Area-Driven FPGA Logic Synthesis Using Reinforcement Learning

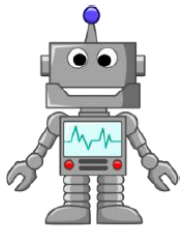
Guanglei Zhou and Jason Anderson

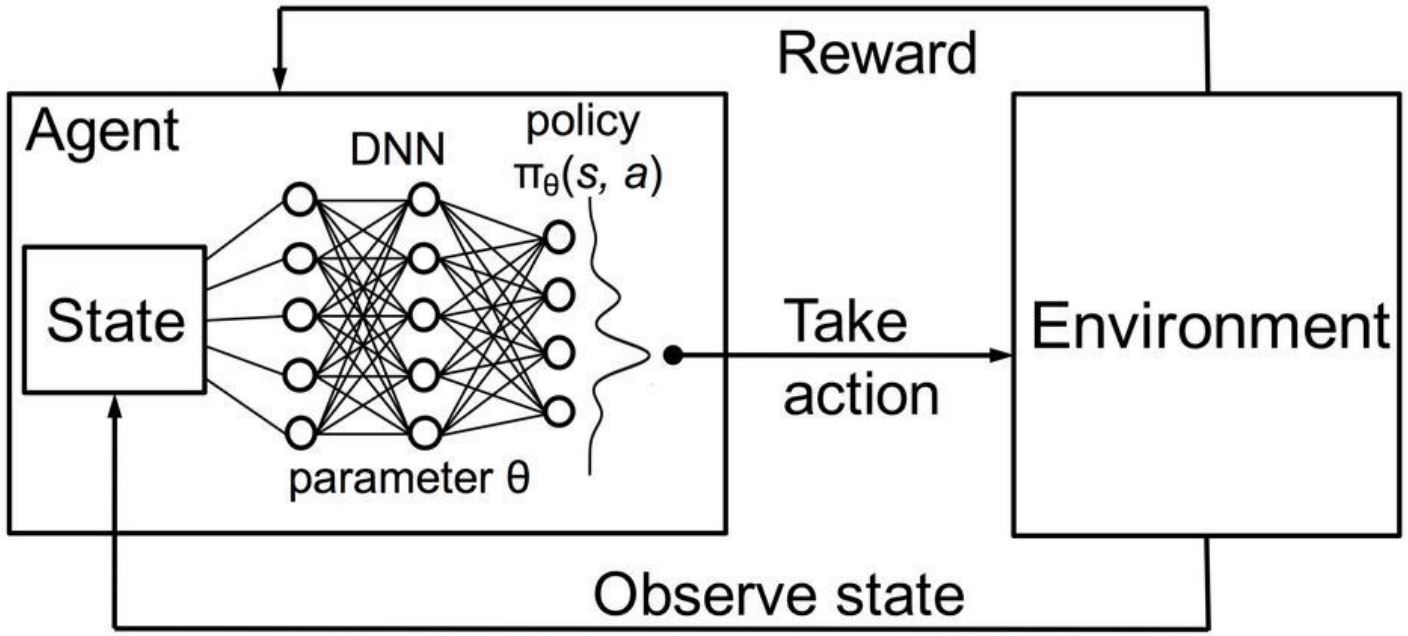
Department of ECE, University of Toronto

28th Asia and South Pacific Design Automation Conference

January 17th, 2023

Reinforcement Learning-Based Logic Synthesis



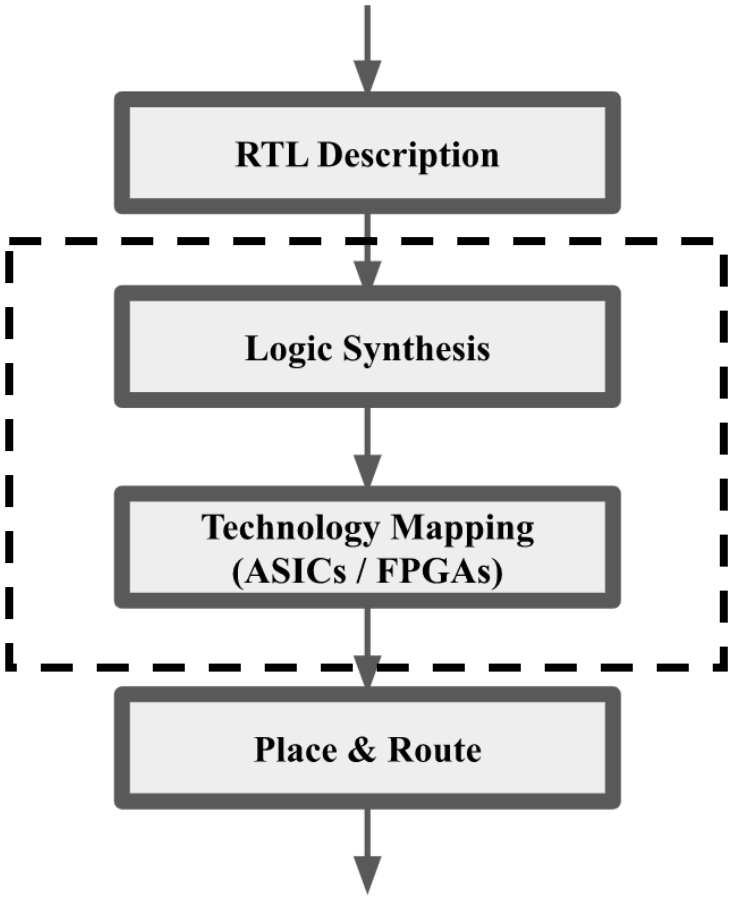


Logic Synthesis CAD Tool

How to use RL to better optimize RTL circuits?

