Learning to Condition (L2C)

A Neural Heuristic for Scalable MPE Inference

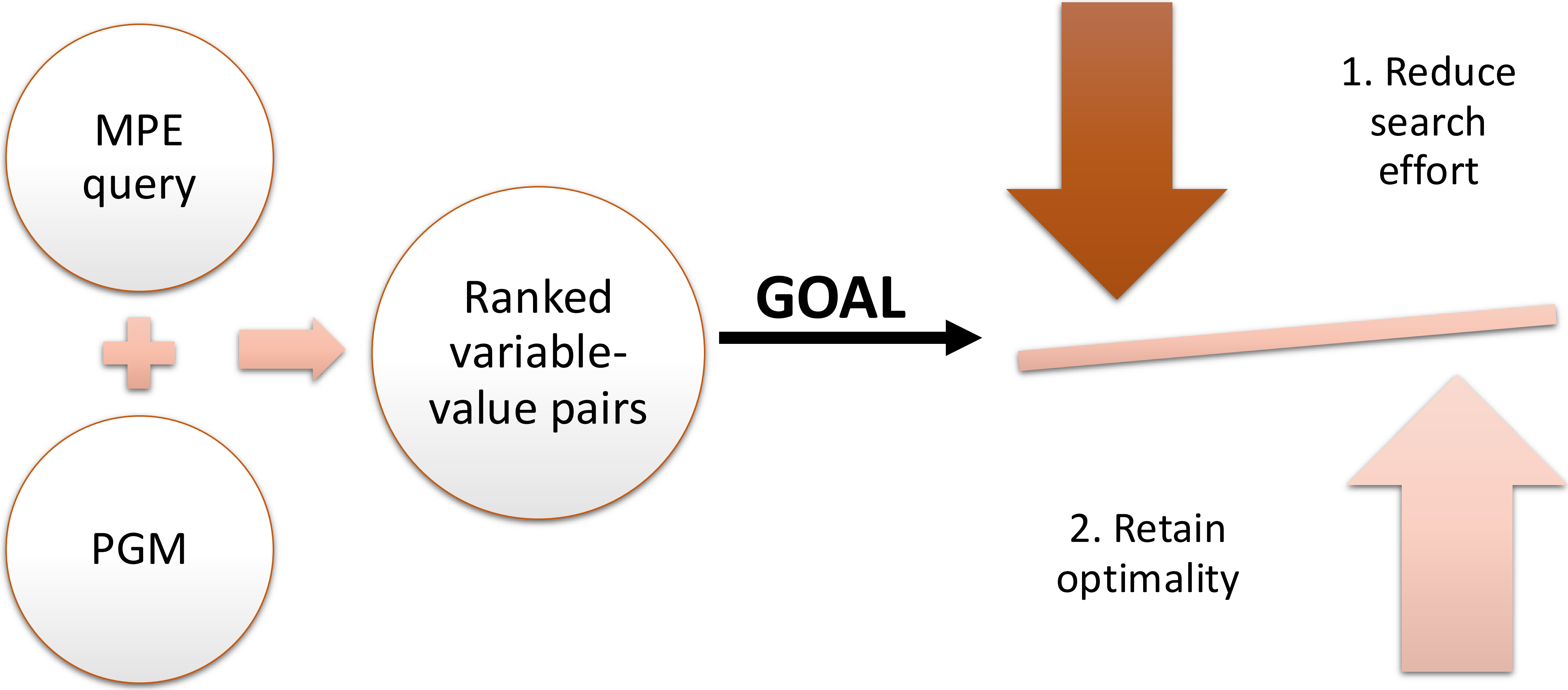
Brij Malhotra*, Shivvrat Arya+, Tahrima Rahman*, Vibhav Gogate*

UTD*, NJIT+

Classical Heuristics have **limited** scalability

- MPE goal: $$\mathrm{MPE}(Q,e) = \arg\max_q P_M(q \mid e) = \arg\max_q \sum_{f\in\mathcal{F}} f((q,e)_{S(f)})$$
- Traditional variable value ordering heuristics, used with Branch & Bound solvers for MPE inference such as min-fill or max-degree or most constraining value, often **fail** to generalize across diverse structures.
- Poor variable selection can significantly slow down inference and compromise solution quality under time constraints.
- There is a need to develop efficient and scalable heuristics that can be easily integrated with the exact solvers

