IWLS Programming Contest 2020: Team 3's Report

Authors: Po-Chun Chien[†], Yu-Shan Huang[†], Hao-Ren Wang[†], Jie-Hong Roland Jiang[†‡]

ALCom Lab

[†]Graduate Institute of Electronics Engineering

[‡]Department of Electrical Engineering
National Taiwan University

Outline

- ☐ Problem Description
- ☐ Our approach
  - ▪ DT-based model
  - ▪ NN-based model
  - ▪ bagging ensemble
- ☐ Experimental results
- ☐ Conclusions

PROBLEM DESCRIPTION

Problem Description

□ Learn an unknown Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ from a training dataset consisting of input-output pairs.

□ The learned function should be in the form of And-Inverter Graph (AIG) with strict hardware cost (≤ 5000 gates), and will be evaluated by its prediction accuracy in hidden testing dataset.

Benchmarks

☐ Each benchmark is provided in PLA format and contains 6400 minterms in training, validation and testing set respectively.

**The 100 Functions in our Benchmark Set: Arithmetic, Random Logic, ML**

|	
00-09	2 MSBs of k-bit adders for k in {16, 32, 64, 128, 256}
10-19	MSB of k-bit dividers and remainder circuits for k in {16, 32, 64, 128, 256}
20-29	MSB and middle bit of k-bit multipliers for k in {8, 16, 32, 64, 128}
30-39	k-bit comparators with k in {8, 16, ..., 4096}
40-49	LSB and middle bit of k-bit square-rooters with k in {16, 32, 64, 128, 256}
50-59	10 outputs of PicoJava design with 16-200 inputs and roughly balanced on- & offset
60-69	10 outputs of MCNC i10 design with 16-200 inputs and roughly balanced on- & offset
70-79	5 other outputs from MCNC benchmarks + 5 symmetric functions of 16 inputs
80-89	10 binary classification problems from MNIST group comparisons
90-99	10 binary classification problems from CIFAR-10 group comparisons

OUR APPROACH

DT-based Model

☐ Binary decision tree
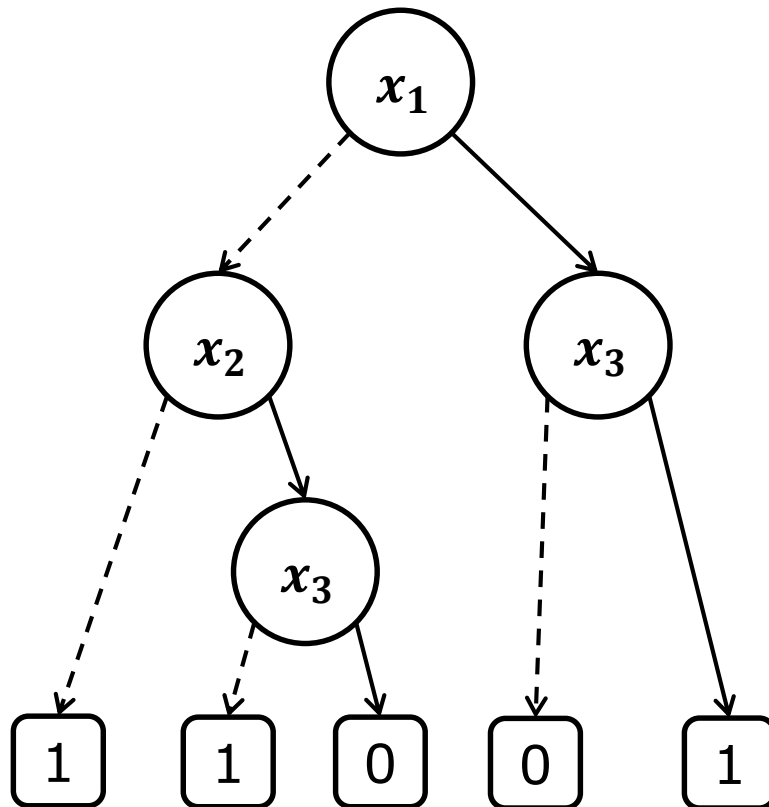
x_1	x_2	x_3	y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Entropy

☐ $entropy = -p \log_2 p - (1-p) \log_2(1-p),$
    where p is the probability of true label ($y = 1$).
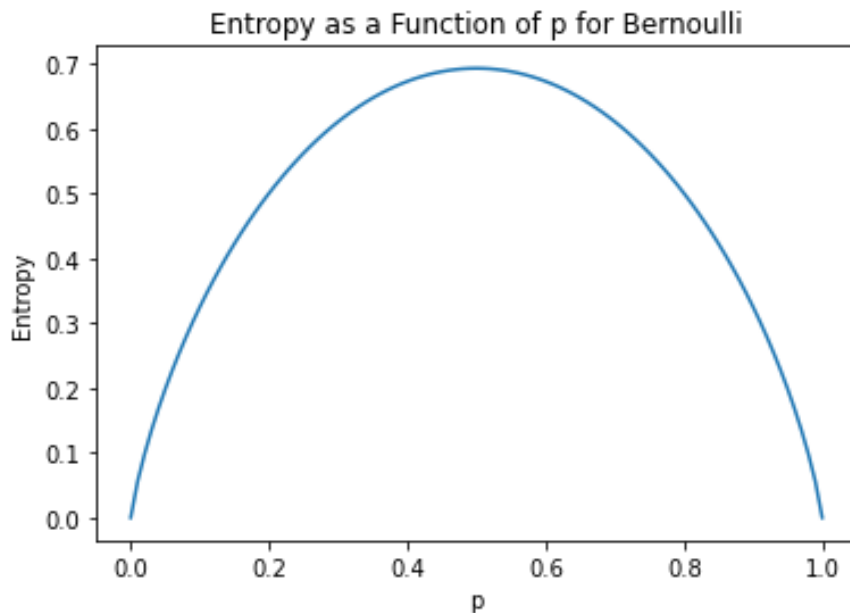
When $p = 0$ or 1, we have the lowest $entropy = 0 \rightarrow$ no uncertainty.

When $p = 0.5$, we have the maximum $entropy = 1 \rightarrow$ highest uncertainty.

