Hotspot Detection Using Squish-Net

Haoyu Yang, The Chinese University of Hong Kong (Hong Kong SAR); Piyush Pathak, Frank Gennari, Ya-Chieh Lai, Cadence Design Systems (United States); Bei Yu, The Chinese University of Hong Kong (Hong Kong SAR).

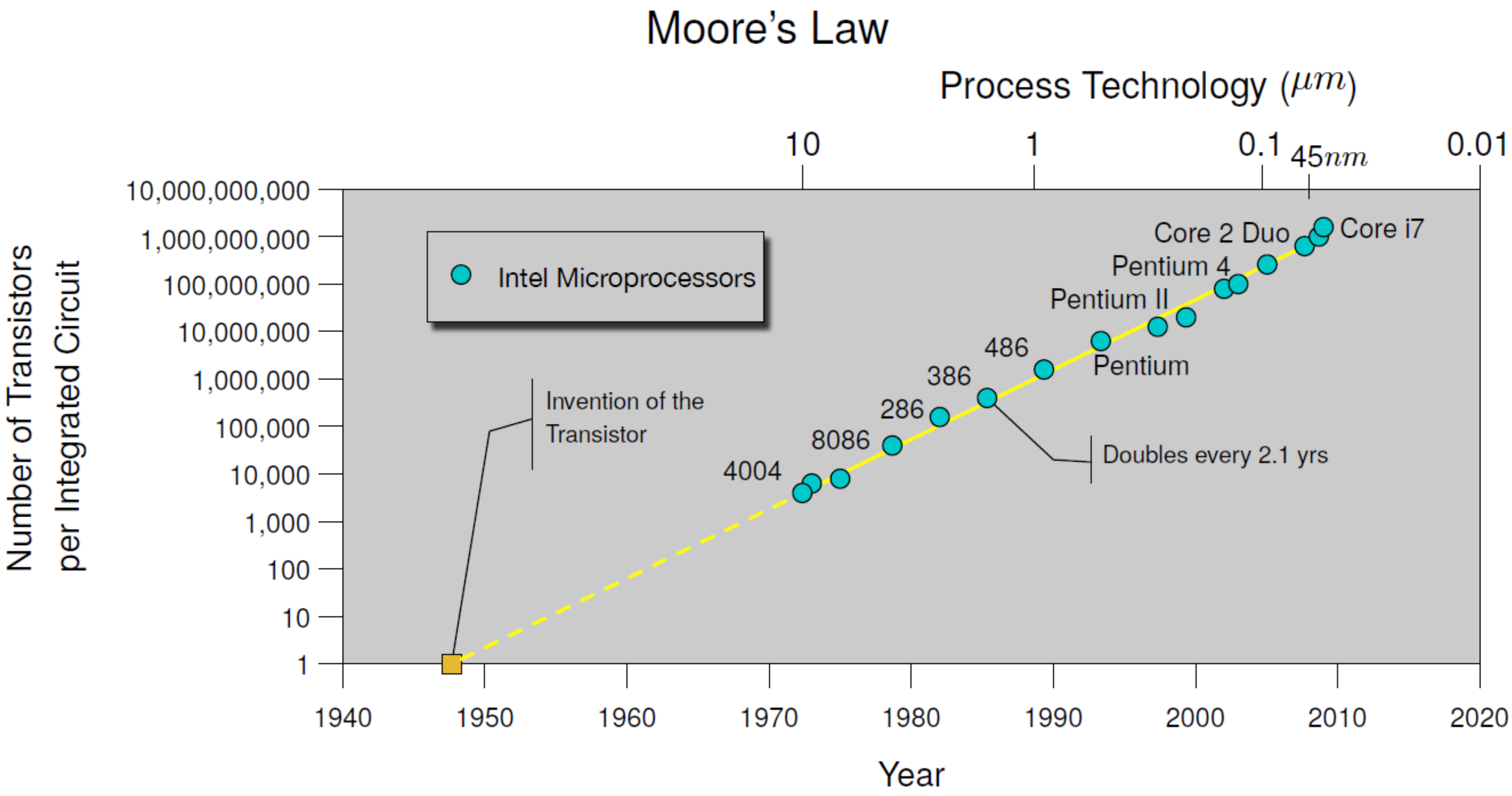
cādence®

Outline

- Introduction
- Adaptive Squish Pattern
- The Squish-Net Framework
- Results
- Conclusion

