How deep learning can drive physical synthesis towards more predictable legalization

Renan Netto, Sheiny Fabre, Tiago Augusto Fontana, Vinicius Livramento, Laércio Pilla, José Luís Güntzel

Embedded Computing Lab (ECL) — Dept. of Computer Science and Statistics
Federal University of Santa Catarina (UFSC)

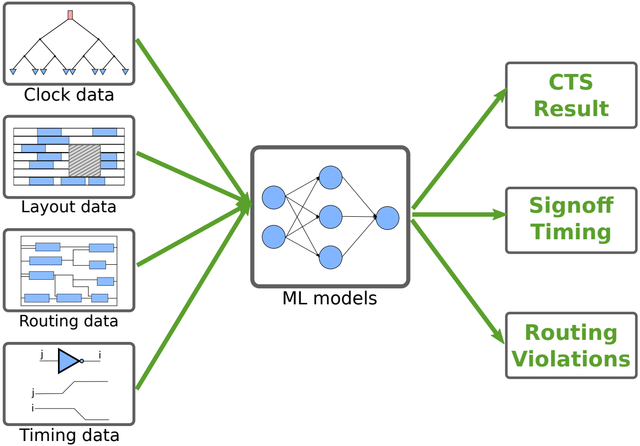
ISPD - April 14-17 - San Francisco, CA

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

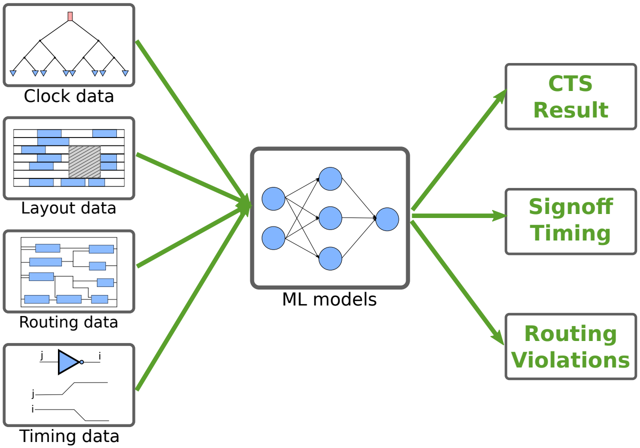
Ophidian

Outline

Machine learning applications in physical design



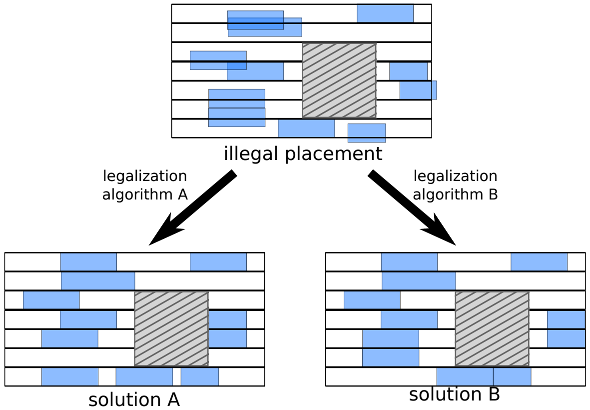
Machine learning applications in physical design



It has not been used to predict legalization yet!

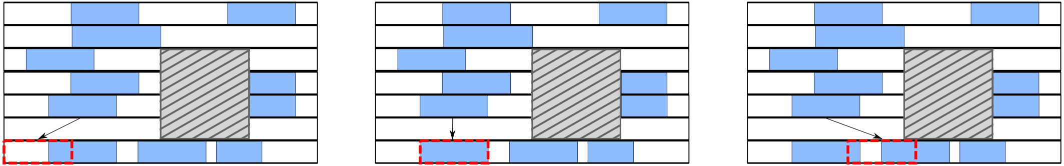
Machine learning applications in legalization

1) Choosing among different legalization algorithms



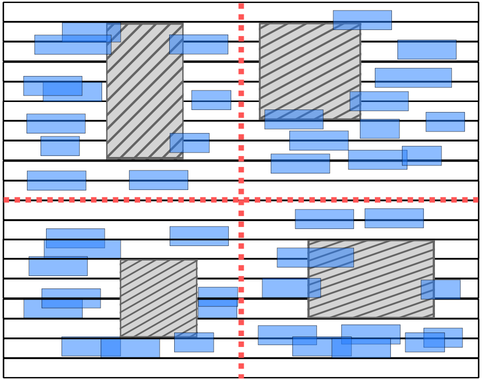
Machine learning applications in legalization

2) Guiding an incremental placement technique



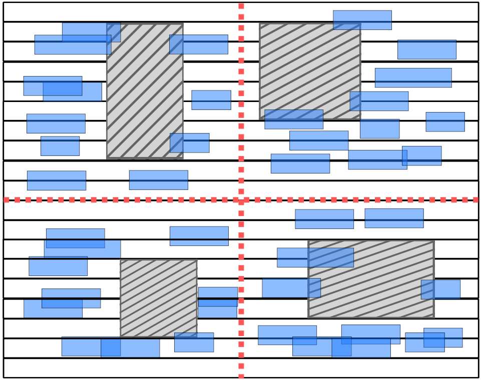
Machine learning applications in legalization

3) Guiding an circuit partitioning strategy



Machine learning applications in legalization

3) Guiding an circuit partitioning strategy



This work explores mainly option (3), but also partially explores option (2)

Contributions

- **Feature extraction strategy** for training machine learning models.
- Evaluation of **different ML models** in order to select the best one for this problem.
- We employed the best ML model as a **pruning mechanism** for a circuit partitioning strategy.