Information Gain

- ☐ The branching variable is selected based on maximum information gain.
- ☐ Information gain of node n of variable x:

$$E_n - p_0 E_0 - p_1 E_1$$

where E_n is the entropy of n, E_0 is the entropy of the 0-child of n, E_1 is the entropy of the 1-child of n, p_0 and p_1 are the ratio of the data with $x = 0$ and $x = 1$, respectively.
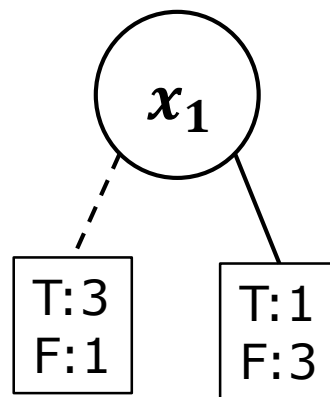
Growing DT

□ Choosing the 1st branching variable

example:

initial entropy: $-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

x_1	x_2	x_3	y
0	0	0	0
0	0	1	1
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	0
1	1	1	0
1	1	0	1

x_1

T:3 F:1   T:1 F:3

x_2

T:2 F:2   T:2 F:2
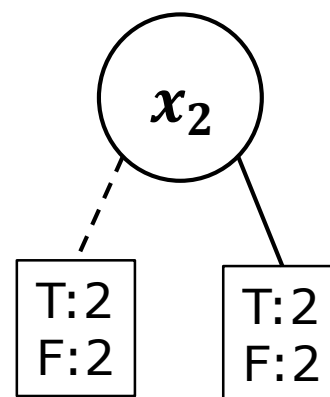
x_3

T:1 F:2   T:3 F:2

entropy   0.81   0.81   1   1   0.92   0.97

info. Gain
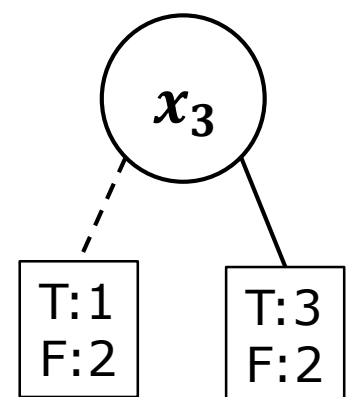
$1 - \frac{4}{8}\cdot 0.81 + \frac{4}{8}\cdot 0.81 = 0.19$
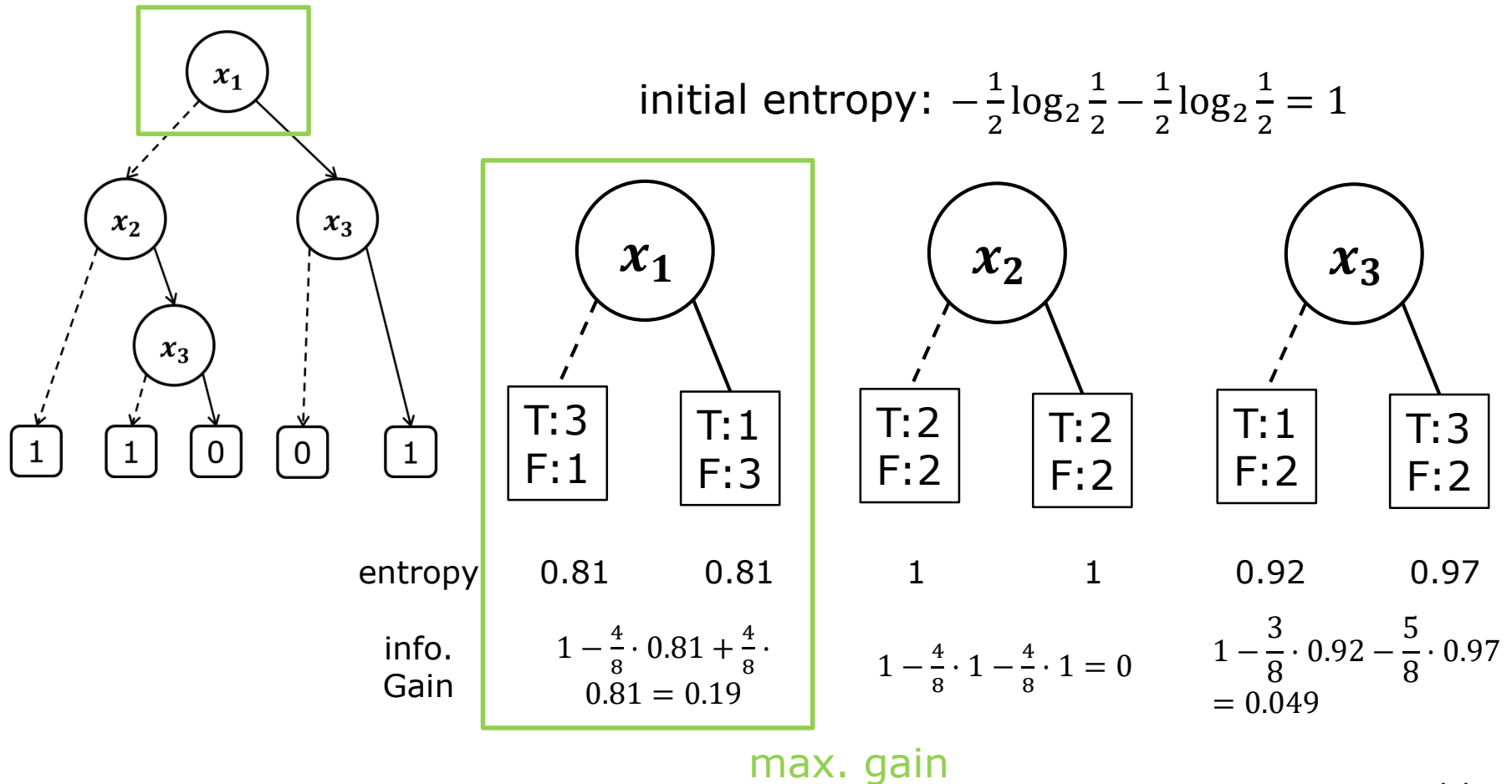
$1 - \frac{4}{8}\cdot 1 - \frac{4}{8}\cdot 1 = 0$

$1 - \frac{3}{8}\cdot 0.92 - \frac{5}{8}\cdot 0.97 = 0.049$

Growing DT

□ Choosing the 1st branching variable

initial entropy: $-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

	x_1		x_2		x_3	
	T:3 F:1	T:1 F:3	T:2 F:2	T:2 F:2	T:1 F:2	T:3 F:2
entropy	0.81	0.81	1	1	0.92	0.97