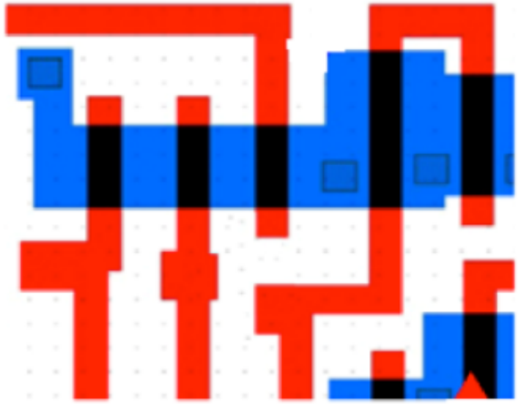
Introduction

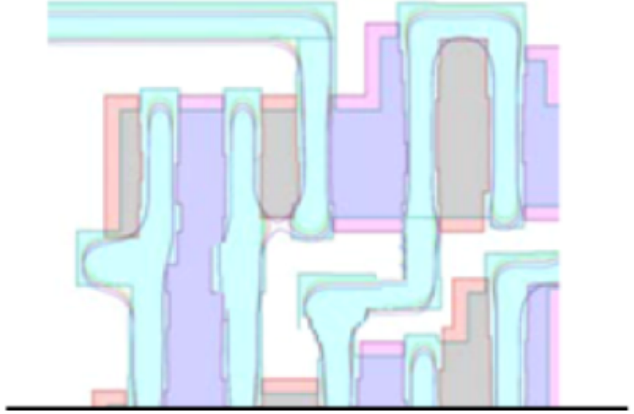
Moore’s Law Extreme Scaling

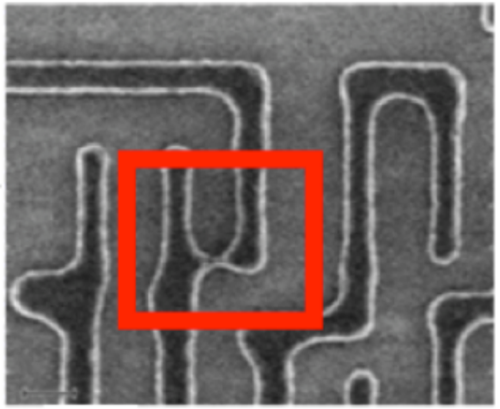


cādence®

Lithography Proximity Effect



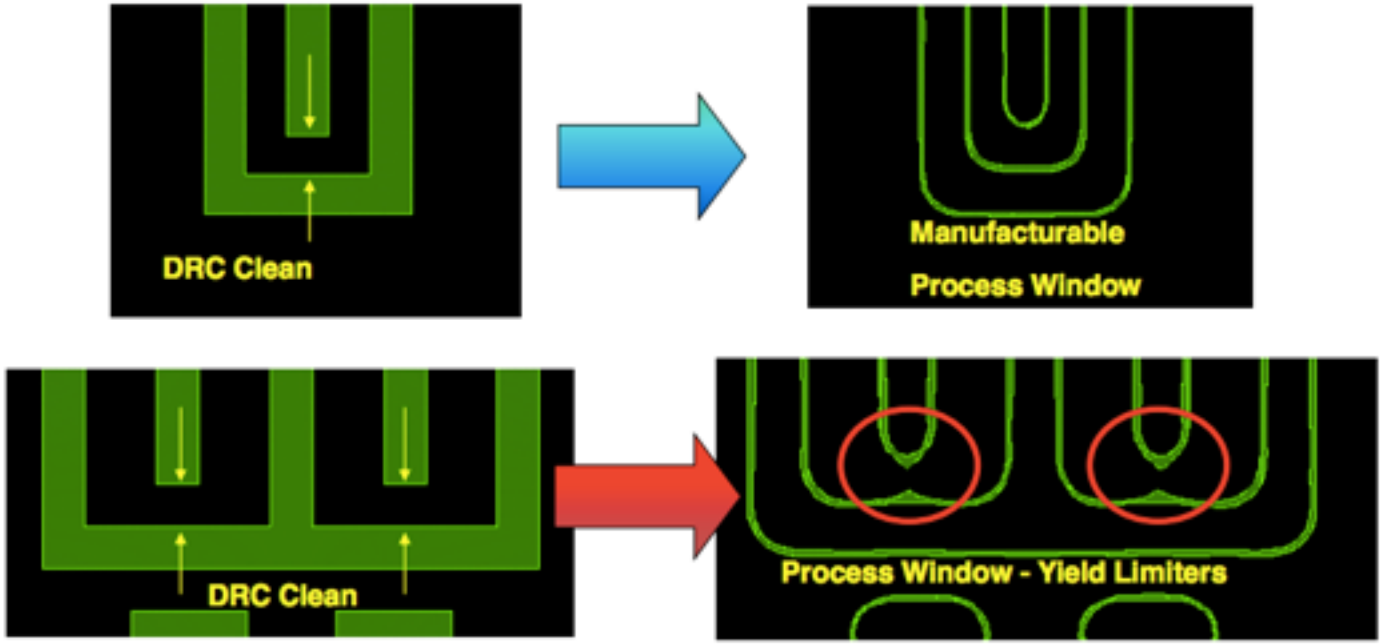




- What you see $\neq$ what you get
- Diffraction information loss
- RET: OPC, SRAF, MPL
- Worse on designs under 10nm or beyond