Table of related works

Work	Prediction	Features	ML model
Kahng et al.	CTS outcome	clock data	non-convolutional
Han et al.	signoff timing	timing data	non-convolutional
Zhou et al.	# of DRCs	layout data, routing data	non-convolutional
Chan et al.	locations of DRCs	layout data, routing data	non-convolutional
Fabrizi et al.	short violations	layout data, routing data	non-convolutional
Xie et al.	# and location of DRCs	circuit snapshot	convolutional
This work	legalization quality	layout data, circuit snapshot	non-convolutional, convolutional

Table of related works

Work	Prediction	Features	ML model
Kahng et al.	CTS outcome	clock data	non-convolutional
Han et al.	signoff timing	timing data	non-convolutional
Zhou et al.	# of DRCs	layout data, routing data	non-convolutional
Chan et al.	locations of DRCs	layout data, routing data	non-convolutional
Fabrizi et al.	short violations	layout data, routing data	non-convolutional
Xie et al.	# and location of DRCs	circuit snapshot	convolutional
This work	legalization quality	layout data, circuit snapshot	non-convolutional, convolutional

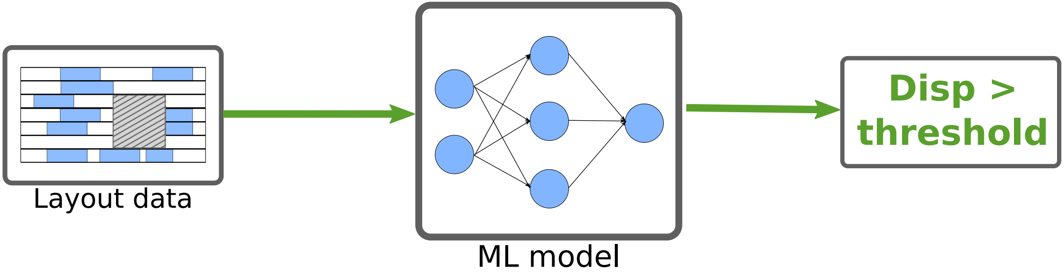
Table of related works

Work	Prediction	Features	ML model
Kahng et al.	CTS outcome	clock data	non-convolutional
Han et al.	signoff timing	timing data	non-convolutional
Zhou et al.	# of DRCs	layout data, routing data	non-convolutional
Chan et al.	locations of DRCs	layout data, routing data	non-convolutional
Fabrizi et al.	short violations	layout data, routing data	non-convolutional
Xie et al.	# and location of DRCs	circuit snapshot	convolutional
This work	legalization quality	layout data, circuit snapshot	non-convolutional, convolutional

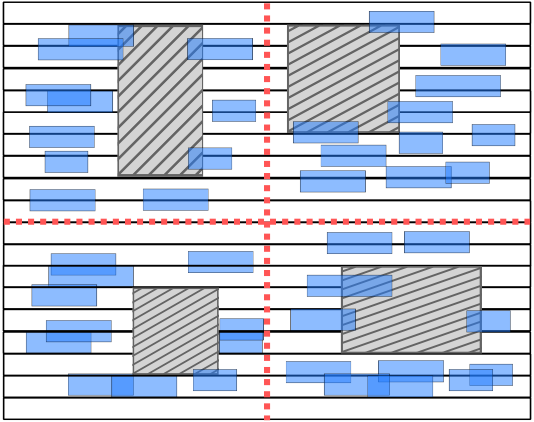
Table of related works

Work	Prediction	Features	ML model
Kahng et al.	CTS outcome	clock data	non-convolutional
Han et al.	signoff timing	timing data	non-convolutional
Zhou et al.	# of DRCs	layout data, routing data	non-convolutional
Chan et al.	locations of DRCs	layout data, routing data	non-convolutional
Fabrizi et al.	short violations	layout data, routing data	non-convolutional
Xie et al.	# and location of DRCs	circuit snapshot	convolutional
This work	legalization quality	layout data, circuit snapshot	non-convolutional, convolutional

