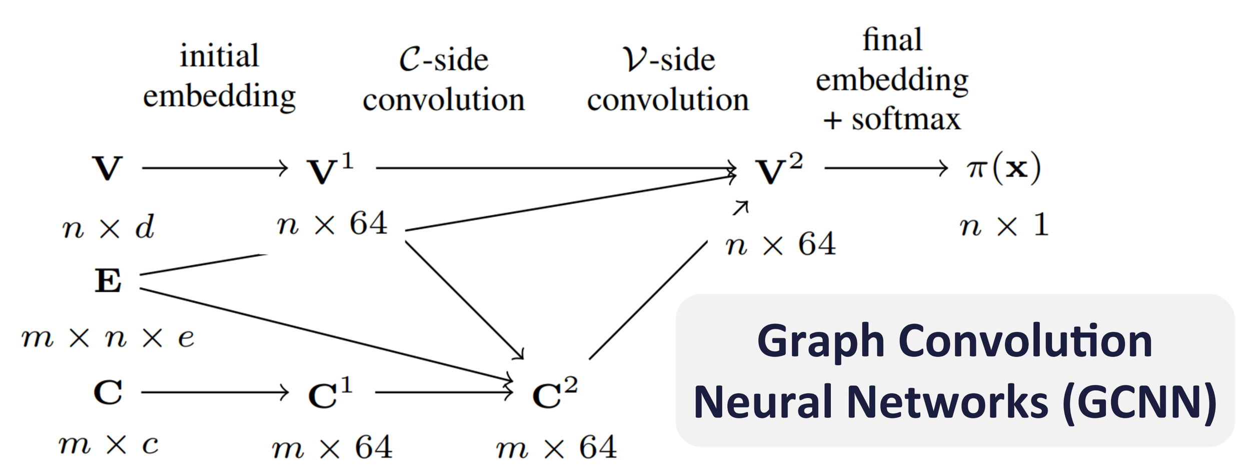
CAMBranch: Contrastive Learning with Augmented MILPs for Branching

Jiacheng Lin*[1], Meng Xu*[2], Zhihua Xiong[2], Huangang Wang+[2]
[1] University of Illinois Urbana-Champaign [2] Tsinghua University
* Equal Contribution + Corresponding Author

Paper

Machine Learning can accelerate Mixed Integer Linear Programming (MILP) solving.





Model	Easy Time	Easy Wins	Easy Nodes	Medium Time	Medium Wins	Medium Nodes	Hard Time	Hard Wins	Hard Nodes
FSB	34.94	0/100	54	242.51	0/100	114	995.40	0/82	84
RPB	30.63	9/100	79	177.25	2/100	196	830.90	2/93	178
GCNN	24.72	25/100	169	145.17	13/100	405	680.78	5/95	449

Capacitated Facility Location

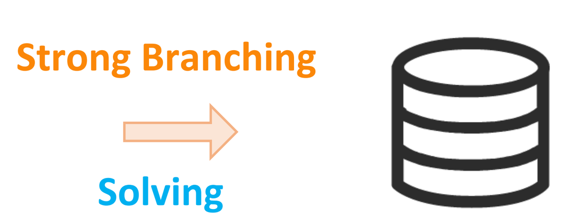
(Gasse et al., 2019)

Solving Time is decreased by ~150 seconds for *Hard* level MILP instances.

Collecting experts for Imitation Learning is computationally intensive and time-consuming.