info. Gain

$1 - \frac{4}{8}\cdot 0.81 + \frac{4}{8}\cdot 0.81 = 0.19$

$1 - \frac{4}{8}\cdot 1 - \frac{4}{8}\cdot 1 = 0$

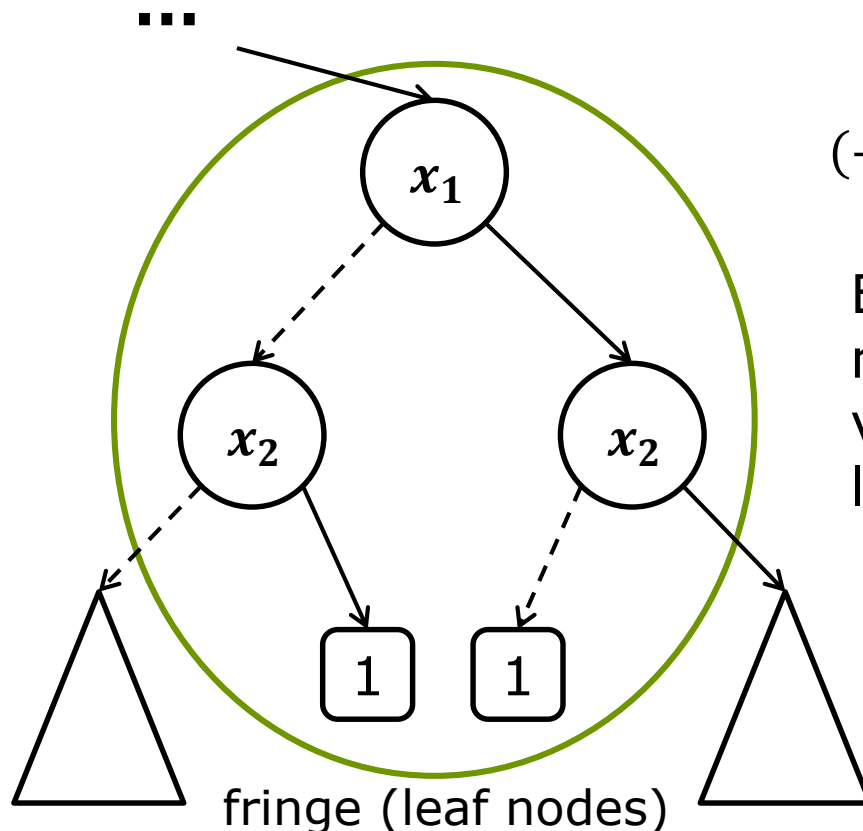$1 - \frac{3}{8}\cdot 0.92 - \frac{5}{8}\cdot 0.97 = 0.049$
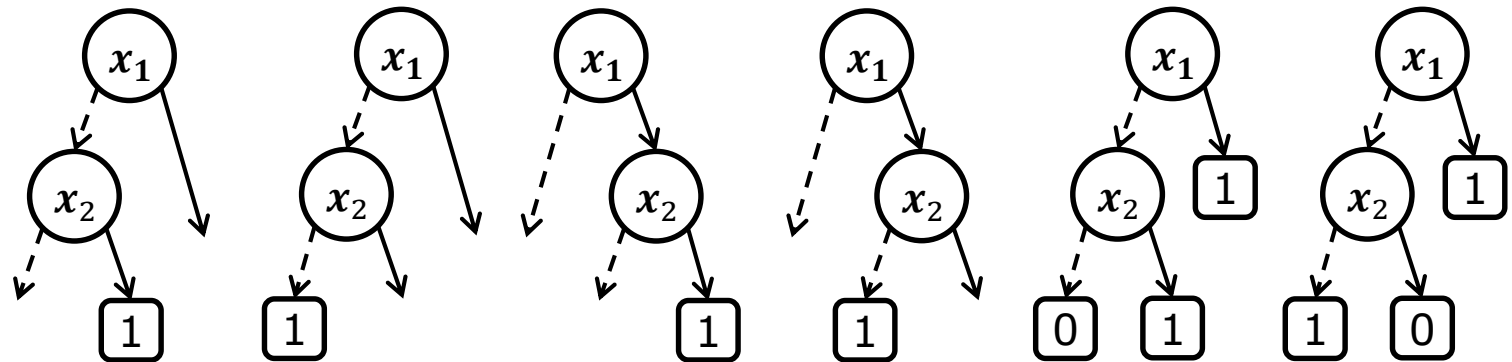
max. gain

11

DT-based Model

☐ Fringe-feature extraction [1, 2]

$$(\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2) = x_1 \oplus x_2$$

Extract $x_{new} = x_1 \oplus x_2$ as the new composite feature of 2 variables, and add it to the list of decision variables.

fringe (leaf nodes)

[1] Pagallo et al., 1990. [2] Oliveira et al., 1993.

DT-based Model

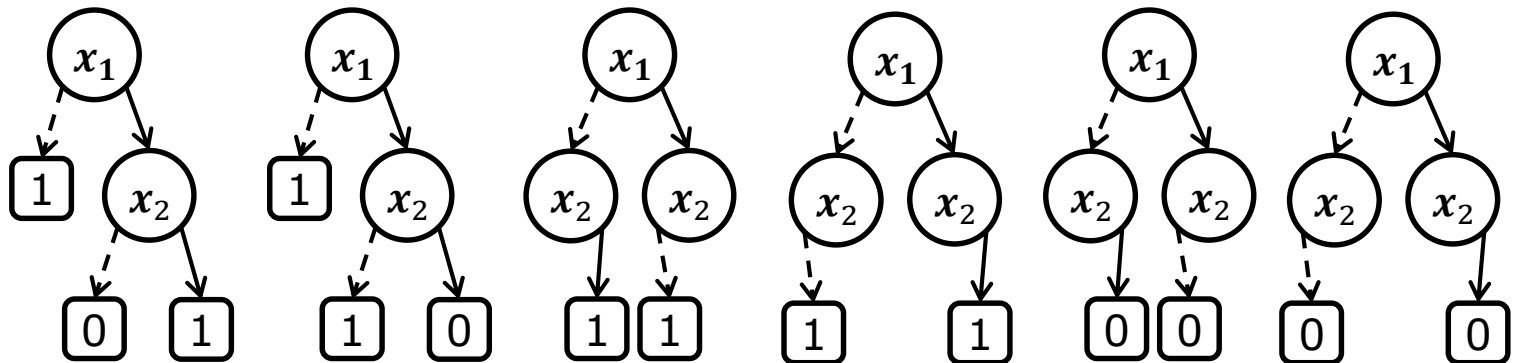$\overline{x_1}x_2$ $\overline{x_1}\,\overline{x_2}$ $x_1 x_2$ $x_1\overline{x_2}$ $x_1 + x_2$ $x_1 + \overline{x_2}$