UNIVERSITY OF TORONTO

Motivation



- Logic synthesis involves a rich set of optimization commands
- The order of such commands can lead to different results (a.k.a. ordering problem)
- Fixed recipe is often used for various circuits
  ✖ non-optimal solution
- Benefit of having a bespoke sequence for different circuits is promising

UNIVERSITY OF TORONTO

Prior Work

- Limitation for existing approaches[1,2]: trained and tested on the same individual circuits.
  - Hard to distinguish whether gains are just memorization or actual learning is happening

[1]K. Zhu, M. Liu, H. Chen, Z. Zhao, and D. Z. Pan, “Exploring Logic Optimizations with Reinforcement Learning and Graph Convolutional Network,” in Proc. of ACM/IEEE Workshop on Machine Learning for CAD, 2020, pp. 145-150

[2]A. Hosny, S. Hashemi, M. Shalan and S. Reda, "DRiLLS: Deep Reinforcement Learning for Logic Synthesis," 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), 2020, pp. 581-586

Prior Work

- Limitation for existing approaches[1,2]: trained and tested on the same individual circuits.
  - Hard to distinguish whether gains are just memorization or actual learning is happening

Our Improvements

- Enhance the observable state that RL “sees”
- Conduct a feature-importance analysis using a random forest classifier
  - Able to show the importance of added features and prune the feature space by 40%
- Outperform a commonly used ABC heuristic recipe (resyn2) and is able to generalize trained experience: separate training and validation circuit sets.

[1]K. Zhu, M. Liu, H. Chen, Z. Zhao, and D. Z. Pan, “Exploring Logic Optimizations with Reinforcement Learning and Graph Convolutional Network,” in Proc. of ACM/IEEE Workshop on Machine Learning for CAD, 2020, pp. 145-150

[2]A. Hosny, S. Hashemi, M. Shalan and S. Reda, "DRiLLS: Deep Reinforcement Learning for Logic Synthesis," 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), 2020, pp. 581-586

Observation State Enhancement

- Extract more features from input circuits using a previous study[1].
  - Enhance state from 8 dimensions (as used in DRiLLS[2]) to 21 dimensions

# of nodes	# of Edges	# of levels	% of ANDs
# of Latches	# of Primary Input	# of Primary Output	% of NOTs

Lower four features will not change

[1]M. Hutton, J. Rose, J. Grossman, and D. Corneil, “Characterization and parameterized generation of synthetic combinational benchmark circuits,” IEEE Trans. on Computer-Aided Design of Integrated Circuits and Syst., vol. 17, no. 10, pp. 985–996, 1998.
[2]A. Hosny, S. Hashemi, M. Shalan and S. Reda, "DRiLLS: Deep Reinforcement Learning for Logic Synthesis," ASP-DAC, 2020, pp. 581-586

UNIVERSITY OF TORONTO

Observation State Enhancement

- Extract more features from input circuits using a previous study[1].
  - Enhance state from 8 dimensions (as used in DRiLLS[2]) to 21 dimensions

# of nodes	# of Edges	# of levels	% of ANDs
# of Latches	# of Primary Input	# of Primary Output	% of NOTs

Lower four features will not change

1	# of Nodes	12	Primary output shape based on the delay level
2	# of Edges	13	Average of node fan-out
3	# of Flip Flops	14	Standard deviation of node fan-out
4	# of combinational nodes	15	Average of combinational node fan-out
5	# of LUTs in the mapped FPGA	16	Standard deviation of combinational node fan-out
6	# of Levels in the mapped FPGA	17	Average of pi node fan-out
7	Node shape based on delay level	18	Standard deviation of pi node fan-out
8	Input shape based on the delay level	19	Average of D-type flip flop node fan-out
9	Output shape based on the delay level	20	Standard deviation of D-type flip flop node fan-out
10	Latches shape based on the delay level	21	Reconvergence value
11	Edge length distribution shape based on the delay level		

Circuit characteristics used by RL agent

[1]M. Hutton, J. Rose, J. Grossman, and D. Corneil, “Characterization and parameterized generation of synthetic combinational benchmark circuits,” IEEE Trans. on Computer-Aided Design of Integrated Circuits and Syst., vol. 17, no. 10, pp. 985–996, 1998.

[2]A. Hosny, S. Hashemi, M. Shalan and S. Reda, "DRiLLS: Deep Reinforcement Learning for Logic Synthesis," ASP-DAC, 2020, pp. 581-586

