Automatic Layout Generation with Applications in Machine Learning Engine Evaluation

Haoyu Yang\textsuperscript{1}, Wen Chen\textsuperscript{1}, Piyush Pathak\textsuperscript{2}, Frank Gennari\textsuperscript{2}, Ya-Chieh Lai\textsuperscript{2}, Bei Yu\textsuperscript{1}

\textsuperscript{1}The Chinese University of Hong Kong
\textsuperscript{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.

```python
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

<table>
<thead>
<tr>
<th>cellname</th>
<th>wire_cd</th>
<th>track_pitch</th>
<th>min_t2t</th>
<th>max_t2t</th>
<th>min_length</th>
<th>max_length</th>
<th>t2t_grid</th>
<th>total_x</th>
<th>total_y</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1</td>
<td>0.016</td>
<td>0.032</td>
<td>0.012</td>
<td>0.2</td>
<td>0.1</td>
<td>1</td>
<td>0.005</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>test2</td>
<td>0.016</td>
<td>0.032</td>
<td>0.2</td>
<td>1.0</td>
<td>0.1</td>
<td>1</td>
<td>0.012</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

(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

<table>
<thead>
<tr>
<th>cellname</th>
<th>via_x</th>
<th>via_y</th>
<th>density</th>
</tr>
</thead>
<tbody>
<tr>
<td>via1</td>
<td>0.07</td>
<td>0.07</td>
<td>0.5</td>
</tr>
<tr>
<td>via2</td>
<td>0.07</td>
<td>0.07</td>
<td>0.3</td>
</tr>
<tr>
<td>via3</td>
<td>0.07</td>
<td>0.07</td>
<td>0.2</td>
</tr>
</tbody>
</table>

(f) High Density
(g) Medium Density
(h) Low Density
Performance Evaluation on Generation Tools

Throughput ($\mu m^2 / \text{second}$)

- Metal: 625
- Via: 403
Feature Tensor

- $k$-channel hyper-image
- Compatible with CNN
- Storage and computational efficiency

<table>
<thead>
<tr>
<th>Layer</th>
<th>Kernel Size</th>
<th>Stride</th>
<th>Output Node #</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv1-1</td>
<td>3</td>
<td>1</td>
<td>$12 \times 12 \times 16$</td>
</tr>
<tr>
<td>conv1-2</td>
<td>3</td>
<td>1</td>
<td>$12 \times 12 \times 16$</td>
</tr>
<tr>
<td>maxpooling1</td>
<td>2</td>
<td>2</td>
<td>$6 \times 6 \times 16$</td>
</tr>
<tr>
<td>conv2-1</td>
<td>3</td>
<td>1</td>
<td>$6 \times 6 \times 32$</td>
</tr>
<tr>
<td>conv2-2</td>
<td>3</td>
<td>1</td>
<td>$6 \times 6 \times 32$</td>
</tr>
<tr>
<td>maxpooling2</td>
<td>2</td>
<td>2</td>
<td>$3 \times 3 \times 32$</td>
</tr>
<tr>
<td>fc1</td>
<td>N/A</td>
<td>N/A</td>
<td>250</td>
</tr>
<tr>
<td>fc2</td>
<td>N/A</td>
<td>N/A</td>
<td>2</td>
</tr>
</tbody>
</table>
Minimize difference with ground truths

\[ y_n^* = [1, 0], \ y_h^* = [0, 1]. \]  

\[ F \in \begin{cases} 
  \mathcal{N}, & \text{if } y(0) > 0.5 \\
  \mathcal{H}, & \text{if } y(1) > 0.5 
\end{cases} \]
BBL-Recall The Training Procedure [Yang+, DAC’17]

- Minimize difference with ground truths

\[ y^*_n = [1, 0], \quad y^*_h = [0, 1]. \quad (1) \]

\[ F \in \begin{cases} \mathcal{N}, & \text{if } y(0) > 0.5 \\ \mathcal{H}, & \text{if } y(1) > 0.5 \end{cases} \quad (2) \]

- Shifting decision boundary

\[ F \in \begin{cases} \mathcal{N}, & \text{if } y(0) > 0.5 + \lambda \\ \mathcal{H}, & \text{if } y(1) > 0.5 - \lambda \end{cases} \quad (3) \]
BBL-Recall The Training Procedure [Yang+, DAC’17]

- Minimize difference with ground truths

\[ y_n^* = [1, 0], \ y_h^* = [0, 1]. \] (1)

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

- Shifting decision boundary \((X)\)

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

- Biased ground truth

\[ y_n^* = [1 - \epsilon, \epsilon] \] (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
The AUC objective:
\[
\mathcal{L}_\Phi(f) = \frac{1}{N_+ N_-} \sum_{i=1}^{N_+} \sum_{j=1}^{N_-} \Phi \left( f(x_i^+) - f(x_j^-) \right).
\]

Approximation candidates:

- **PSL** \( \Phi_{PSL}(z) = (1 - z)^2 \)
- **PHL** \( \Phi_{PHL}(z) = \max(1 - z, 0) \)
- **PLL** \( \Phi_{PLL}(z) = \log(1 + \exp(-\beta z)) \)
- **R** \( \Phi_{R*}(z) = \begin{cases} -(z - \gamma)^p, & \text{if } z > \gamma \\ 0, & \text{otherwise} \end{cases} \)
The Dataset

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

- 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

(ii) False Alarm Rate

Accuracy (%)

<table>
<thead>
<tr>
<th></th>
<th>BBL</th>
<th>SMBoost</th>
<th>PHL</th>
<th>PLL</th>
<th>PSL</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>98.4</td>
<td>97.7</td>
<td>99.9</td>
<td>99.5</td>
<td>92.9</td>
<td>93.8</td>
</tr>
<tr>
<td></td>
<td>71</td>
<td>59.1</td>
<td>79</td>
<td>35.9</td>
<td>31.5</td>
<td></td>
</tr>
</tbody>
</table>

False Alarm Rate (%)

<table>
<thead>
<tr>
<th></th>
<th>BBL</th>
<th>SMBoost</th>
<th>PHL</th>
<th>PLL</th>
<th>PSL</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>26.28</td>
<td>33.3</td>
<td>24.3</td>
<td>9.6</td>
<td>60.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.23</td>
<td>1.36</td>
<td></td>
<td></td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

Metal

Via
Thank You