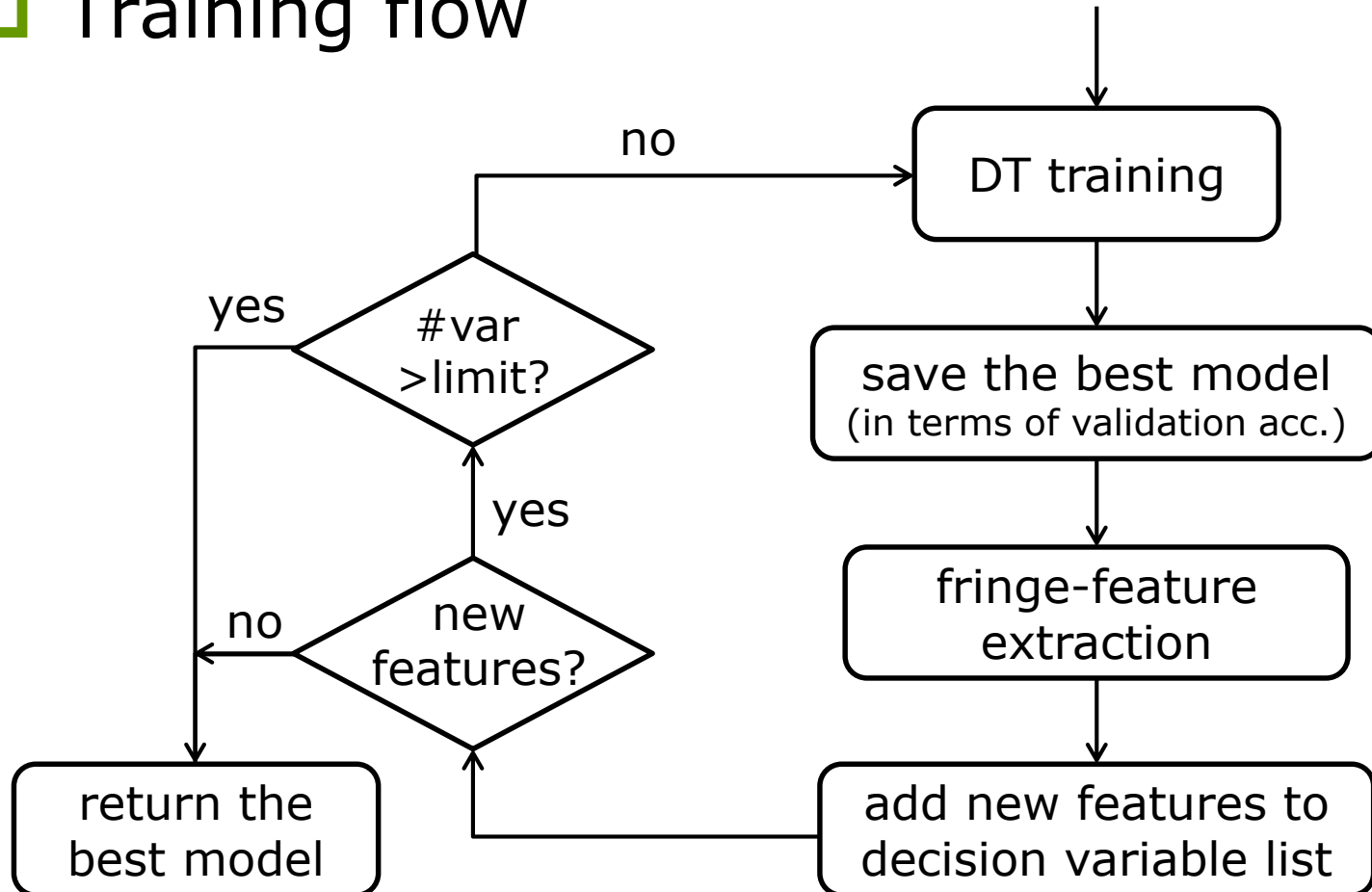$\overline{x_1} + x_2$ $\overline{x_1} + \overline{x_2}$ $x_1 \oplus x_2$ $\overline{x_1 \oplus x_2}$ $\overline{x_1 \oplus x_2}$ $x_1 \oplus x_2$

$- - -$: 0-edge   ——: 1-edge   ◯ : decision node   ☐ : leaf node
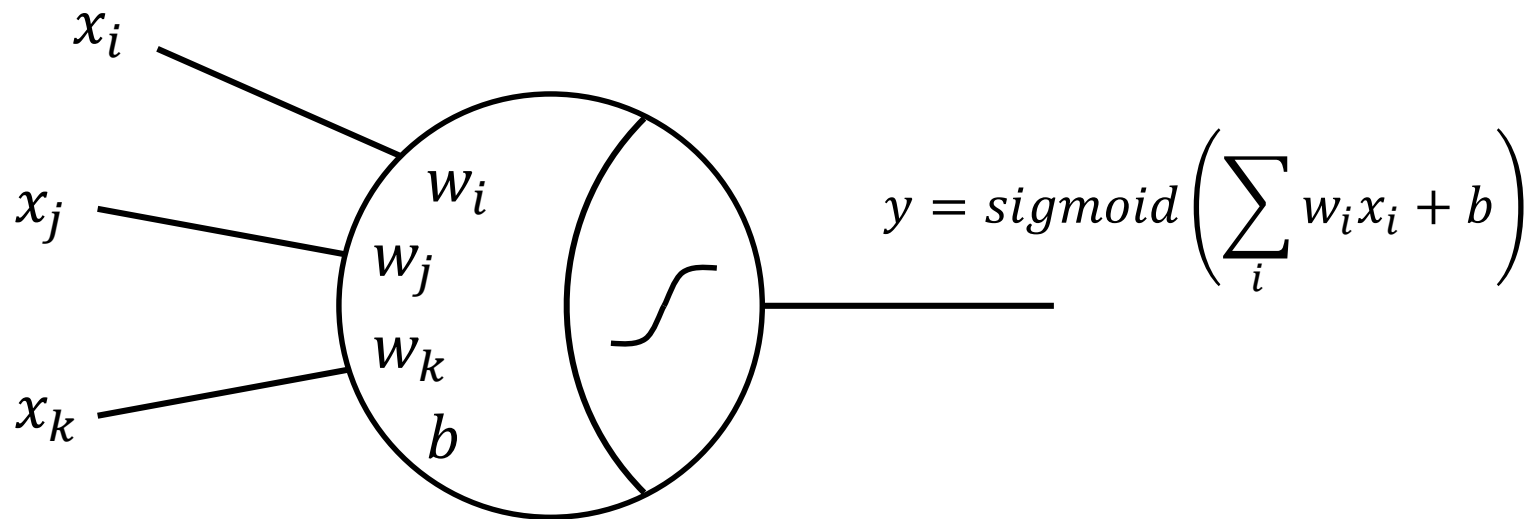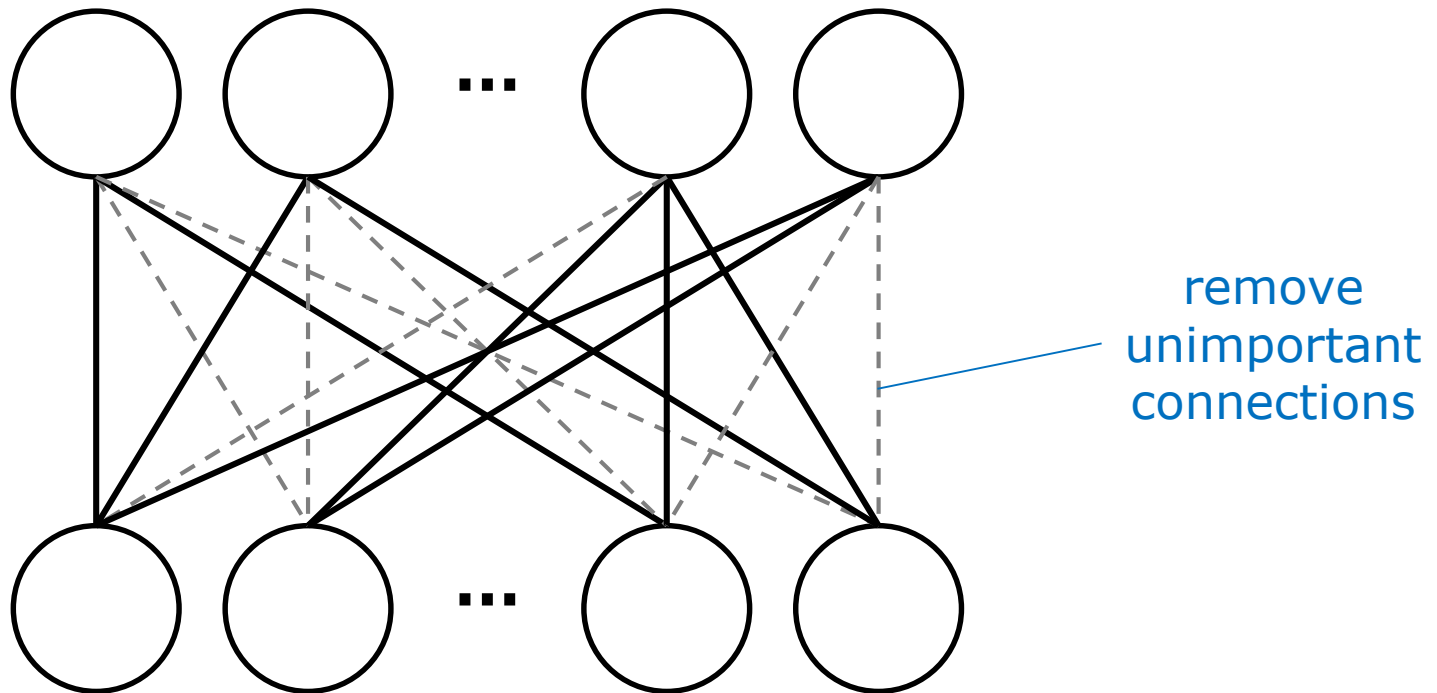
15

