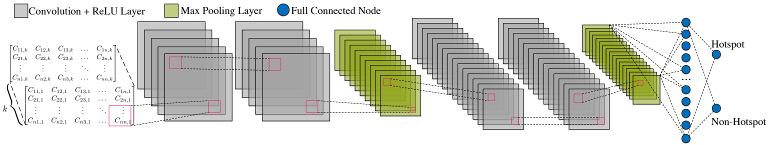
Automatic Layout Generation with Applications in Machine Learning Engine Evaluation

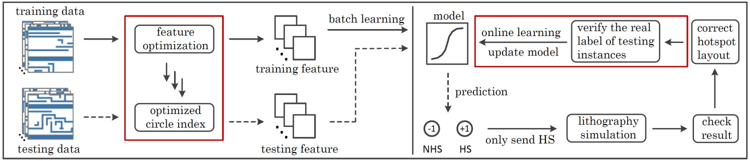
Haoyu Yang[1], Wen Chen[1], Piyush Pathak[2], Frank Gennari[2],
Ya-Chieh Lai[2], Bei Yu[1]

[1]The Chinese University of Hong Kong
[2]Cadence Design Systems, Inc.

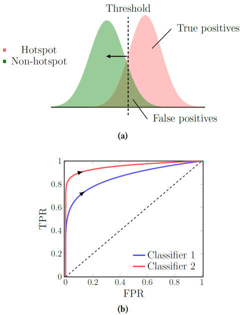
Exploding Machine Learning for Layout Printability Estimation



(a) BBL [Yang+,DAC'17]



(b) SMBoost [Zhang+,ICCAD'16]



(c) LithoROC [Ye+,ASPDAC'19]

Klayout Python Interface

An example of creating a layout with a single cell and single layer and puts one rectangle on that layer.

```
import pya
layout = pya.Layout()
top = layout.create_cell(“TOP”)
l1 = layout.layer(1, 0)
top.shapes(l1).insert(pya.Box(0, 0,
1000, 2000))
layout.write(“t.gds”)
```

- Layout: a rich set of methods to manipulate and query the layout hierarchy, the geometrical objects, the meta information and other features of the layout database.
- Cell: consists of a set of shape containers (called layers).
- Layer, Shape.

Metal Layer Generation

Global Configurations

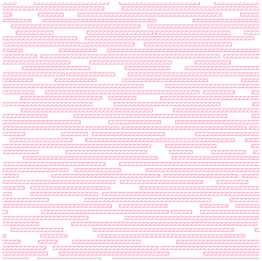
- Cell Name
- Total X: Cell bounding box size in x direction.
- Total Y: Cell bounding box size in y direction.

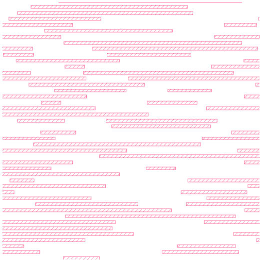
Wire Configurations

- Wire CD: Wire width.
- Track Pitch: Metal wire pitch.
- T2T Distance (min/max): Line-end to line-end distance of wires on single track.
- Wire Length (min/max)
- T2T Grid: Controls the unit size of T2T Distance.

Metal Layer Generation: Examples

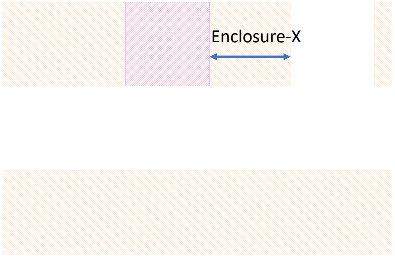
cellname	wire_cd	track_pitch	min_t2t	max_t2t	min_length	max_length	t2t_grid	total_x	total_y
test1	0.016	0.032	0.012	0.2	0.1	1	0.005	5	5
test2	0.016	0.032	0.2	1.0	0.1	1	0.012	5	5


(d) Small T2T (test1)


(e) Large T2T (test2)

Via Layer Generation
Via Specific Configurations

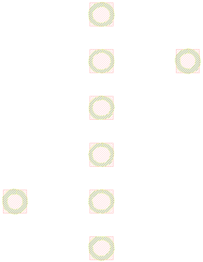
- Via X: Via size along x direction.
- Via Y: Via size along y direction.
- Via density: the probability of a via appearing at a candidate via position.
- Via-Metal-Enclosure: Vias should be away from line-ends by a certain distance.