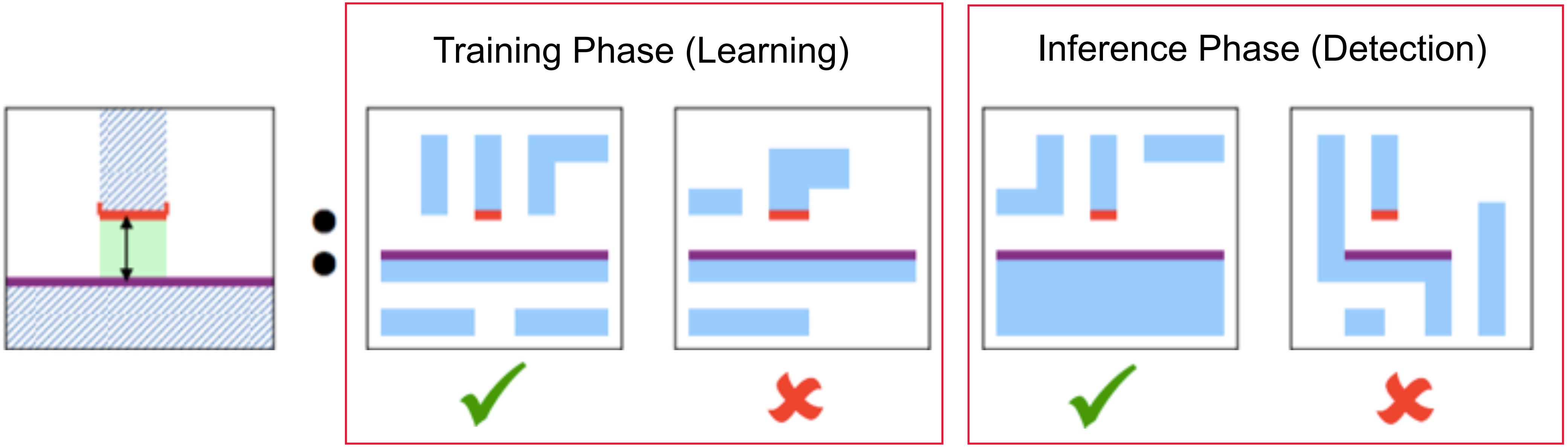
Hospot Detection

- Design-process weak points
  - called “hotspots”
  - source of systematic yield loss.
- Conventional verification methods
  - Process simulations
    - Accurate
    - Can detect new/unknown ”hotspots”
    - Long runtimes
  - Pattern Matching
    - Fast runtimes
    - Less accurate
    - Cannot detect new/unknown “hotspots”



Source: J. Yang et al. “DRCPlus in a router: automatic elimination of lithography hotspots using 2D pattern detection and correction”, Proc. SPIE, 76410 (2010)

Supervised Machine Learning in DFM



Source: V. Dai et al. “Developing DRC plus rules through 2D pattern extraction and clustering techniques”, Proc. SPIE, 7275 (2009)

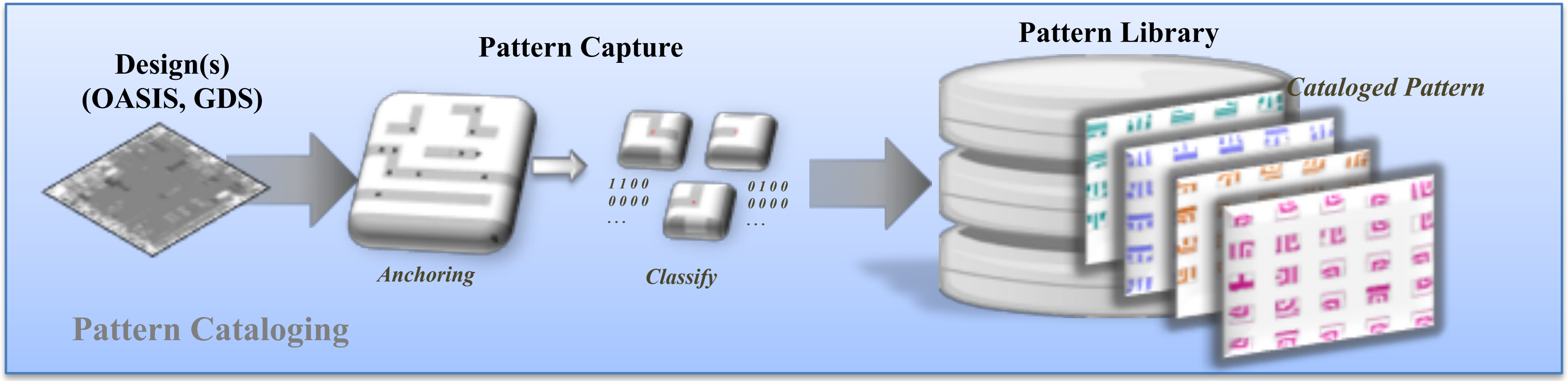
- Application of Machine Learning (ML) for hotspot detection
  - Requirements:
    - Can detect new/unknown hotspots
    - Reasonable detection accuracy
- Problem Statement
  - Build predictive models trained over the known hotspots to detect hotspots on new design data