DT-based Model

□ **Training flow**

NN-based Model

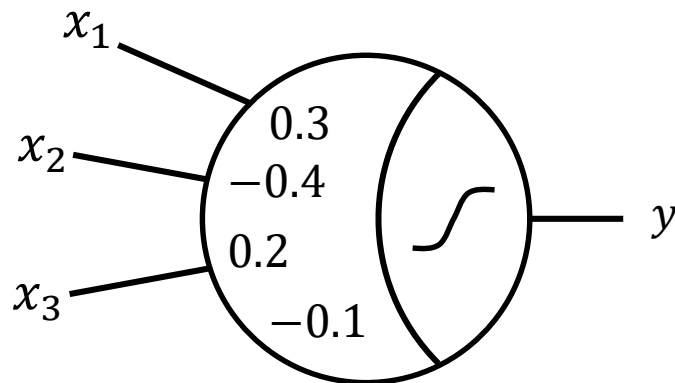☐ 3 layer network, each layer is fully-connected and uses sigmoid as the activation function

$$y = sigmoid\left(\sum_i w_i x_i + b\right)$$

NN-based Model

□ Connection pruning [3]

remove unimportant connections

[3] Han et al., 2015.

NN-based Model

☐ Convert neurons to LUTs

x_1	x_2	x_3	y	$\hat{y}$
0	0	0	0.48	0
0	0	1	0.52	1
0	1	0	0.38	0
0	1	1	0.43	0
1	0	0	0.55	1
1	0	1	0.60	1
1	1	0	0.45	0
1	1	1	0.50	1

NN-based Model

☐ Training flow
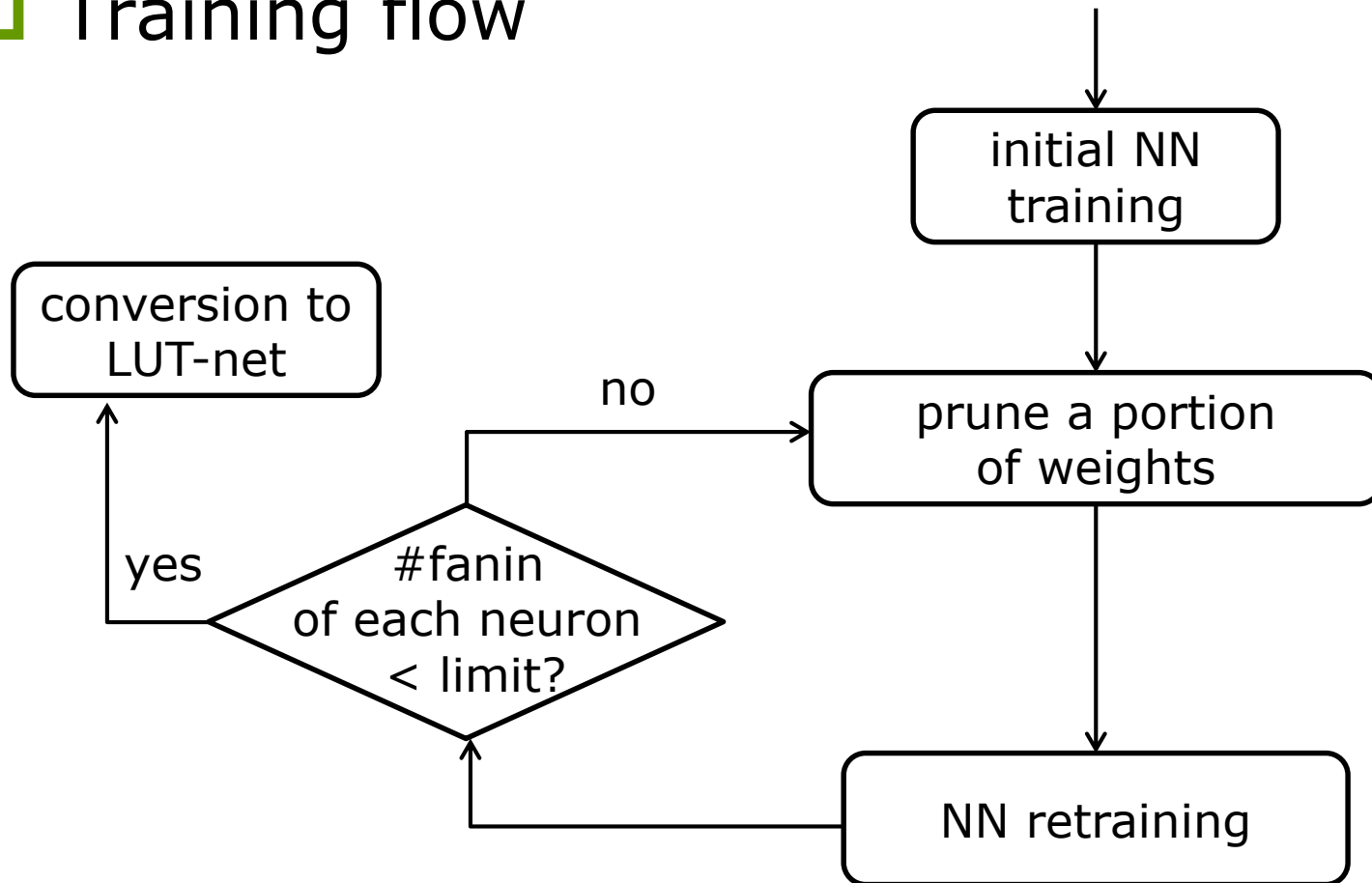
Bagging Ensemble
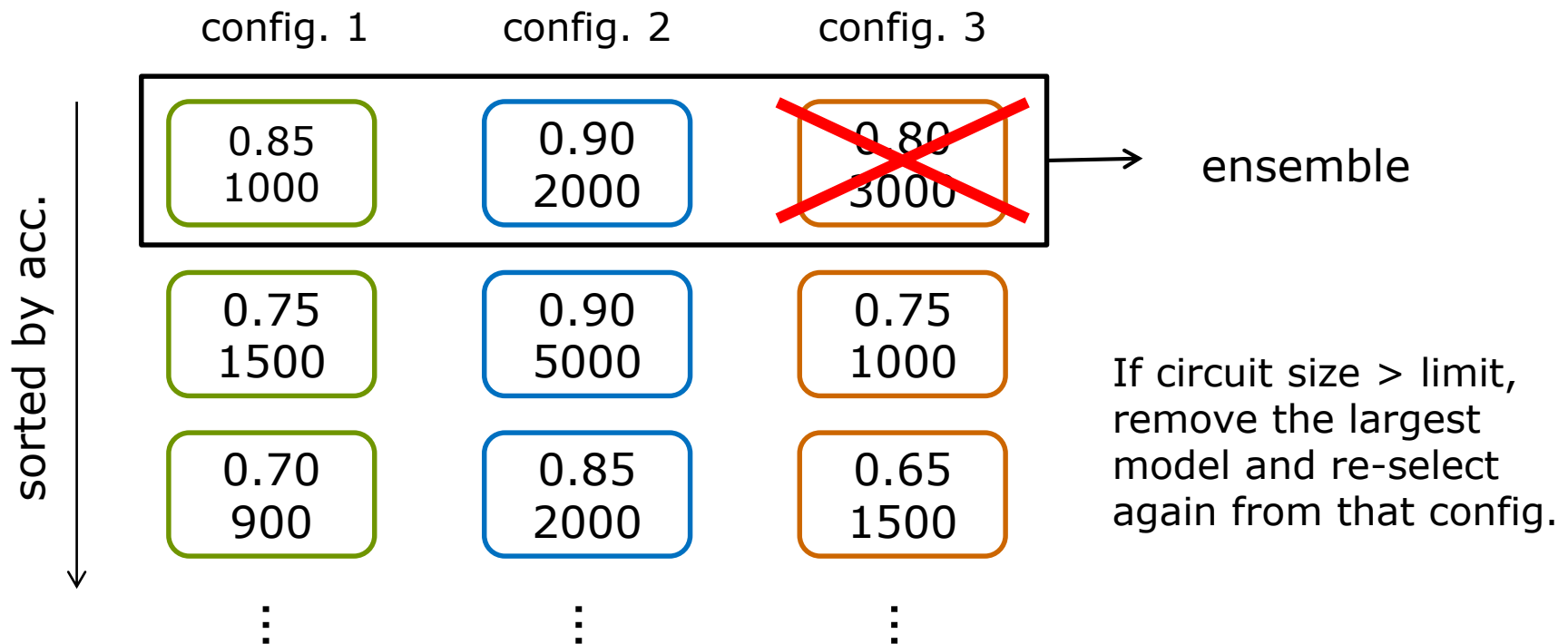
☐ Re-partitioning the dataset