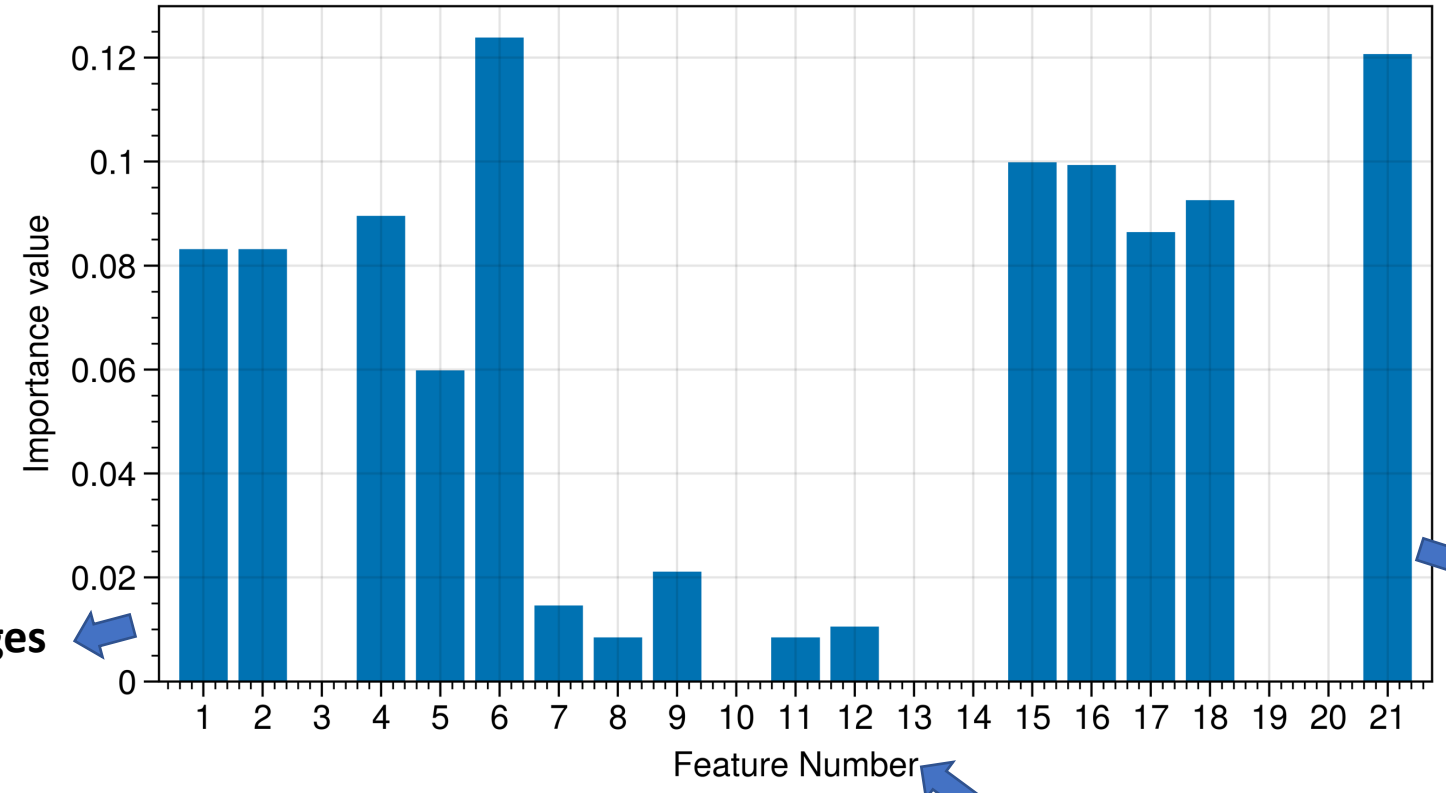
Feature Analysis

- Feature-importance analysis using a random forest classifier

1. Generate 10k data points of [observations, reward]

2. Random forest classifier

3. Feature-importance Value



# of Nodes/ Edges

Reconvergence Value

Please refer to slide 7 table

UNIVERSITY OF TORONTO

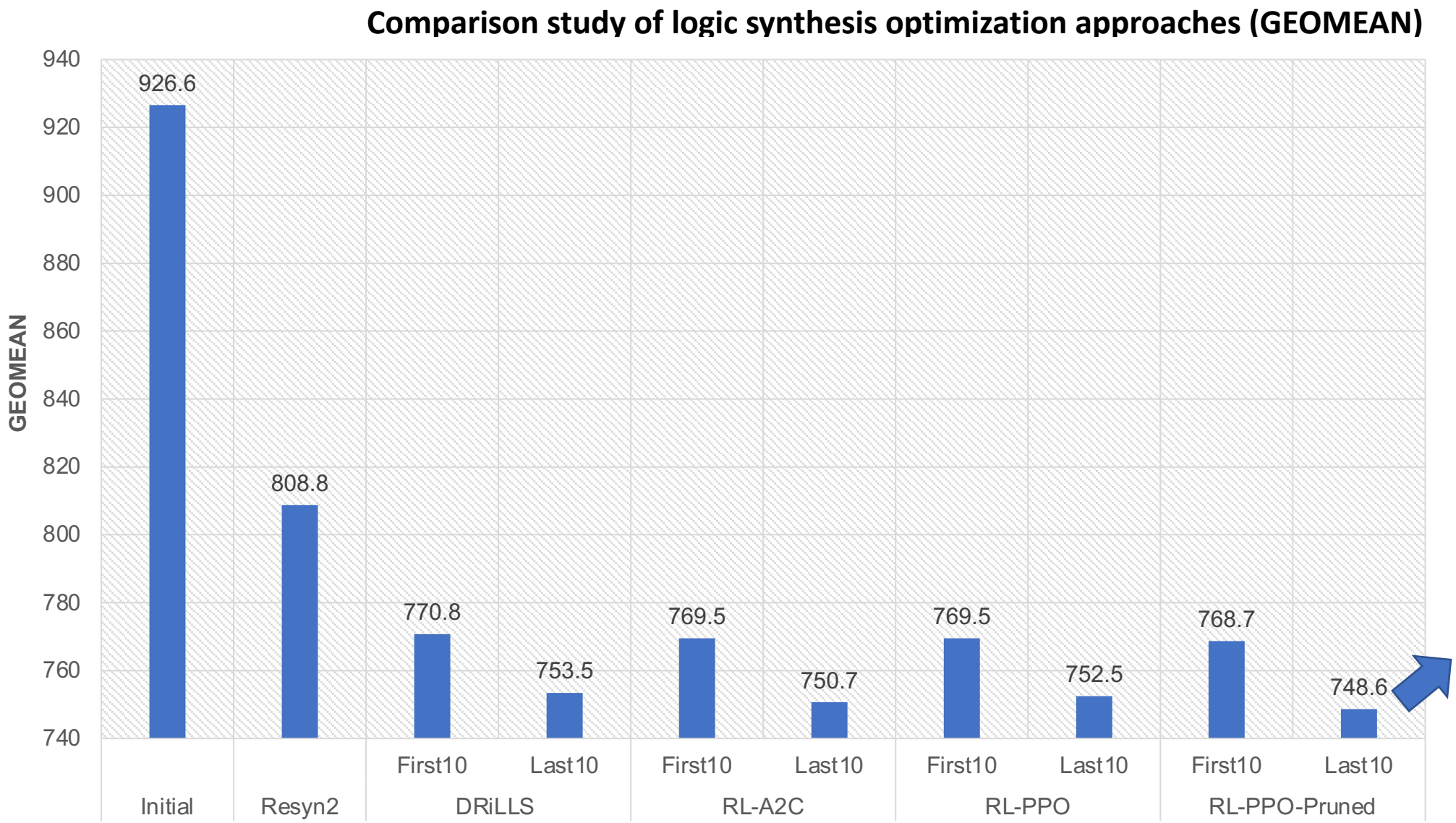
Experimental Setup: Circuit-by-Circuit Style