Methodology overview

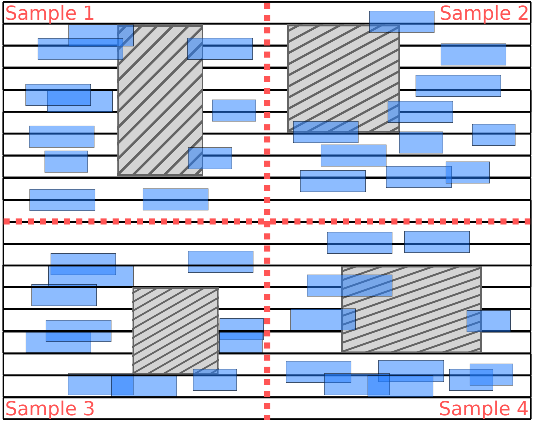


Training data generation



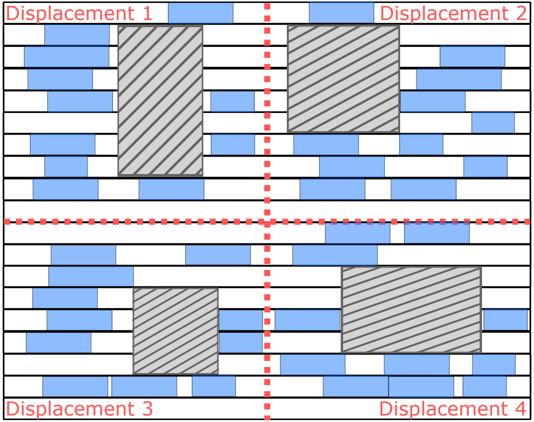
Circuit partitioning using k-d tree ($height = 2$ in the example)

Training data generation



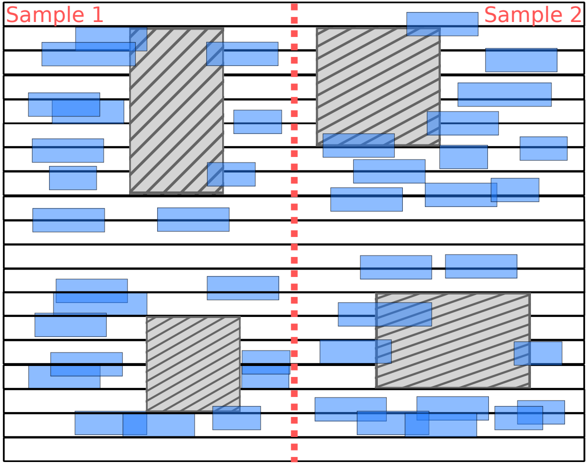
Circuit partitioning using k-d tree ($height = 2$ in the example)

Training data generation

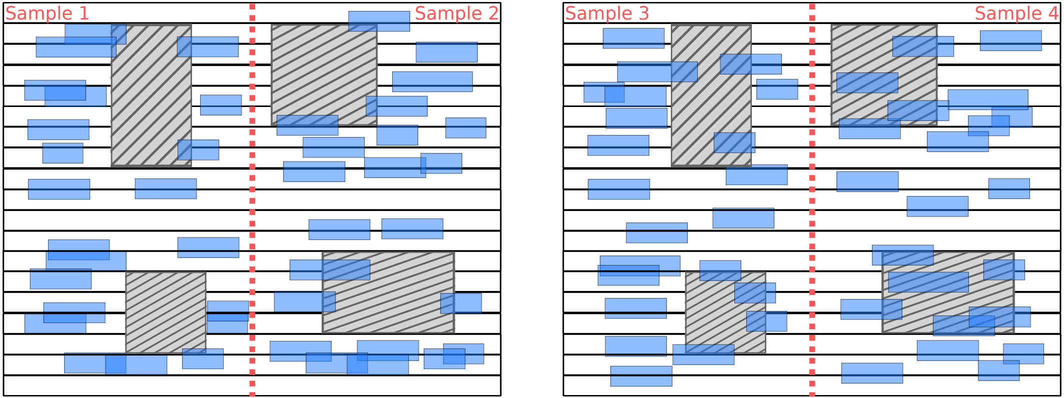


Circuit partitioning using k-d tree ($height = 2$ in the example)