Under each configuration, train multiple models with different methods and hyper-parameters.

Bagging Ensemble

□ Model selection heuristic

model: | val. acc.
       | #gate

config. 1   config. 2   config. 3

sorted by acc. →

| 0.85 | 0.90 | ~~0.80~~ |
| 1000 | 2000 | ~~3000~~ | → ensemble

| 0.75 | 0.90 | 0.75 |
| 1500 | 5000 | 1000 |

If circuit size > limit,
remove the largest
model and re-select
again from that config.

| 0.70 | 0.85 | 0.65 |
| 900 | 2000 | 1500 |

⋮   ⋮   ⋮

EXPERIMENTAL RESULTS

Experimental Setup

☐ Our methods were implemented with ML packages scikit-learn [4] and Pytorch [5].

☐ The synthesized circuits were optimized by ABC [6].

[4] Pedregosa et al., 2011 [5] Paszke et al., 2019. [6] Brayton et al., 2010.

Experimental Results

method	avg. train acc.	avg. valid acc.	avg. test acc.	avg. size (#gate)
DT	90.41%	80.33%	80.15%	303.90
Fr-DT	92.47%	85.37%	85.23%	241.47
NN	82.64%	80.91%	80.90%	10981.38
LUT-Net*[7]	98.37%	72.78%	72.68%	64004.39
ensemble	-	-	87.25%	1550.33

* LUT-Net is trained with the same avg. #connection as NN

[7] Chatterjee et al., 2018.