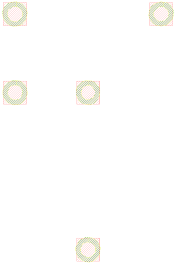
- Vias are relying on its upper and lower metal layers.
- Metal layers are configured as previous.
- Via creation flow.
  1. M_lower generation
  2. M_upper generation
  3. Check candidate via positions
  4. Place vias with according to enclosure and density constraints

Via Layer Generation: Examples & Simulation

cellname	via_x	via_x	density
via1	0.07	0.07	0.5
via2	0.07	0.07	0.3
via3	0.07	0.07	0.2



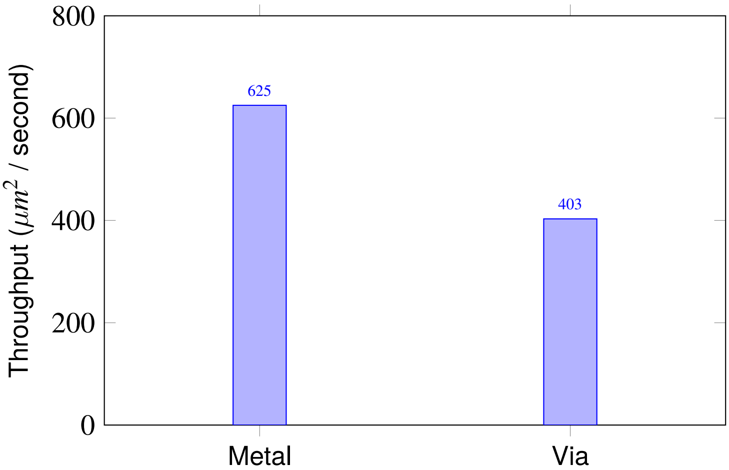
(f) High Density



(g) Medium Density

(h) Low Density

Performance Evaluation on Generation Tools

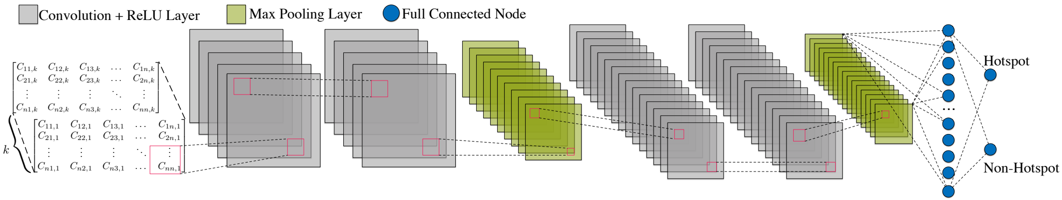


BBL-CNN Architecture [Yang+,DAC'17]

Layer	Kernel Size	Stride	Output Node #
conv1-1	3	1	$12 \times 12 \times 16$
conv1-2	3	1	$12 \times 12 \times 16$
maxpooling1	2	2	$6 \times 6 \times 16$
conv2-1	3	1	$6 \times 6 \times 32$
conv2-2	3	1	$6 \times 6 \times 32$
maxpooling2	2	2	$3 \times 3 \times 32$
fc1	N/A	N/A	250
fc2	N/A	N/A	2

Feature Tensor

- k-channel hyper-image
- Compatible with CNN
- Storage and computional efficiency



BBL-Recall The Training Procedure [Yang+,DAC'17]

- Minimize difference with ground truths

$$\mathbf{y}_n^* = [1, 0], \ \mathbf{y}_h^* = [0, 1]. \tag{1}$$

$$\mathbf{F} \in \begin{cases} \mathcal{N}, & \text{if } \mathbf{y}(0) > 0.5 \\ \mathcal{H}, & \text{if } \mathbf{y}(1) > 0.5 \end{cases} \tag{2}$$

BBL-Recall The Training Procedure [Yang+,DAC'17]

- Minimize difference with ground truths

$$\mathbf{y}_n^* = [1, 0],\ \mathbf{y}_h^* = [0, 1]. \tag{1}$$

$$\mathbf{F} \in \begin{cases} \mathcal{N}, & \text{if } \mathbf{y}(0) > 0.5 \\ \mathcal{H}, & \text{if } \mathbf{y}(1) > 0.5 \end{cases} \tag{2}$$

- Shifting decision boundary

$$\mathbf{F} \in \begin{cases} \mathcal{N}, & \text{if } \mathbf{y}(0) > 0.5 + \lambda \\ \mathcal{H}, & \text{if } \mathbf{y}(1) > 0.5 - \lambda \end{cases} \tag{3}$$

BBL-Recall The Training Procedure [Yang+,DAC'17]

- Minimize difference with ground truths

$$\mathbf{y}_n^* = [1, 0], \ \mathbf{y}_h^* = [0, 1]. \tag{1}$$

$$\mathbf{F} \in \begin{cases} \mathcal{N}, & \text{if } \mathbf{y}(0) > 0.5 \\ \mathcal{H}, & \text{if } \mathbf{y}(1) > 0.5 \end{cases} \tag{2}$$

- Shifting decision boundary (✗)