- Circuit-by-circuit (CBC) style:
  - One RL agent for one benchmark, train and validate using the same circuit from EPFL benchmark
- ABC for logic synthesis optimization
  - Mapped to 6-LUTs using the ABC priority cuts mapper (command: if -a -K 6), targeting area reduction
- Reinforcement learning setting
  - Action space: [balance, rewrite, rewrite -z, resub, resub -z, refactor, refactor-z , undo]
  - State space: 21 features extracted from Ccirc[1] + action history of these 8 commands
- RL algorithm evaluated: PPO, A2C (also used in DRiLLS[1])
- Train for 5k samples

UNIVERSITY OF TORONTO

[1]A. Hosny, S. Hashemi, M. Shalan and S. Reda, "DRiLLS: Deep Reinforcement Learning for Logic Synthesis," ASP-DAC, 2020, pp. 581-586

Training Performance

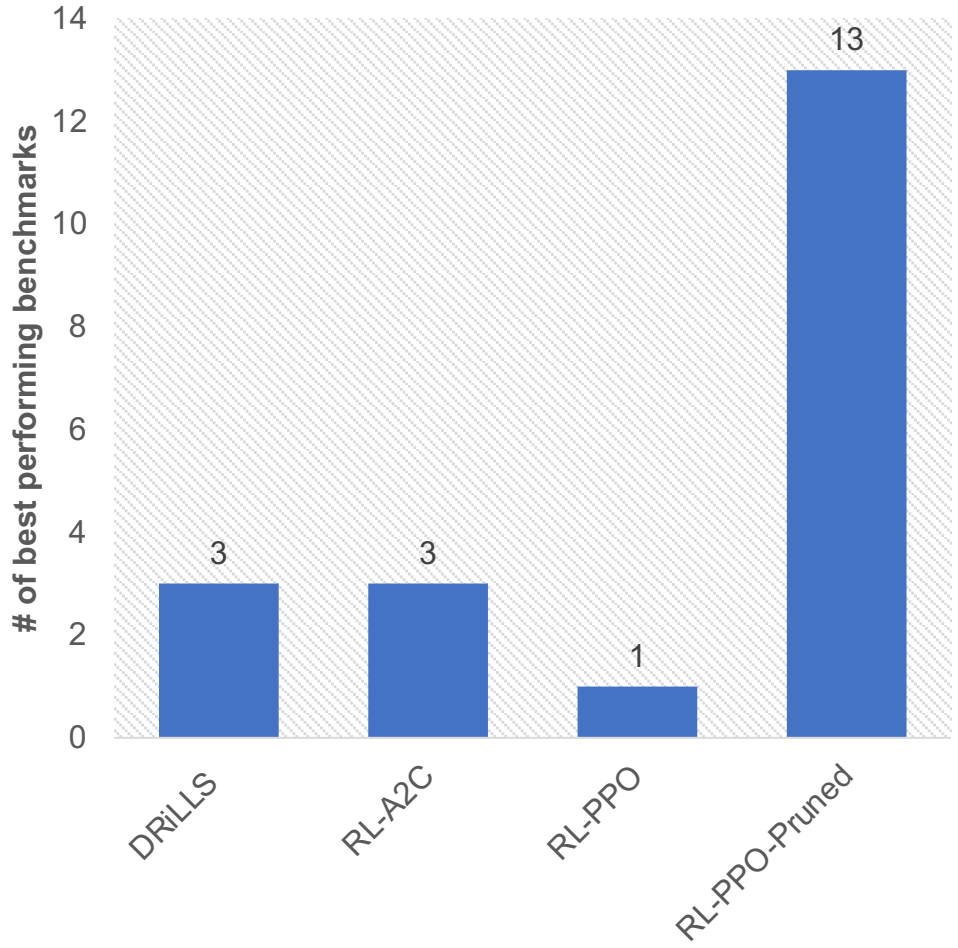


- All the values are Geomean of number of LUTs
- Smaller the value indicates better area reduction