Accuracy Comparison
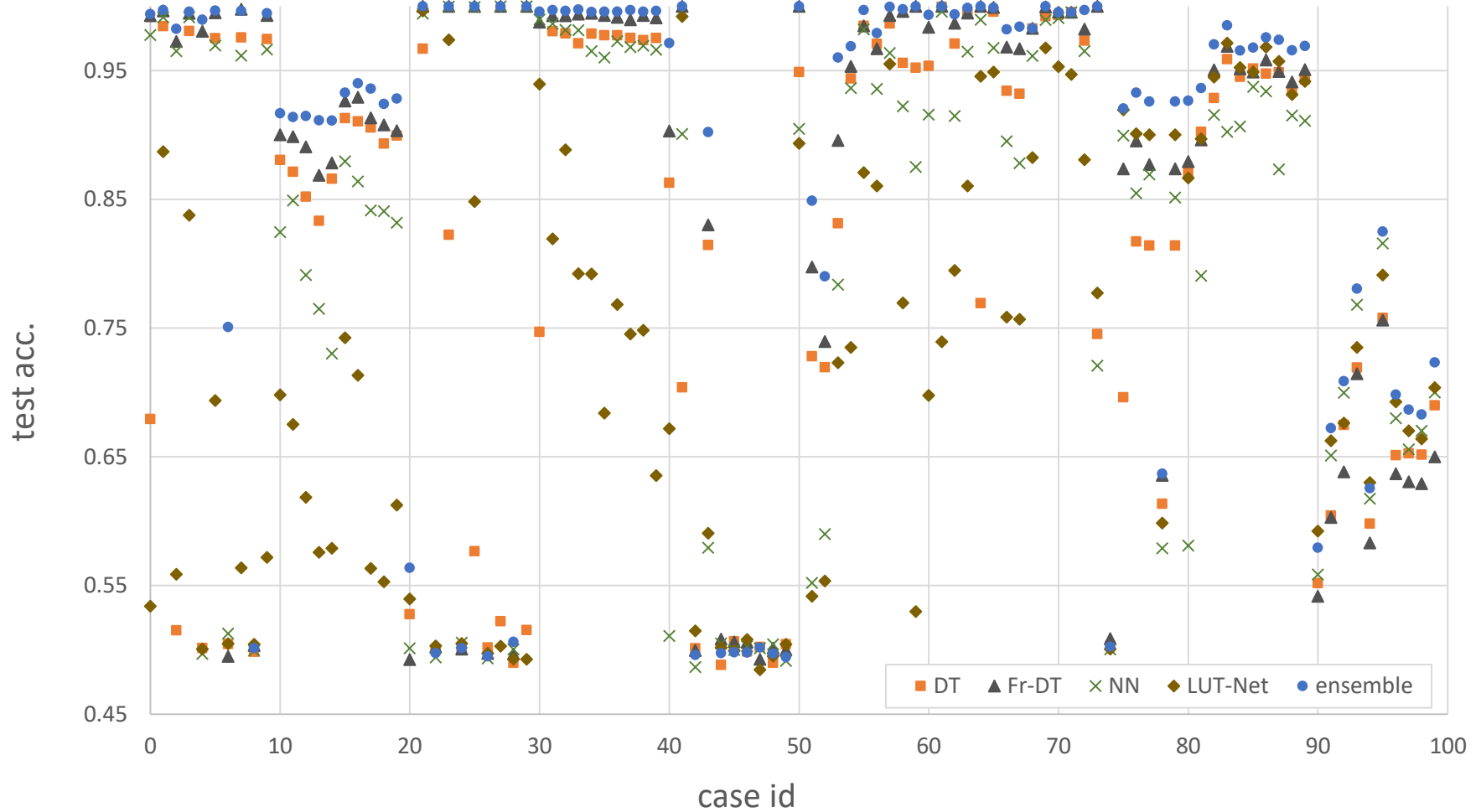
Circuit Size Comparison
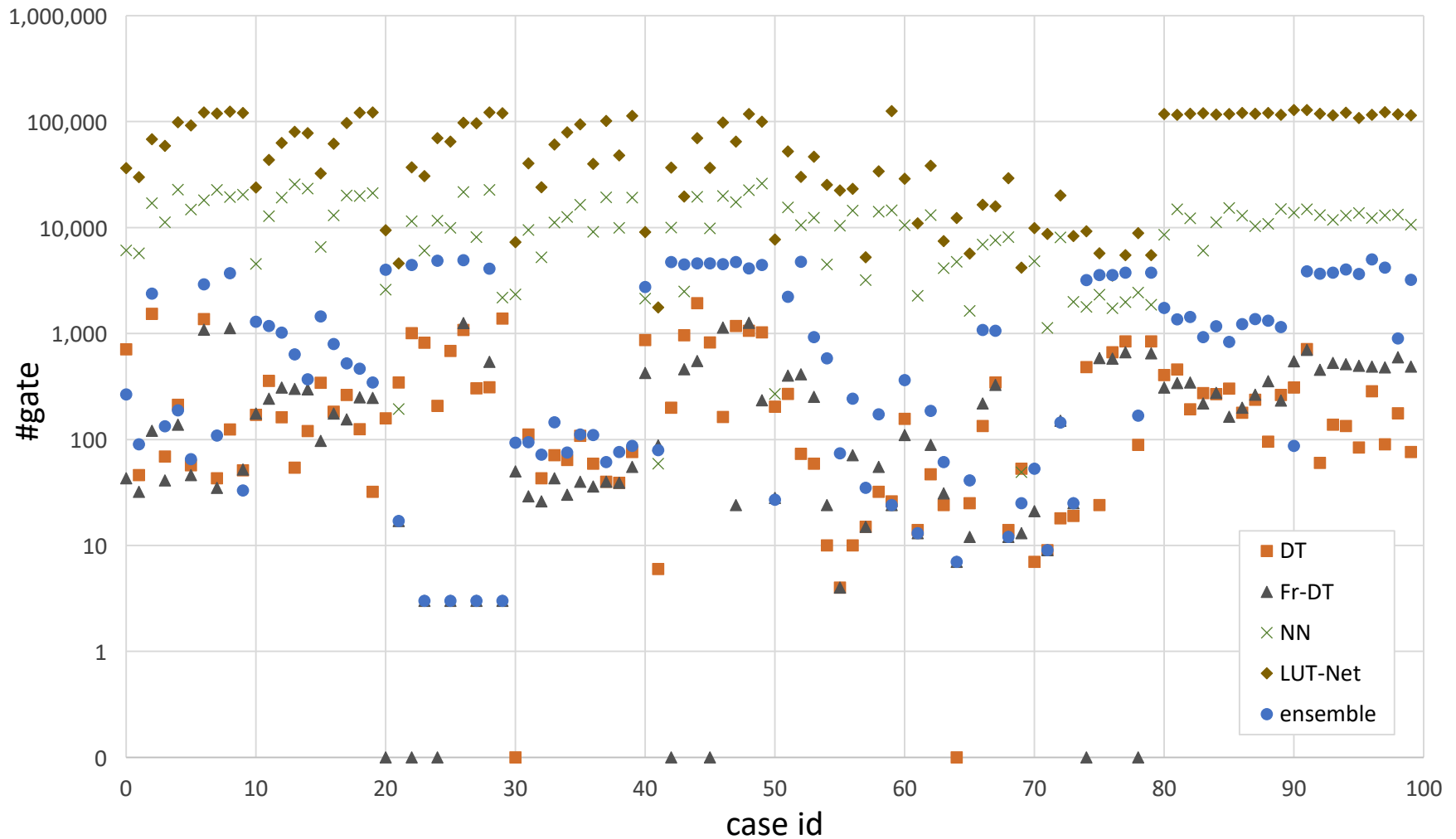
Contest Results

| team	↓ test accuracy	And gates	levels	overfit
1	**88.69**	2517.66	39.96	1.86
7	87.50	1167.50	32.02	**0.05**
8	87.32	1293.92	21.49	0.14
3	87.25	1550.33	21.08	5.76
2	85.95	731.92	80.63	8.70
9	84.65	991.89	103.42	1.75
4	84.64	1795.31	21.00	0.48
5	84.08	1142.83	145.87	4.17
10	80.25	**140.25**	10.90	3.86
6	62.40	356.26	**8.73**	0.88