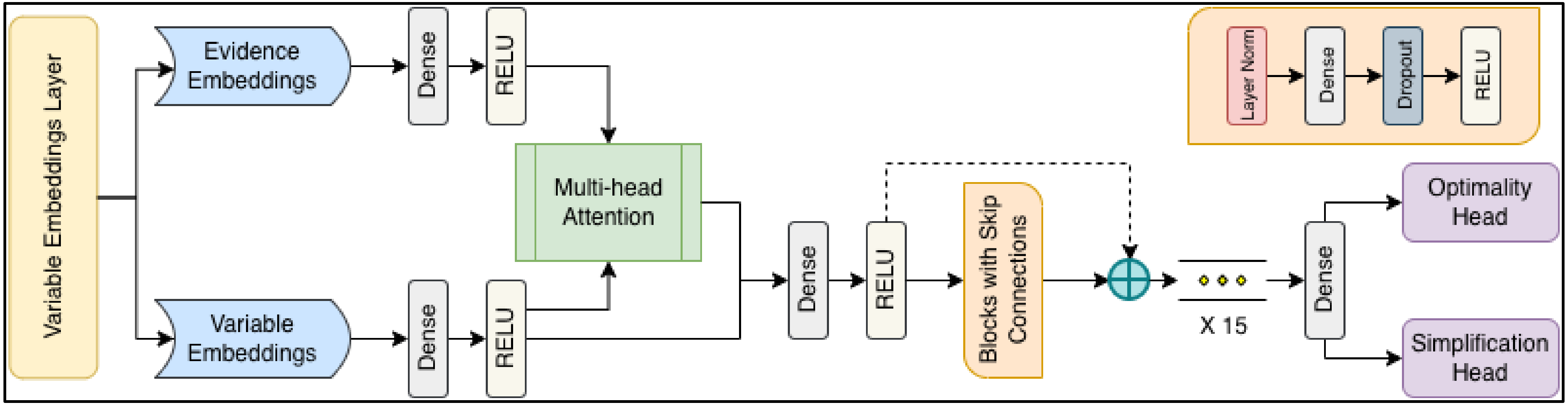
L2C Overview



Data Generation

- Generate data for two objectives – optimality preservation and simplifying inference
- Start by solving the large set of MPE query for optimal assignments
  - Train the neural network to output probability of optimality over variable assignments
- Using optimal assignments, solve for conditioned query
  - Record solver statistics upon conditioning
  - Time and search nodes required to solve the query and the log-likelihood values achieved under time constraints
  - Train neural network to rank variable assignments according to the joint distribution of the three scores.

Architecture Diagram



Multi-task Training

1. Optimality head is trained with BCE loss for every assignment:

$$L_{\text{opt}} = -\sum_{Q_i \in Q} \sum_{q \in \{0,1\}} \left[y_i^{(q)} \log \hat{y}_i^{(q)} + (1 - y_i^{(q)}) \log\left(1 - \hat{y}_i^{(q)}\right) \right]$$

2. Simplification head is trained with softmax ranking of solver statistics:

$$L_{\text{rank}} = -\sum_{C \in \mathcal{C}} p_C \log \hat{p}_C$$

- **Overall loss** is a convex combination of the two losses.

$$L = \lambda_{opt} \cdot L_{opt} + \lambda_{rank} \cdot L_{rank}$$

Inference-Time Strategies

Greedy Conditioning

Select high-confidence variable–value pairs above threshold τ.

Beam Search

Expand **W sequences** using L2C scores; best final sequence defines evidence E^*.

NN-Guided B&B

Choose variables that **recursively simplify subproblems** & tighten bounds.

Results - Conditioning Impact

$$\mathcal{LL}\ \ gap: \frac{1}{N}\sum_{i=1}^{N}\frac{\mathcal{LL}^{S}_{(i)}-\mathcal{LL}^{D}_{(i)}}{|\mathcal{LL}^{S}_{(i)}|}\times 100$$

where $\mathcal{LL}^{S}_{(i)}$ and $\mathcal{LL}^{D}_{(i)}$ are the log-likelihood score with and without conditioning