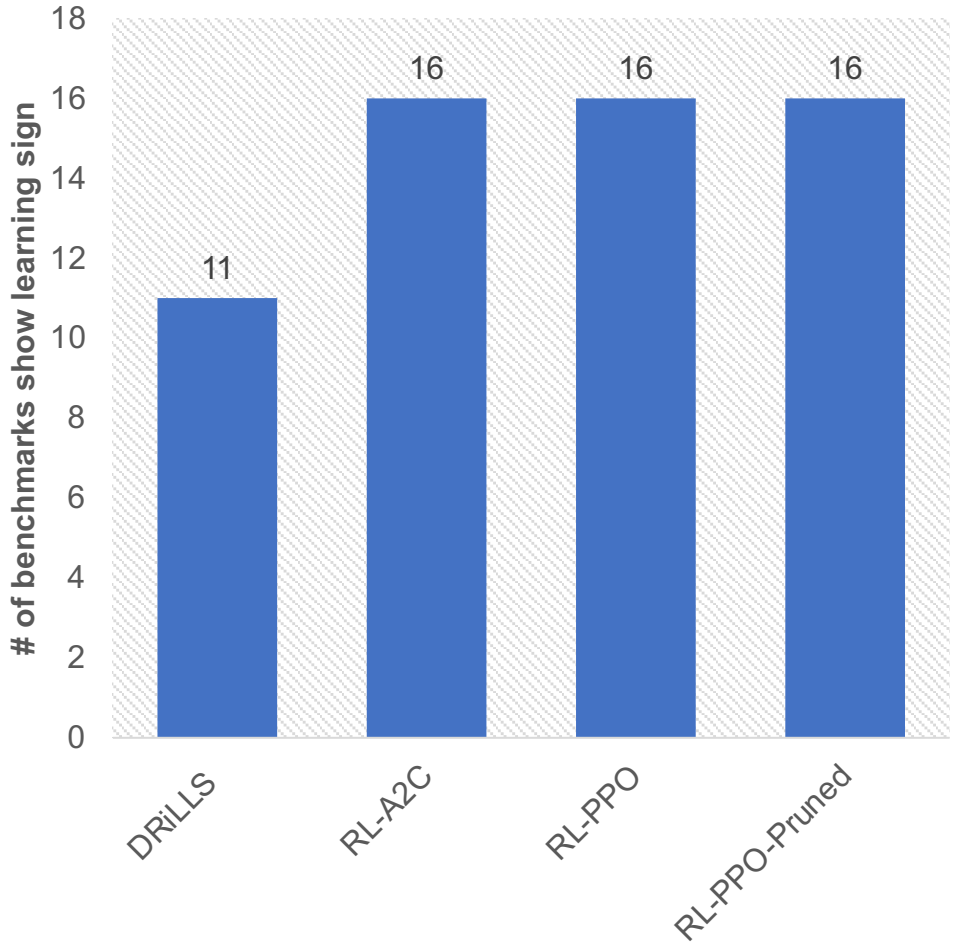
Best performing model

UNIVERSITY OF TORONTO

Training Performance

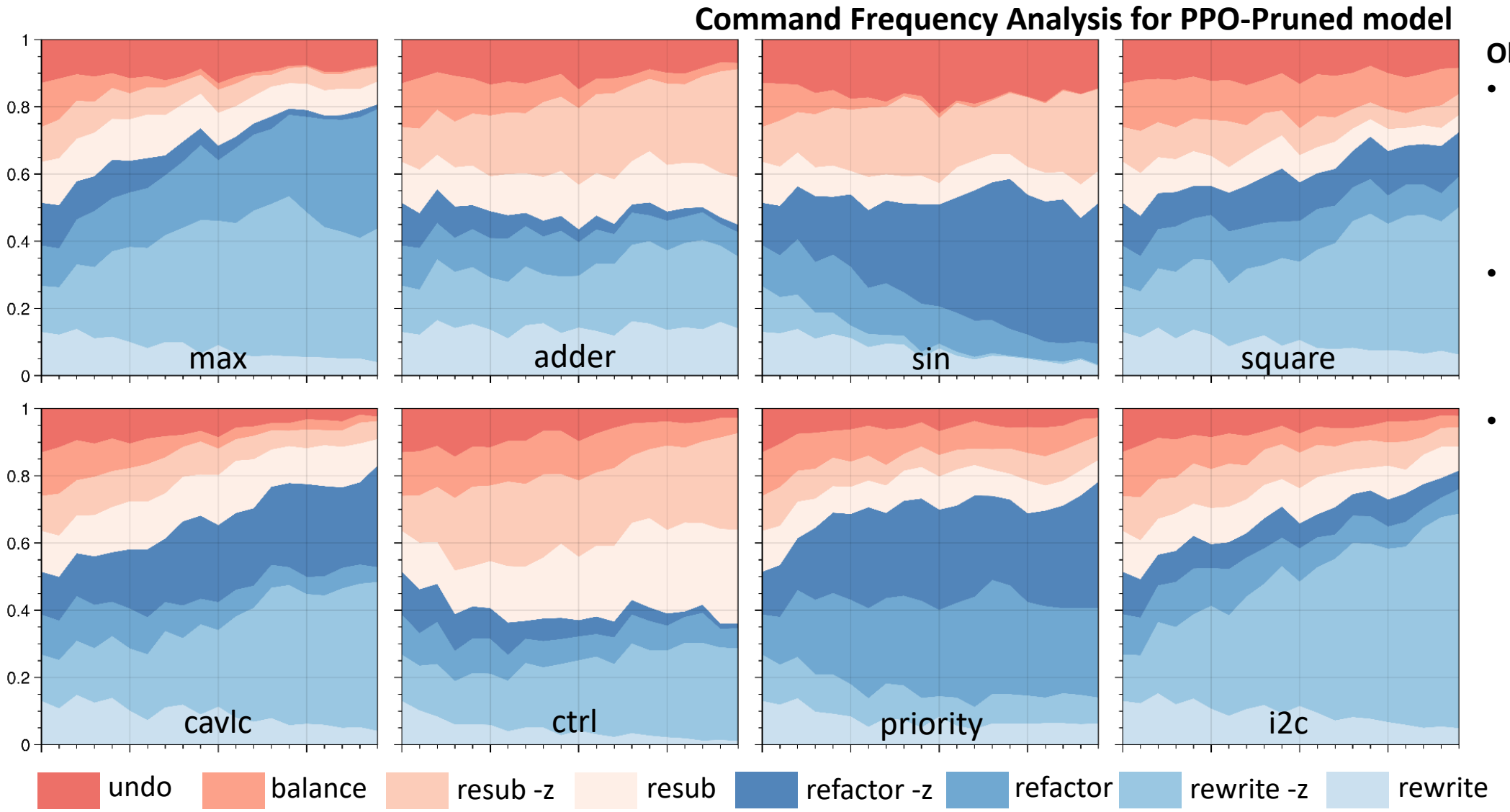


The count of best explored benchmarks of each RL method



Learning sign count of each RL method

Command Frequency Analysis



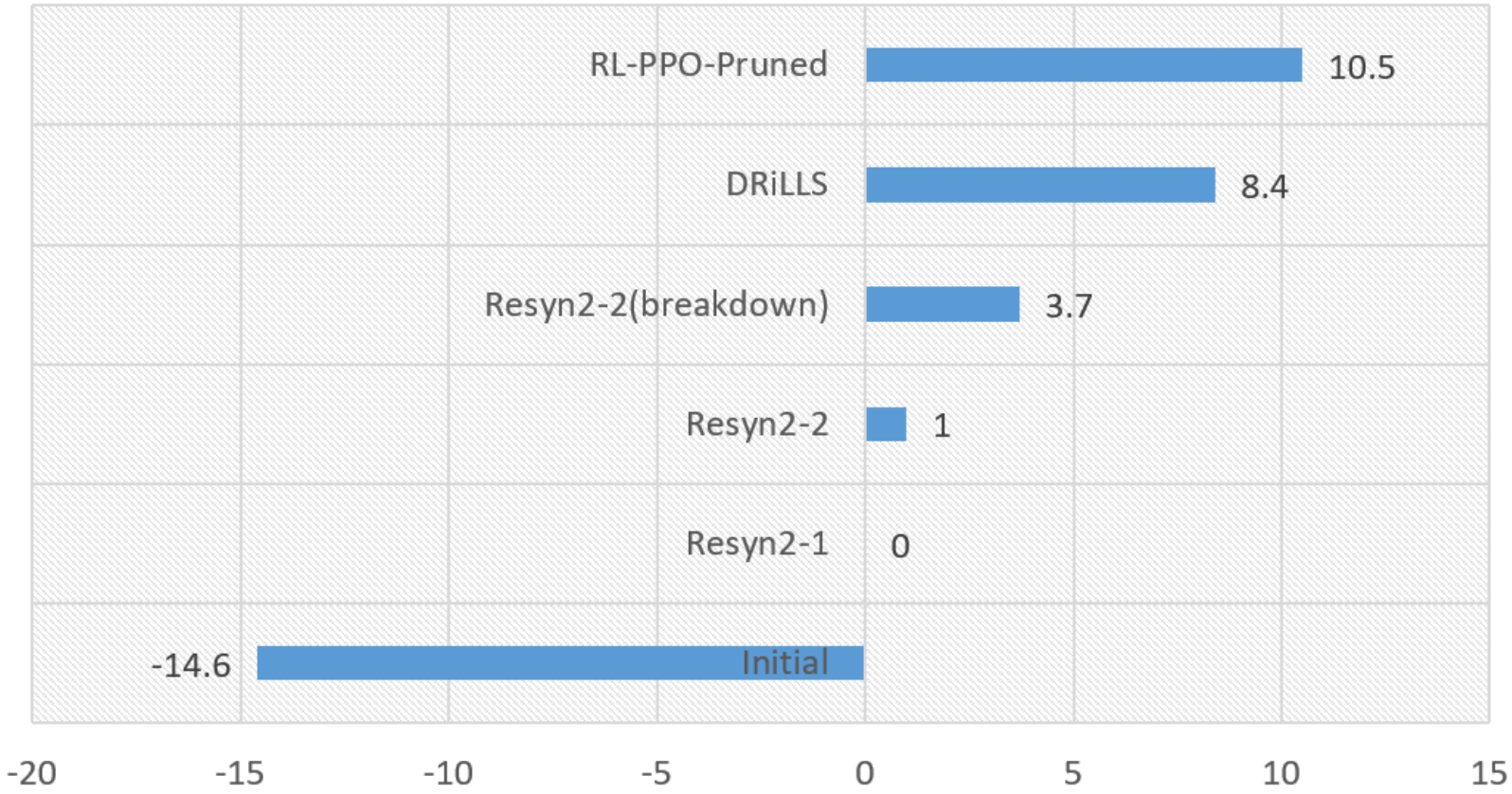
Observations:

- At first, same frequency (12.5% each) and then it varies
- Certain commands are more useful for certain circuits
- *rewrite –z* is the most popular command

UNIVERSITY OF TORONTO

Inference Performance

y-axis lists all the experiments where DRiLLS[1] is the prior work and x-axis is the percentage improvement against the resyn2-1.