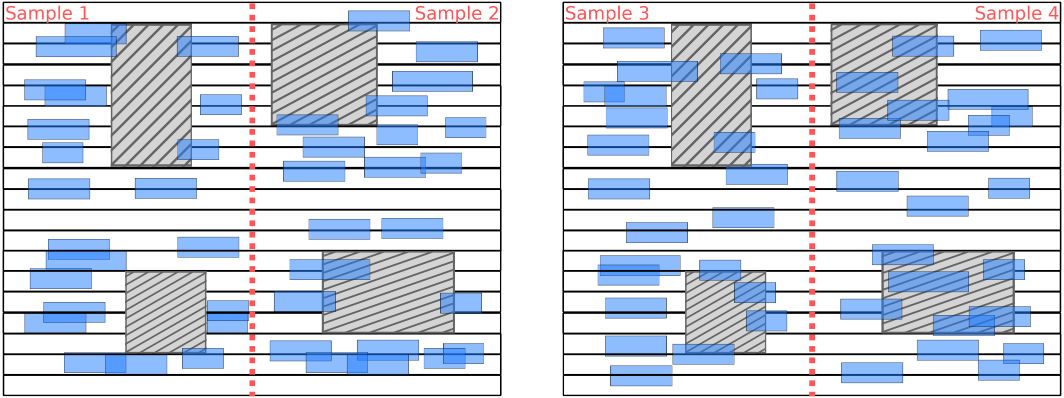
Handling partitions of different sizes



Handling partitions of different sizes

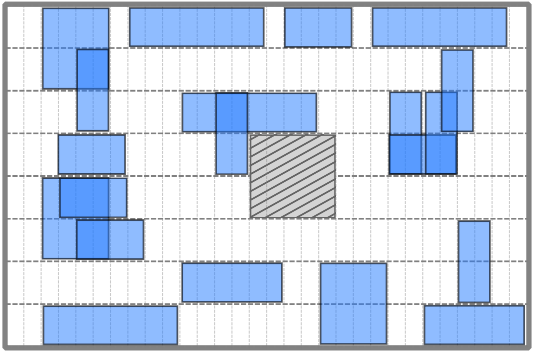


Handling partitions of different sizes



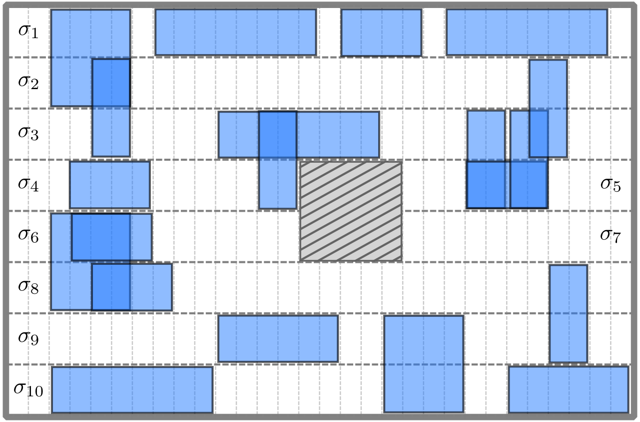
Actual values: $1 \leq height \leq 9$ and 1024 samples for each height.

Feature selection: non-convolutional ML models

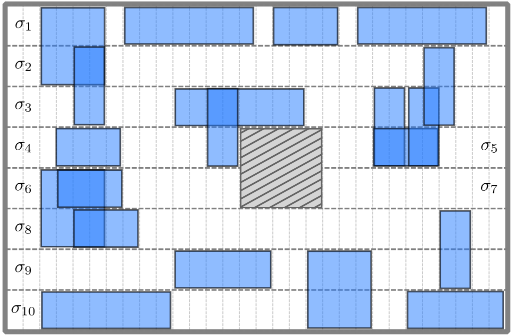


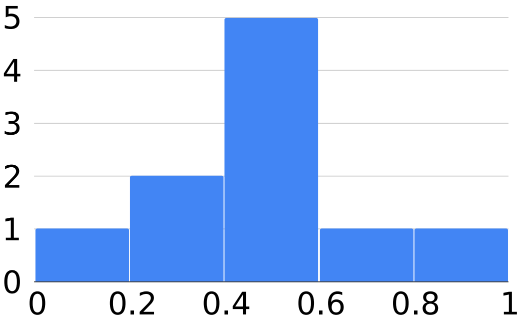
- $D = 0.49$
  - Density of the partition area
- $A = [64, 48]$
  - Area occupied by cells of each height

Feature selection: non-convolutional ML models

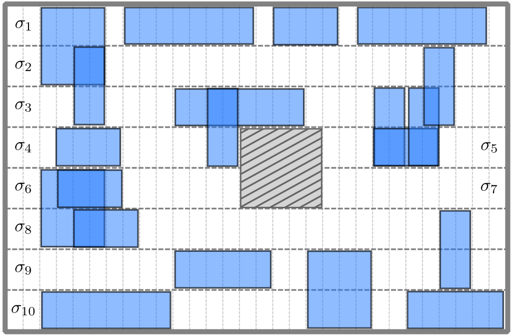


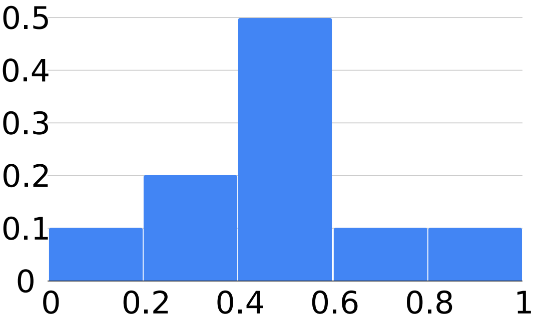
Feature selection: non-convolutional ML models