Expert Data Collection: Solving training MILP instances with Strong Branching (expert strategy).

$$\min_{x} \ c^{\top}x \quad \text{s.t. } Ax \leqslant b, \quad l \leqslant x \leqslant u, \quad x_j \in \mathbb{Z}, \ \forall j \in \mathcal{I}$$



Collecting 100k expert samples (*Easy* Level) instances requires

- **26.65 hours for *Set Covering Problem***
- **12.48 hours for *Combinatorial Auction Problem***
- **84.79 hours for *Capacitated Facility Location Problem***
- **53.45 hours for *Maximum Independent Set Problem***

As the complexity of MILPs scales up in practical

The expert data collecting time dramatically increases

Our Solution: Augmented MILPs

$$\min_{x} \ c^{\top}x \quad \text{s.t. } Ax \leqslant b, \quad l \leqslant x \leqslant u, \quad x_j \in \mathbb{Z}, \ \forall j \in \mathcal{I}$$

$\hat{x} = x + s$; $s_i \in \mathbb{Z}$ if $x_i \in \mathbb{Z}$, else $s_i \in \mathbb{R}$ →

$$\min_{\hat{x}} \ c^{\top}\hat{x} - c^{\top}s \quad \text{s.t. } A\hat{x} \leqslant As + b, \quad l + s \leqslant \hat{x} \leqslant u + s, \quad \hat{x}_j \in \mathbb{Z}, \ \forall j \in \mathcal{I}$$

$\hat{b} = As + b$, $\hat{l} = l + s$, $\hat{u} = u + s$ →

$$\min_{\hat{x}} \ c^{\top}\hat{x} - c^{\top}s \quad \text{s.t. } A\hat{x} \leqslant \hat{b}, \quad \hat{l} \leqslant \hat{x} \leqslant \hat{u}, \quad \hat{x}_j \in \mathbb{Z}, \ \forall j \in \mathcal{I}$$

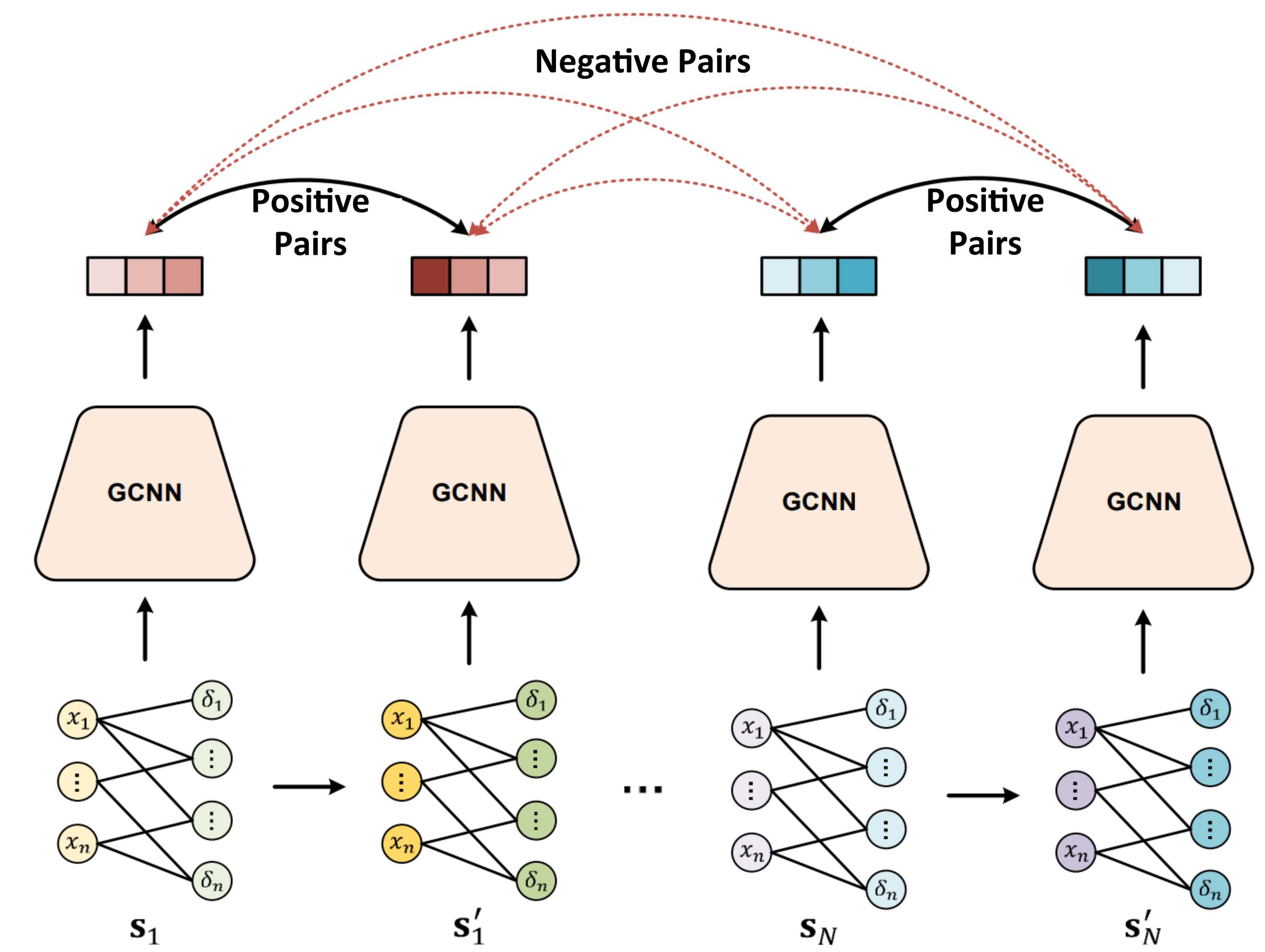
Characteristics of Augmented MILPs (AMILPs)