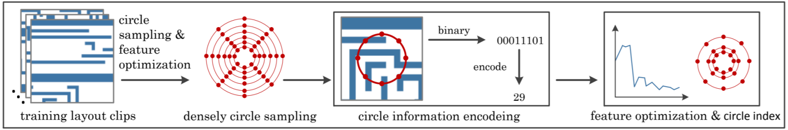
$$\mathbf{F} \in \begin{cases} \mathcal{N}, & \text{if } \mathbf{y}(0) > 0.5 + \lambda \\ \mathcal{H}, & \text{if } \mathbf{y}(1) > 0.5 - \lambda \end{cases} \tag{3}$$

- Biased ground truth

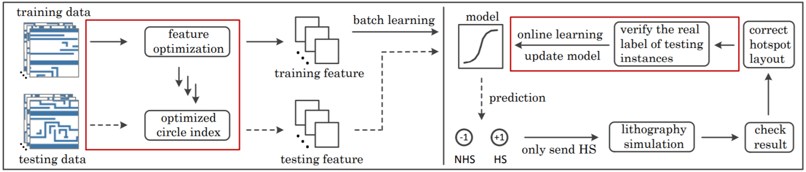
$$\mathbf{y}_n^* = [1 - \epsilon, \epsilon] \tag{4}$$

Smooth Boosting [Zhang+,ICCAD'16]

- Firstly, we densely sample the circles from the training data.
- Secondly, we optimally select circles by DP algorithm.
- Thirdly, we use the obtained circle index to extract features.



Smooth Boosting [Zhang+,ICCAD'16]



A New Hotspot Detection Framework

- New performance metric: runtime & performance trade-off
- Feature optimization based on mutual information
- Online learning

Optimizing Receiver Operating Characteristic [Ye+,ASPDAC'19]

The AUC objective:

$$\mathcal{L}_\Phi(f) = \frac{1}{N_+N_-} \sum_{i=1}^{N_+} \sum_{j=1}^{N_-} \Phi\left(f\left(\mathbf{x}_i^+\right) - f\left(\mathbf{x}_j^-\right)\right).$$

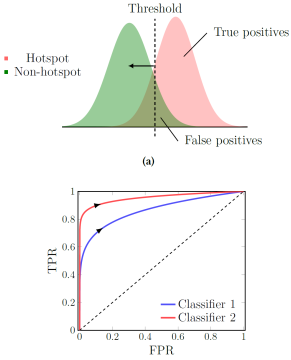
Approximation candidates:

PSL $\Phi_{\text{PSL}}(z) = (1-z)^2$

PHL $\Phi_{\text{PHL}}(z) = \max(1-z, 0)$

PLL $\Phi_{\text{PLL}}(z) = \log(1+\exp(-\beta z))$

R $\Phi_{\text{R}*}(z) = \begin{cases} -(z-\gamma)^p, & \text{if } z > \gamma \\ 0, & \text{otherwise} \end{cases}$



(a)
(b)

The Dataset

Via: Total 10403 2000$\times$2000 via patterns, simulated with industrial simulator.

- Training Set
- 2774 hotspots
- 5226 non-hotspots

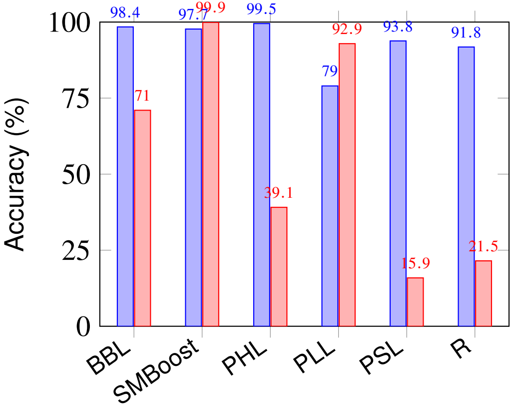
- Testing Set
- 841 hotspots
- 1562 non-hotspots

Metal: Merged ICCAD2012 CAD Contest Benchmark.

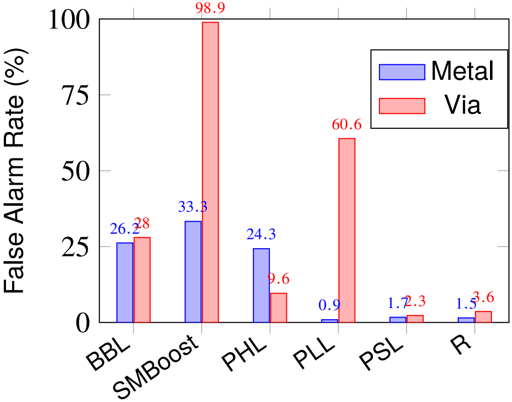
- Training Set
- 1204 hotspots
- 17096 non-hotspots

- Testing Set
- 2524 hotspots
- 13503 non-hotspots

Machine Learning Engine Evaluation



(i) Accuracy



(j) False Alarm

Thank You