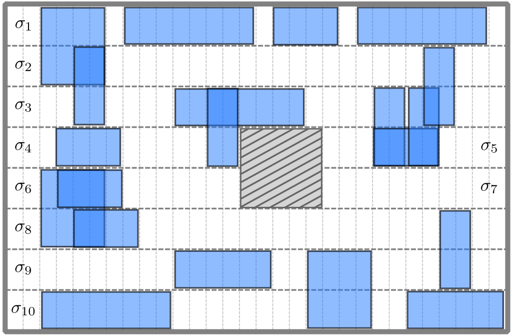


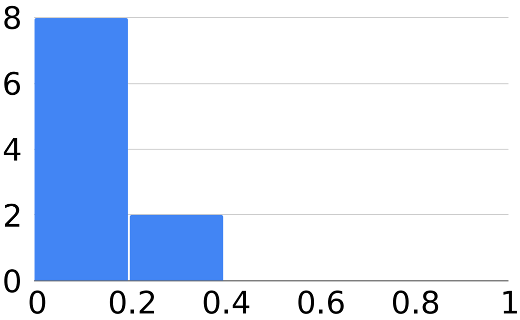
Feature selection: non-convolutional ML models



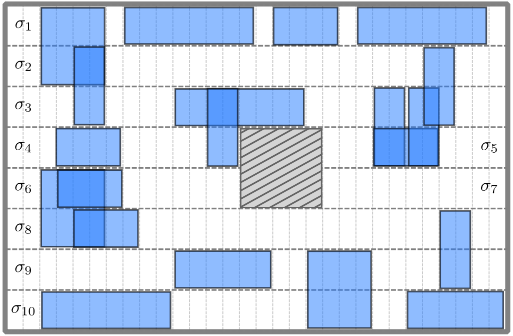


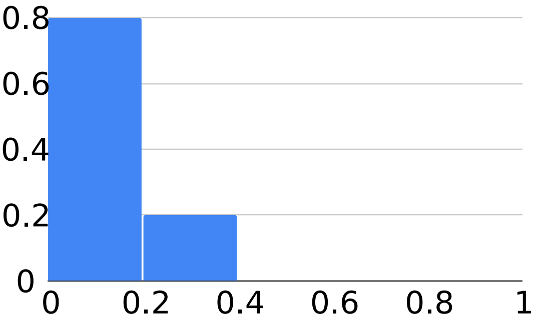
Feature selection: non-convolutional ML models



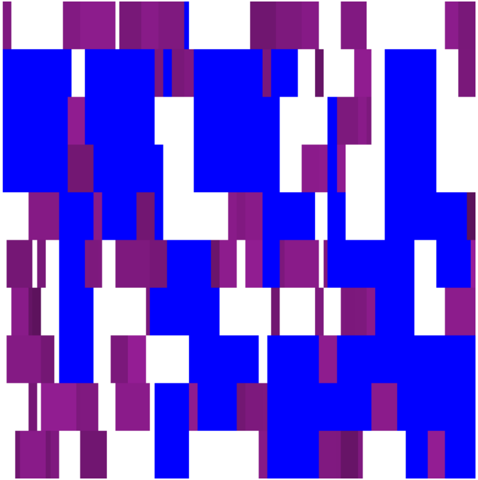


Feature selection: non-convolutional ML models



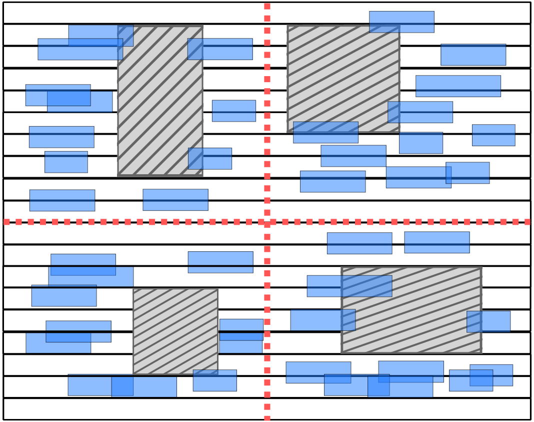


Feature selection: convolutional model



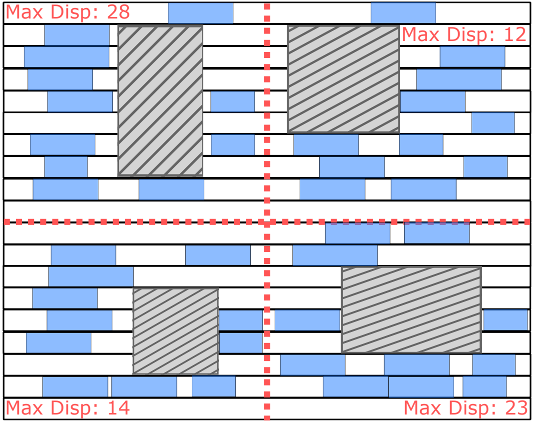
- Partition snapshot
  - Fixed cells: blue
  - Movable cells: shades of pink