Circuit-by-circuit style inference performance (GEOMEAN)

[1]A. Hosny, S. Hashemi, M. Shalan and S. Reda, "DRiLLS: Deep Reinforcement Learning for Logic Synthesis," 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), 2020

UNIVERSITY OF TORONTO

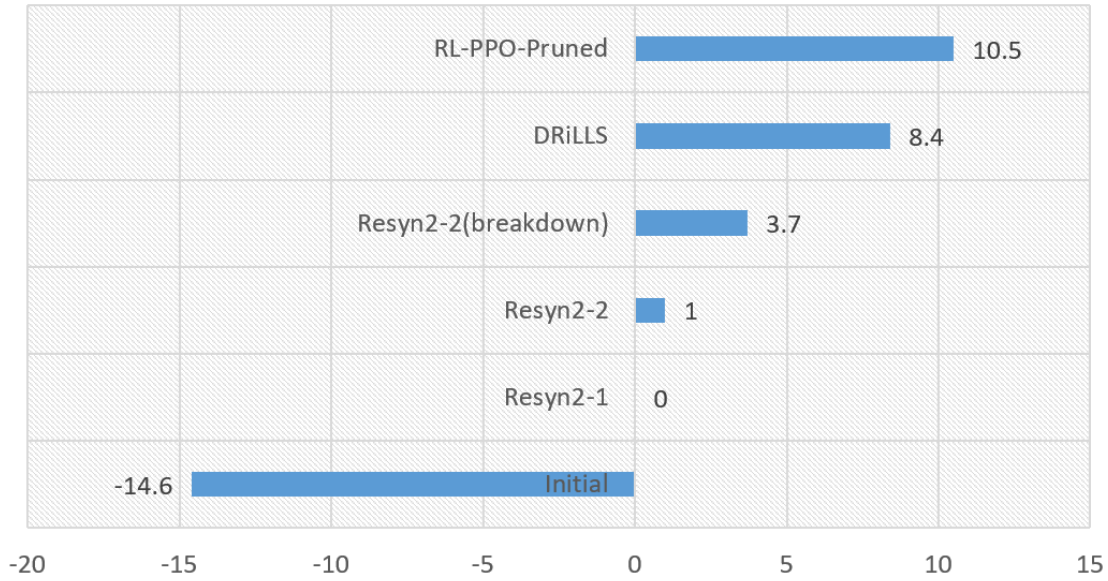
Experimental Setup: Generalized Style

- Generalized style: training and validation circuit sets

- Experiment 1: Train and validate on the same set
  - Training set: Full 16 EPFL benchmark circuits
  - Validation set: Full 16 EPFL benchmark circuits

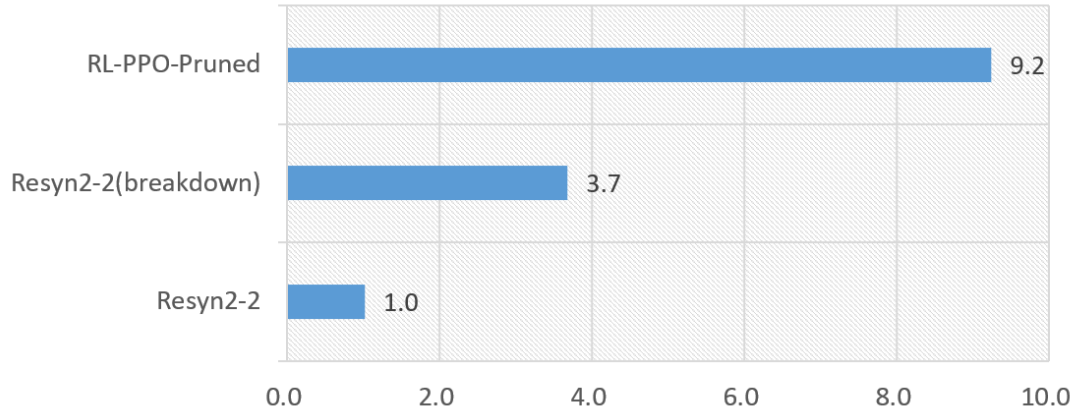
- Experiment 2: Train on one set and validate on the other set
  - Training set: Half of 16 EPFL benchmark circuits
  - Validation set: The other half of 16 EPFL benchmark circuits

Inference Performance

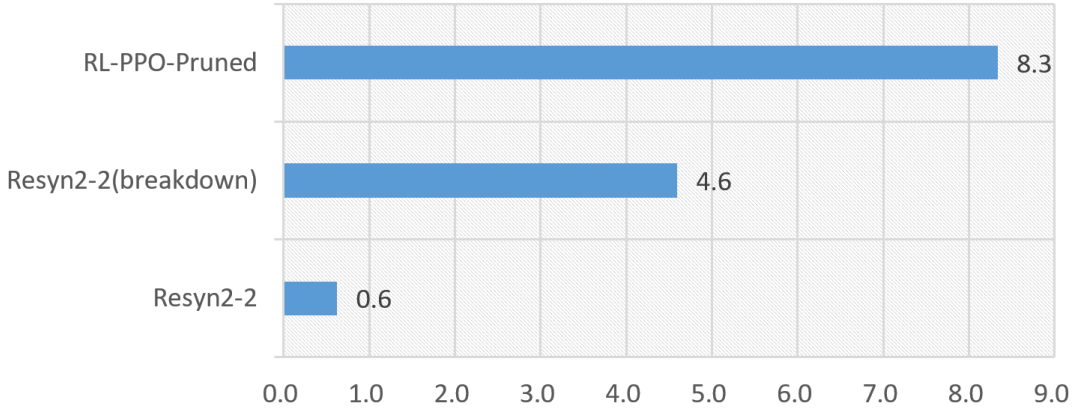


Circuit-by-circuit style inference performance (GEOMEAN)

y-axis lists all the experiments and x-axis is the percentage improvement against the resyn2-1.