- **Each MILP can generate multiple AMILPs**
- **Generated AMILPs share identical Variable Selection Decisions to its original MILPs**

Generate labeled expert data without solving AMILPs

Next, by lemmas and theorems, obtain AMILP's Bipartite Graph features which will be fed into the GCNN for feature extraction.

Leveraging Contrastive Learning between MILPs and AMILPs



Original Bipartite Graph $(\mathcal{G}_{\text{ori}}, \mathbf{C}_{\text{ori}}, \mathbf{E}_{\text{ori}}, \mathbf{V}_{\text{ori}})$
Augmented Bipartite Graph $(\mathcal{G}_{\text{aug}}, \mathbf{C}_{\text{aug}}, \mathbf{E}_{\text{aug}}, \mathbf{V}_{\text{aug}})$

$$c^{\mathcal{G}}_{\text{ori}} = \text{MLP}(\text{Concat}(\text{MaxPool}(\mathbf{C}'_{\text{ori}}), \text{MeanPool}(\mathbf{C}'_{\text{ori}})))$$
$$v^{\mathcal{G}}_{\text{ori}} = \text{MLP}(\text{Concat}(\text{MaxPool}(\mathbf{V}'_{\text{ori}}), \text{MeanPool}(\mathbf{V}'_{\text{ori}})))$$
$$c^{\mathcal{G}}_{\text{aug}} = \text{MLP}(\text{Concat}(\text{MaxPool}(\mathbf{C}'_{\text{aug}}), \text{MeanPool}(\mathbf{C}'_{\text{aug}})))$$
$$v^{\mathcal{G}}_{\text{aug}} = \text{MLP}(\text{Concat}(\text{MaxPool}(\mathbf{V}'_{\text{aug}}), \text{MeanPool}(\mathbf{V}'_{\text{aug}})))$$

Variable and Constraint Node Pooling

$$g_{\text{ori}} = \text{MLP}(\text{Concat}(c^{\mathcal{G}}_{\text{ori}}, v^{\mathcal{G}}_{\text{ori}}))$$
$$g_{\text{aug}} = \text{MLP}(\text{Concat}(c^{\mathcal{G}}_{\text{aug}}, v^{\mathcal{G}}_{\text{aug}}))$$

Variable and Constraint Node Feature Merging

Loss function

1. Imitation Learning

$$\mathcal{L}_{\text{sup}} = -\frac{1}{N}\sum_{(\mathbf{s}_i, \mathbf{a}_i^*) \in D} \log \pi_\theta(\mathbf{a}_i^* | \mathbf{s}_i)$$