ours → 3

test acc. and circuit size summary of each team

29
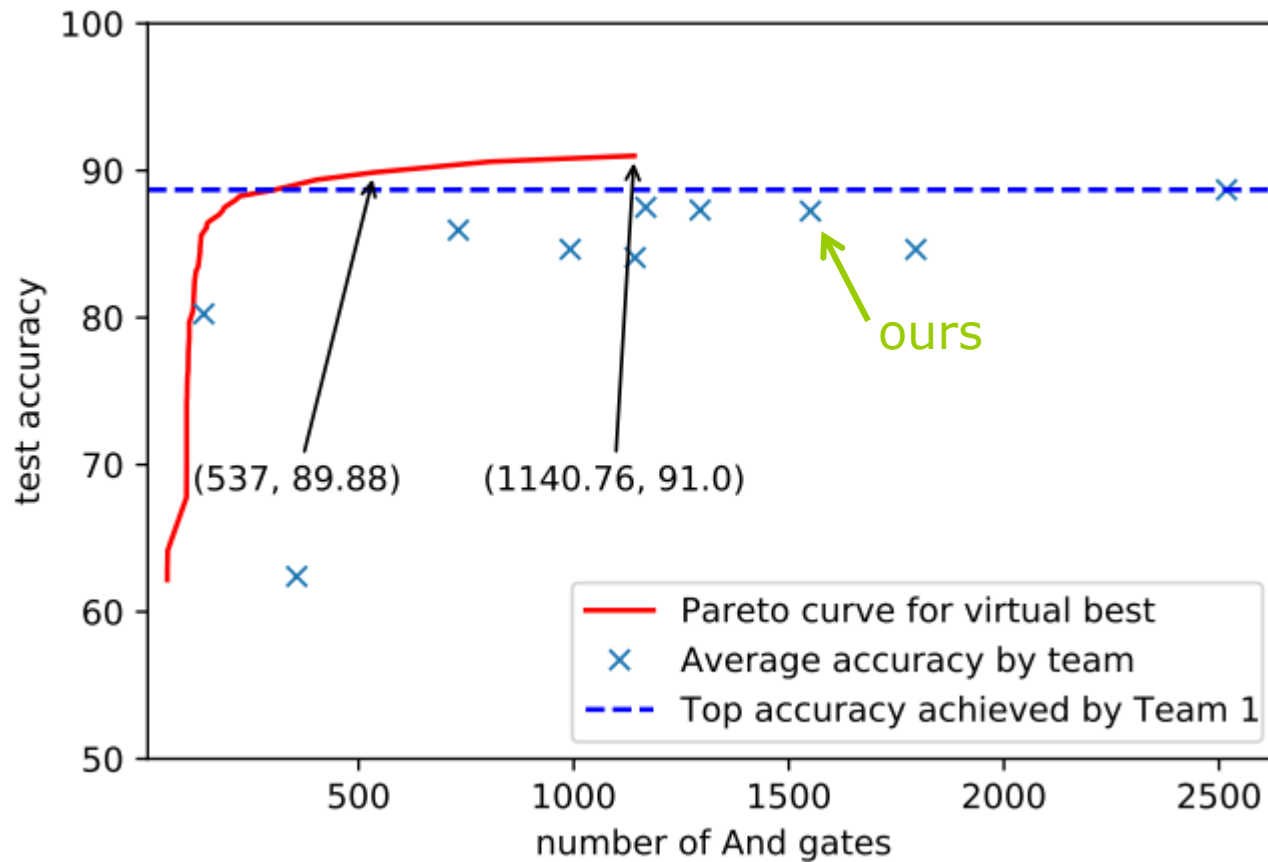
Contest Results

#gate vs. test acc.

Contest Results
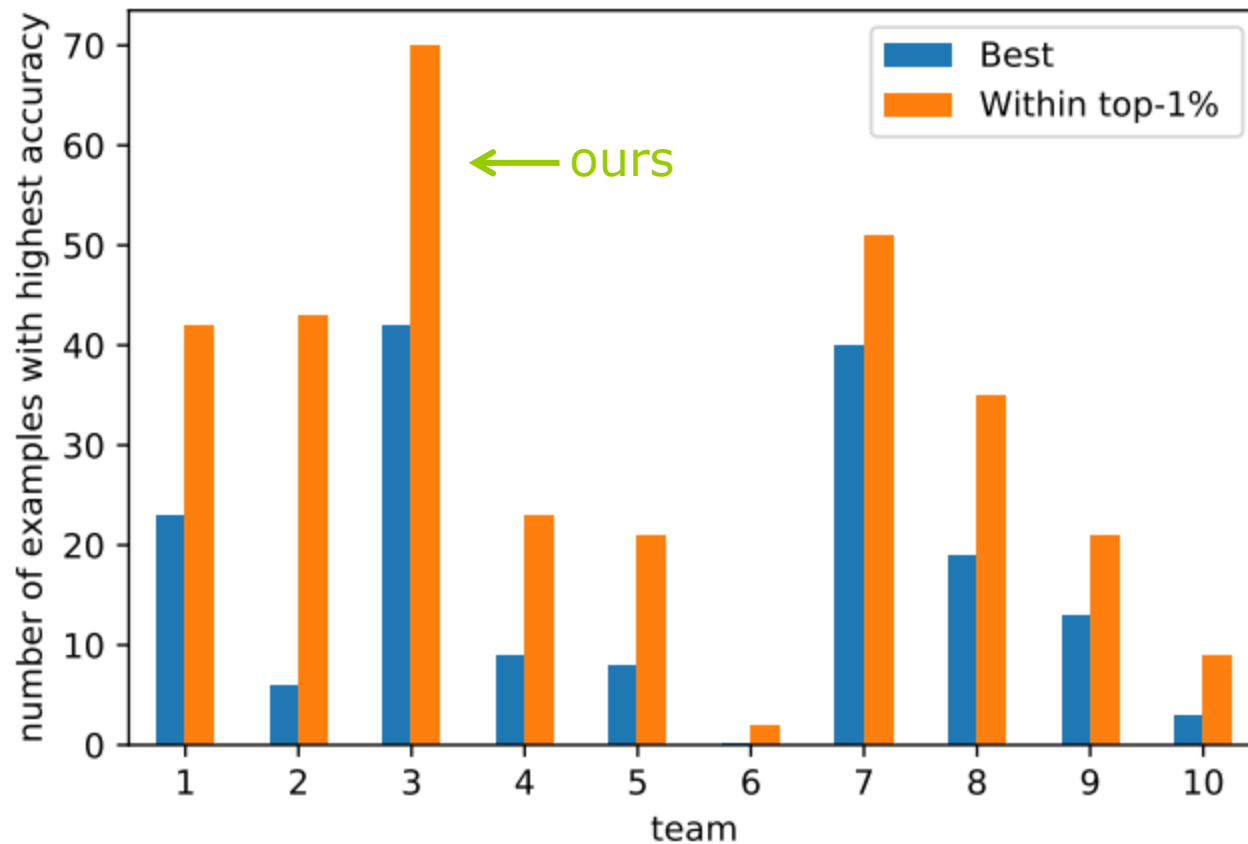
Top-accuracy results achieved by each team.

CONCLUSIONS

Conclusions

- Boolean functions can be learned by DT-based and NN-based methods.
- In our experiments, applying decision tree with fringe feature extraction could generally result in better model in terms of both accuracy and circuit size.
- NN models, though exceeded circuit size limit in many cases, they performed better in some other cases than DT models.
- After ensemble, we could achieve 87.25% accuracy on hidden test set.
- Our team achieved the **highest testing accuracy in most (42 out of 100) cases**, and ranked 4th in terms of the average testing accuracy.

THE END