a.) Generalization Experiment 1



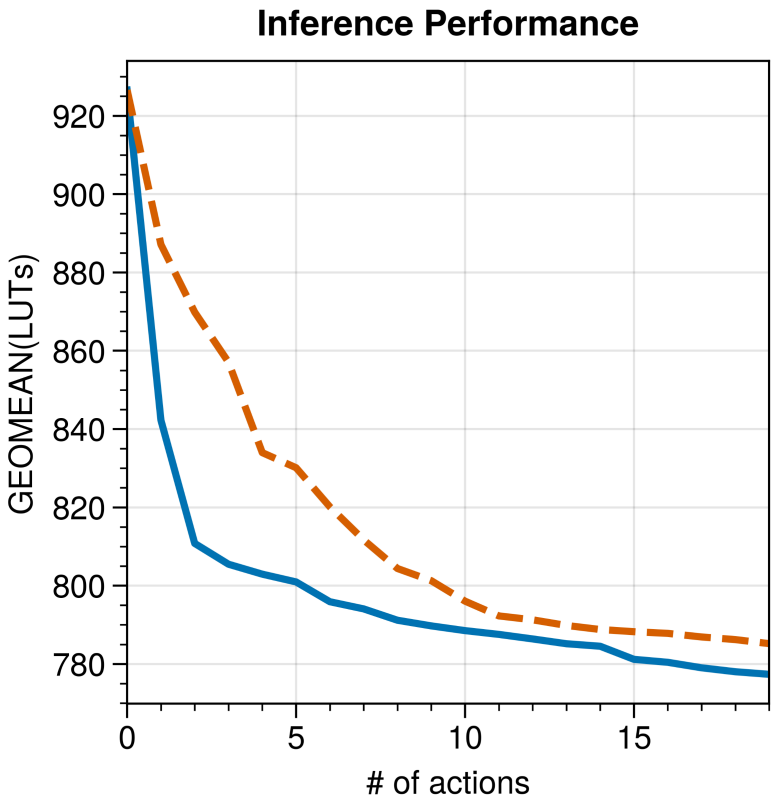
b.) Generalization Experiment 2

Generalized inference performance (GEOMEAN)

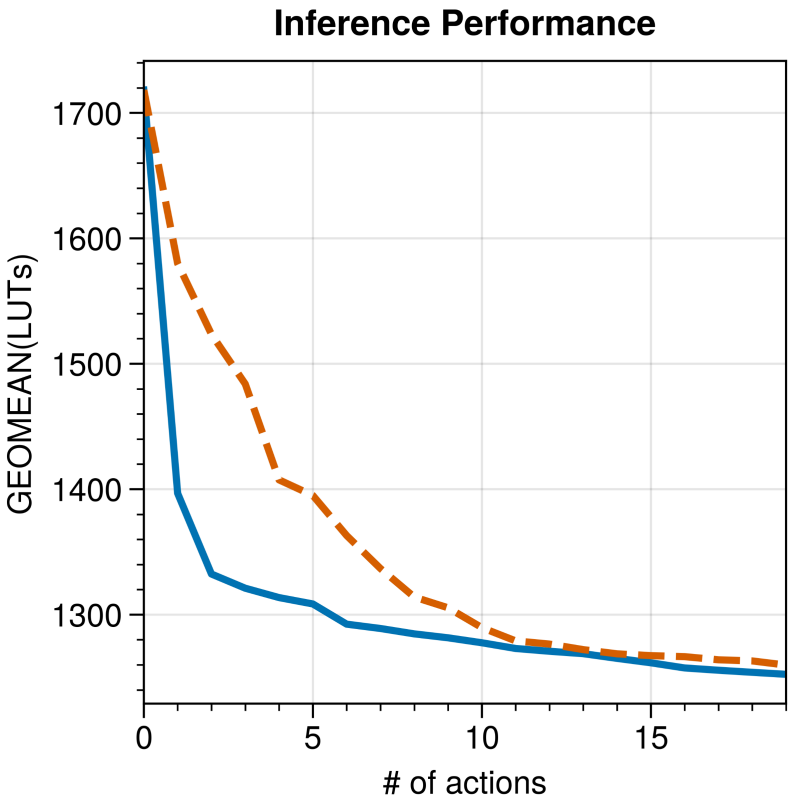
Area reductions of 8.3-10.5% vs. *resyn2*, depending on the selection of the training vs. validation circuit set.

UNIVERSITY OF TORONTO

Effectiveness Analysis



a.) Generalization Experiment 1



b.) Generalization Experiment 2

Effectiveness analysis – trained RL agent (blue) vs. random (orange)

UNIVERSITY OF TORONTO

Conclusion

- We present a logic synthesis framework that trains an RL agent to intelligently choose logic synthesis optimizations based on the characteristics of the circuit under optimization.

- Unlike previously published RL approaches, our approach demonstrates generalized learning across multiple circuits, and can use the trained agent in inference mode without the need for further training.

- The immediate next step is to extend the evaluation to depth-driven synthesis, automate the selection of episode length as well as fine-tuned a trained RL agent.

UNIVERSITY OF TORONTO

Thank You for Listening, Questions?

UNIVERSITY OF TORONTO