2. Contrastive Learning

$$\mathcal{L}^{(\text{infoNCE})} = -\sum_{i=1}^{n_{\text{batch}}} \log\left(\frac{\exp\left(\tilde{g}_{\text{ori}}^{\mathrm{T}}(i) \cdot \tilde{g}_{\text{aug}}(i)\right)}{\sum_{j=1}^{n_{\text{batch}}} \exp\left(\tilde{g}_{\text{ori}}^{\mathrm{T}}(i) \cdot \tilde{g}_{\text{aug}}(j)\right)}\right)$$

3. Consistency Loss

$$\mathcal{L}^{(\text{Aux})} = \sum_{i=1}^{n_{\text{batch}}} \left(P_{\text{ori}}(i) - P_{\text{aug}}(i)\right)^2$$

The final loss function is $\mathcal{L} = \mathcal{L}^{(\text{sup})} + \lambda_1 \mathcal{L}^{(\text{infoNCE})} + \lambda_2 \mathcal{L}^{(\text{Aux})}$

Experimental Results

Model	Easy Time	Easy Wins	Easy Nodes	Medium Time	Medium Wins	Medium Nodes	Hard Time	Hard Wins	Hard Nodes
FSB	4.71	0/100	10	97.6	0/100	90	1396.62	0/64	381
RPB	2.61	1/100	21	19.68	2/100	713	142.52	29/100	8971
GCNN	1.96	43/100	87	11.30	74/100	695	158.81	19/94	12089
GCNN (10%)	1.99	44/100	102	12.38	16/100	787	144.40	10/100	10031
CAMBranch (10%)	2.03	12/100	91	12.68	8/100	758	**131.79**	**42/100**	**9074**

Combinatorial Auction

Model	Easy Time	Easy Wins	Easy Nodes	Medium Time	Medium Wins	Medium Nodes	Hard Time	Hard Wins	Hard Nodes
FSB	34.94	0/100	54	242.51	0/100	114	995.40	0/82	84
RPB	30.63	9/100	79	177.25	2/100	196	830.90	2/93	178
GCNN	24.72	25/100	169	145.17	13/100	405	680.78	5/95	449
GCNN (10%)	26.30	15/100	180	124.49	48/100	406	672.88	11/95	423
CAMBranch (10%)	24.91	**50/100**	183	**124.36**	37/100	**390**	**470.83**	**77/95**	428

Capacitated Facility Location

Model	Easy Time	Easy Wins	Easy Nodes	Medium Time	Medium Wins	Medium Nodes	Hard Time	Hard Wins	Hard Nodes
FSB	28.85	10/100	19	1219.15	0/62	81	3600.00	-	-
RPB	10.73	11/100	78	133.30	5/100	2917	965.67	10/40	17019
GCNN	7.17	11/100	90	164.51	4/99	5041	1020.58	0/17	21925
GCNN (10%)	7.18	26/100	103	122.65	8/89	3711	695.96	2/20	17034
CAMBranch (10%)	**6.92**	**42/100**	**90**	**61.51**	**83/100**	**1479**	**496.86**	**33/40**	**10828**

Maximum Independent Set

Our CAMBranch, trained with only 10% of the full dataset outperforms GCNN trained with full data.

Table 3: The results of evaluating the instance-solving performance for the Combinatorial Auction problem by utilizing the complete training dataset. Bold numbers denote the best results.

Model	Easy Time	Easy Wins	Easy Nodes	Medium Time	Medium Wins	Medium Nodes	Hard Time	Hard Wins	Hard Nodes
GCNN (10%)	1.99	2/100	102	12.38	3/100	787	144.40	2/100	10031
GCNN (100%)	1.96	4/100	**87**	11.30	7/100	695	158.81	4/94	12089
CAMBranch (10%)	2.03	1/100	91	12.68	2/100	758	131.79	11/100	9074
CAMBranch (100%)	**1.73**	**93/100**	88	**10.04**	**88/100**	**690**	**109.96**	**83/100**	**8260**

CAMBranch is not limited to data-limited scenarios, which also serves a valuable tool for data argumentation with full dataset.