Circuit partitioning strategy



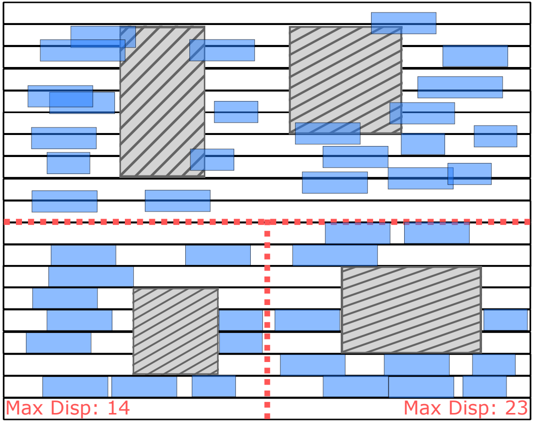
- Partitions circuit using a k-d tree data structure
- Each partition is legalized independently

Proposal: merge partitions that result in large displacement



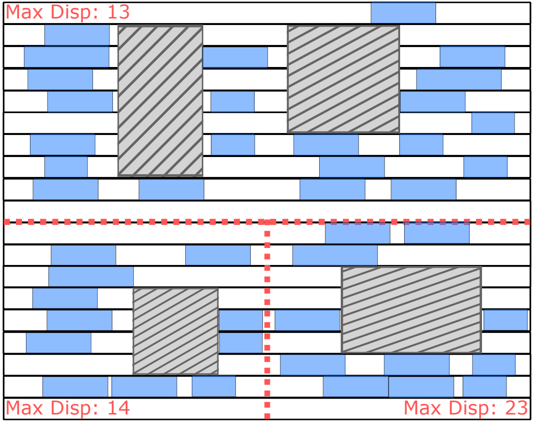
Example with displacement threshold of 25

Proposal: merge partitions that result in large displacement



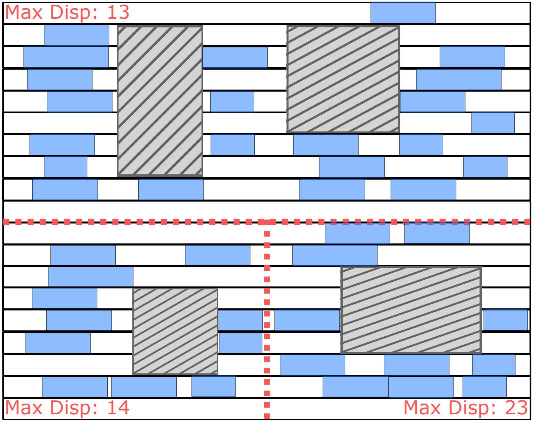
Example with displacement threshold of 25

Proposal: merge partitions that result in large displacement



Example with displacement threshold of 25

Circuit partitioning strategy



- Two ways of verifying partitions with large displacement:
  - Running the legalization algorithm
  - Using ML model to predict those partitions

Experimental infrastructure

Benchmarks

Training set

pci_bridge32_b_md3: 29K cells
fft_a_md2: 31K cells
des_perf_a_md1: 108K cells
des_perf_1: 113K cells
des_perf_b_md2: 113K cells
edit_dist_a_md2: 127K cells
fft_2_md2: 32K cells
fft_a_md3: 31K cells
des_perf_a_md2: 108K cells
des_perf_b_md1: 113K cells
edit_dist_1_md1: 131K cells
edit_dist_a_md3: 127K cells

Validation set

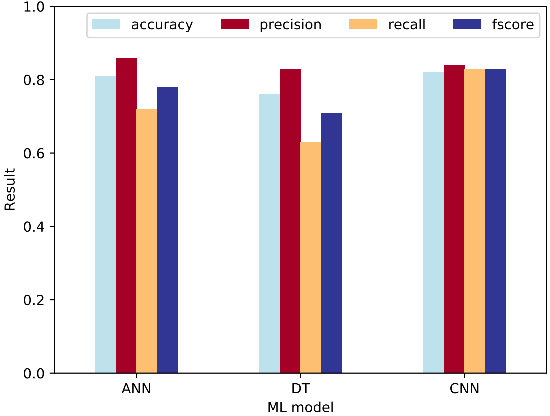
pci_bridge32_a_md1: 29K cells
pci_bridge32_a_md2: 29K cells
pci_bridge32_b_md1: 29K cells
pci_bridge32_b_md2: 29K cells

Test set

superblue18: 768M
superblue4: 795M
superblue16: 981M
superblue5: 1086M
superblue1: 1209M
superblue3: 1213M
superblue10: 1876M
superblue7: 1931M

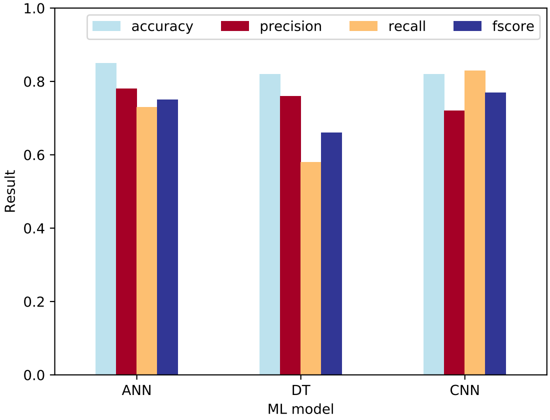
- Three ML models:
  - Artificial Neural Network (**ANN**)
  - Decision Tree (**DT**)
  - Convolutional Neural Network (**CNN**)
- Three different displacement thresholds: 5, 10 and 15 rows
- Metrics: accuracy, precision, recall, F-score