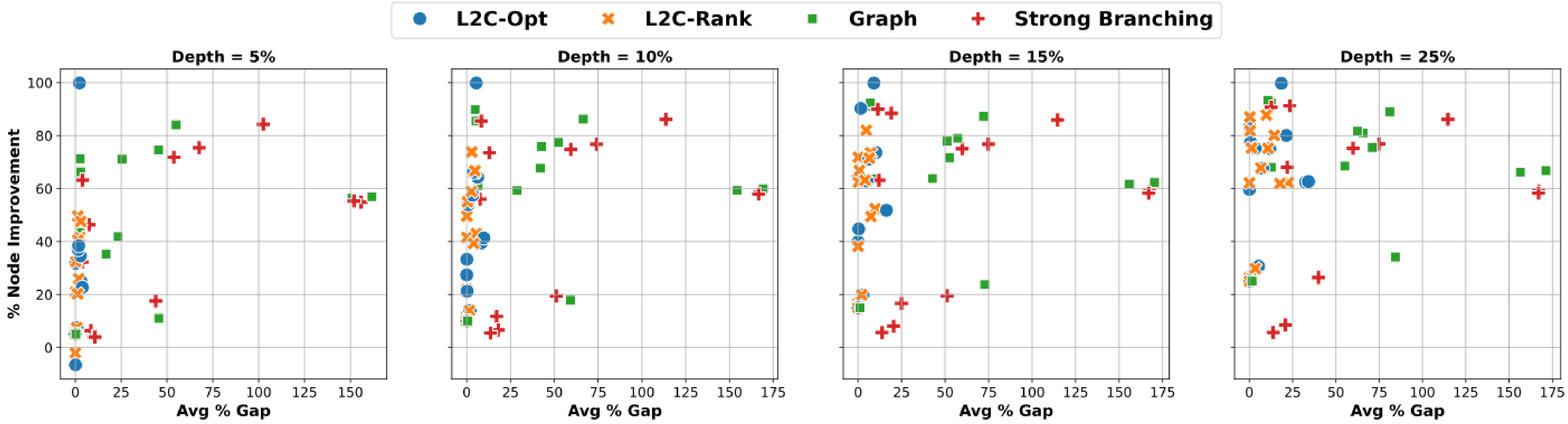
	G	SB	LO	LR
BN 12	5	0	12	12
BN 13	5	1	11	12
BN 30	12		12	12
BN 32	9		12	12
BN 45	0	0	12	12
BN 49	0	0	11	12
BN 53	0	0	10	12
BN 59	0	0	12	12
BN 61	0	0	7	12
BN 65	0	0	10	10
BN 9	2	0	12	12
Prm 60	5	0	12	12
Prm 68	2	0	6	11
Grid 20	0	0	12	12

	5%	10%	15%	20%
BN 12	-2.0	-3.8	-4.9	-5.2
BN 13	-1.0	-2.1	-3.1	-4.1
BN 30	-11.6	-20.1	-25.3	-33.1
BN 32	-63.7	-66.7	-69.2	-72.1
BN 45	-2.0	-2.8	-3.3	-3.5
BN 49	-6.6	-14.0	-15.7	-11.3
BN 53	-2.6	-8.1	-12.0	-10.5
BN 59	-3.2	-9.8	-13.9	-15.2
BN 61	-2.8	-6.8	-13.2	-10.7
BN 65	-0.9	-4.1	-6.2	-6.6
BN 9	-0.4	-0.8	-1.5	-2.9
Prm 60	-0.1	-0.2	-0.3	-0.4
Prm 68	-0.1	-0.1	-0.2	-0.2
Grid 20	-0.2	-0.3	-0.4	-0.6

Fig: (L) Heatmap of wins L2C > classical in most settings & (R) LL gain

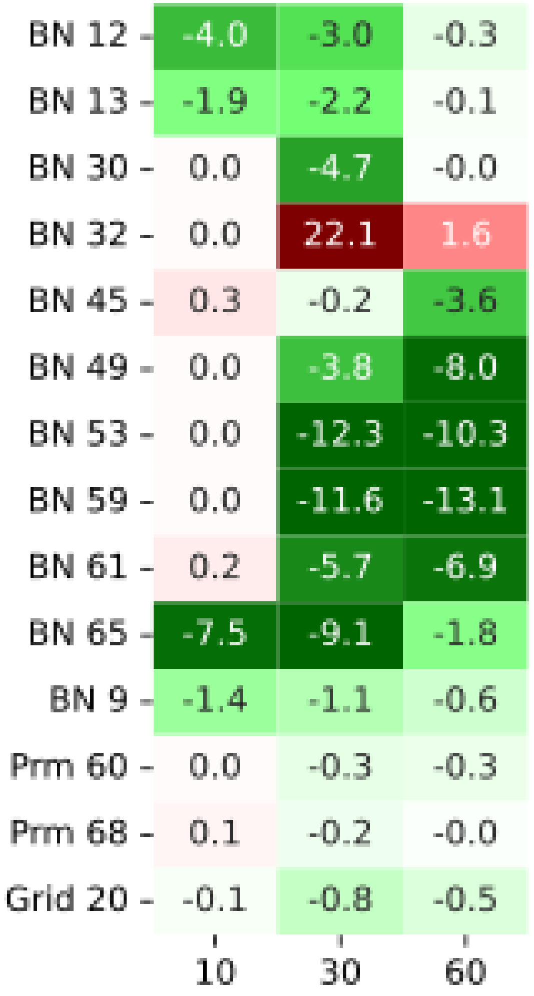
Results - AOBB Node Reduction

- Node reduction vs. LL gap
- L2C achieves large gains with minimal quality loss



Results - B&B Guidance

- When used within SCIP, L2C-guided node selection and branching decisions yield better anytime performance.
- In the figure shown, we compare the log-likelihood gain achieved with L2C as compared to SCIP's default heuristics.



Additional Results

Read the ***Paper***
HERE

Find the ***Code***
HERE