Evaluation of ML models: threshold of 5 rows



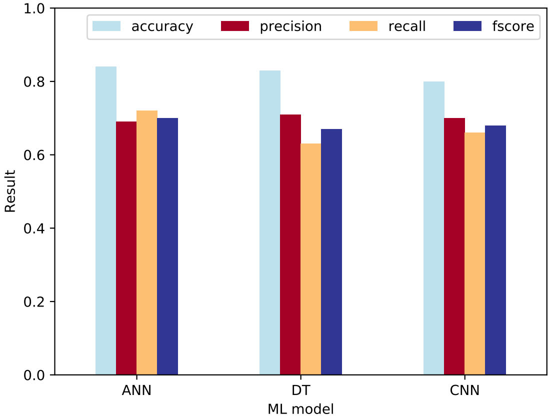
- Similar accuracy between ANN and CNN
- CNN achieves better F-score
- DT is the worst model

Evaluation of ML models: threshold of 10 rows



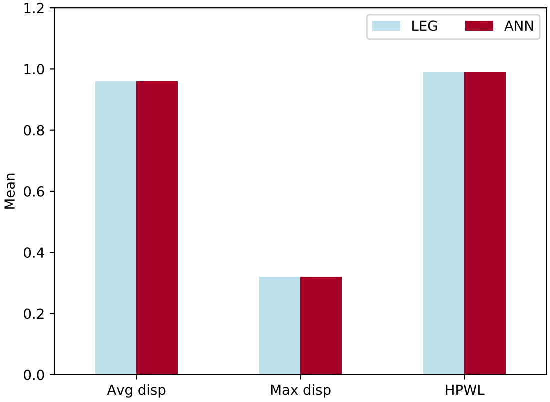
- Slight accuracy increase for ANN and DT
- Precision reduction on all models
- Recall affected only on DT

Evaluation of ML models: threshold of 15 rows



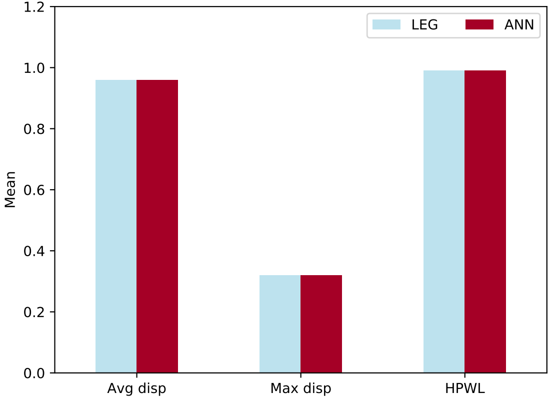
- Even more unbalanced data
- F-score reduction on both ANN and CNN
- Chosen model: **ANN**
    - Higher accuracy and F-score for threshold of 15 rows
    - Lower model complexity

Results of the pruning strategy: threshold of 5 rows



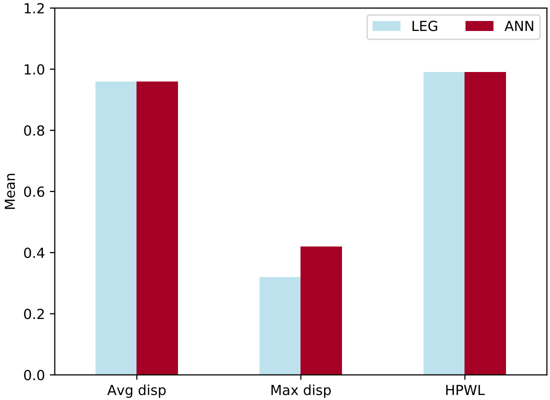
- Ratio of result using pruning strategy by original result (lower than 1 is better)
- Two ways of identifying partitions to merge:
  - Using the legalization algorithm (**LEG**)
  - Using ML model (**ANN**)

Results of the pruning strategy: threshold of 5 rows



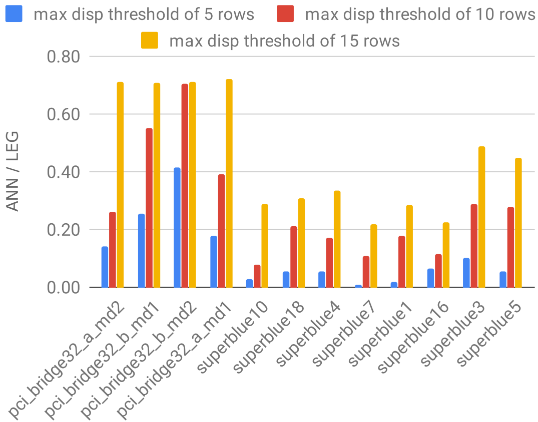
- ANN achieves same results as LEG
- Greater reduction on max displacement
- Avg displacement reduction was more relevant on small circuits
- No significant difference to threshold of 10 rows

Results of the pruning strategy: threshold of 15 rows



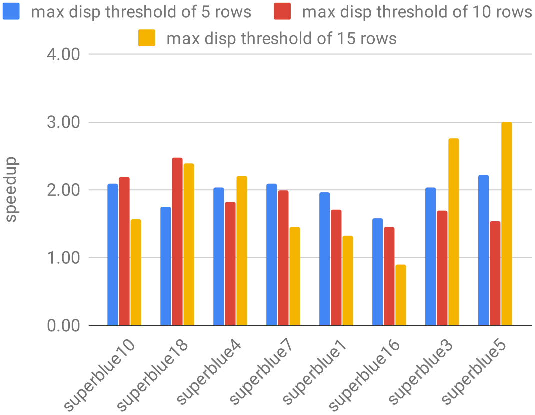
- Slight increase on max displacement for ANN
- Still great improvement compared to original solution

Number of calls to legalization algorithm



- Reduction for all circuits
- Greater reduction for larger circuits and threshold of 5 rows

Legalization time speedup



- Speedup is negligible for small circuits
- ANN is faster for almost all cases
- Higher speedup for lower threshold
- Exceptions are cases where ANN failed to identify some partitions

Conclusions and future work

- We evaluated **three ML models** to improve **predictability** of legalization algorithms
  - Artificial Neural Network
  - Decision Tree
  - Convolutional Neural Network
- Best model was used as pruning mechanism of partitioning strategy:
  - Greatly reduces maximum displacement
  - Avoids up to 99% of calls to legalization algorithm
- Future work:
  - Predicting resulting displacement itself
  - Evaluating on different legalization algorithms
  - Improvements on the CNN model to compensate its higher complexity

How deep learning can drive physical synthesis towards more predictable legalization

Questions?

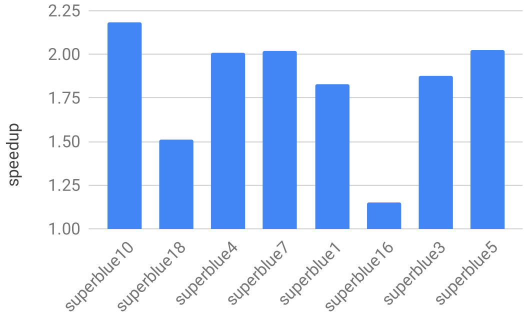
Renan Netto

renan.netto@posgrad.ufsc.br

UNIVERSIDADE FEDERAL DE SANTA CATARINA

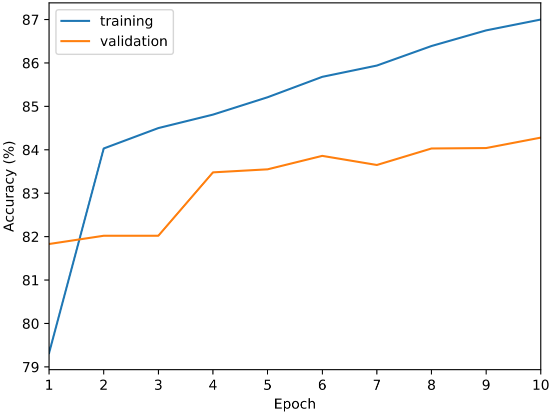
Ophidian

Speedup for best case



- LEG: threshold of 15 rows
- ANN: threshold of 5 rows
- ANN is still faster

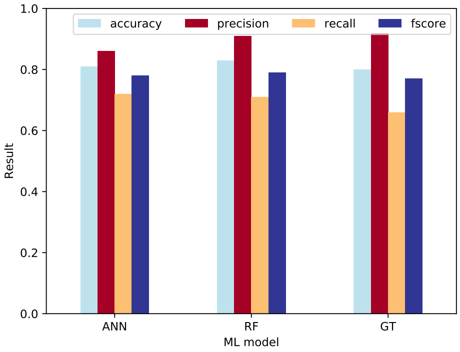
Accuracy along epochs



Experiments with other models: threshold of 5 rows

RF: Random forest

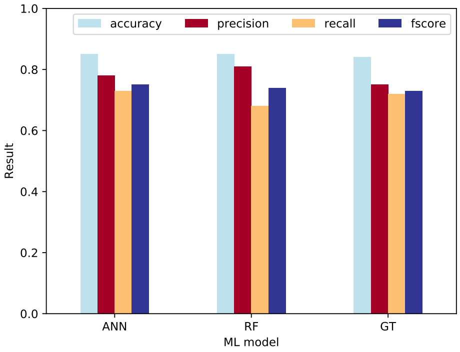
GT: Gradient boosted tree



Experiments with other models: threshold of 10 rows

RF: Random forest

GT: Gradient boosted tree



Experiments with other models: threshold of 15 rows

RF: Random forest

GT: Gradient boosted tree