cādence®

Adaptive Squish Pattern

cādence®

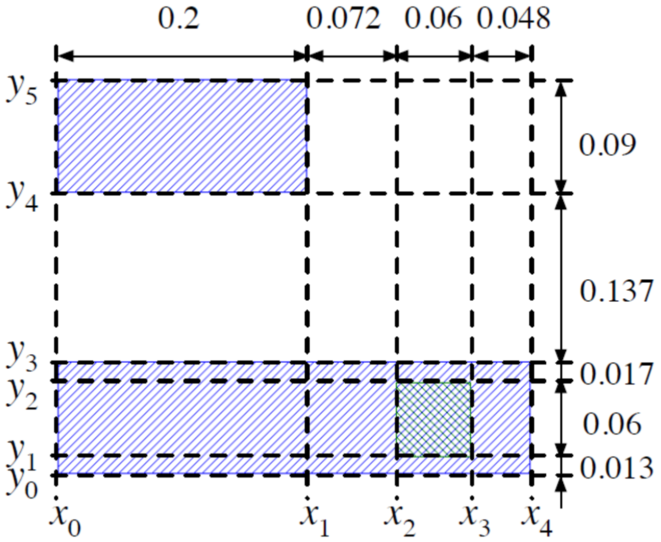
Data Representation: Pattern Catalog



Source: J. Kim et al. “Advanced DFM Analytics with Machine Learning on Layout Hotspot Prediction”, DAC, 2018 (WIP)

1. Systematically window across entire design
2. In every fixed window, centered on the corner or center of anchor layer shape, identify <u>every</u> pattern that exists in that design and target layer(s) (with <u>dimensions</u>)
3. Store a full catalog of all patterns with dimensions!

Adaptive Squish [Yang+, ASPDAC'19]



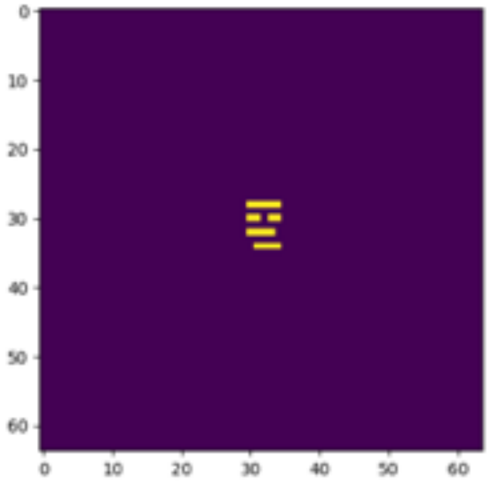
$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 3 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

$$\delta_x = \begin{bmatrix} 0.2 & 0.072 & 0.06 & 0.048 \end{bmatrix},$$

$$\delta_y = \begin{bmatrix} 0.013 & 0.06 & 0.017 & 0.137 & 0.09 \end{bmatrix}.$$

- Lossless
- Storage efficient
- Different topology dimensions for fixed pattern window size

Padding does not solve the problem



Adaptive Squish

Instead of padding, we repeat certain rows or columns of squish topologies

$$\mathbf{T}' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 3 & 3 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

- Which rows/columns are to be repeated/duplicated?
  - In machine learning, if some entries of the input are too large/small, there will be bias related to those entries.
  - Subtract RGB means in conventional image classification tasks.
  - Duplicate rows/columns with larger deltas.

$$\min_{\mathbf{s}} ||\boldsymbol{\delta}'||_\infty$$

$$\text{s.t. } \delta_i' = \delta_i / s_i, \forall i,$$

$$s_i \in \mathbb{Z}^+, \forall i,$$

$$\sum_i s_i = d.$$

cadence

A Greedy Solution for Adaptive Squish

$$3 \times 3 \rightarrow 3 \times 6$$

Algorithm 1 Obtaining adaptive squish patterns with a greedy procedure.

Input: T, δ, a, d_0, d;
Output: T, δ;

```
while d_0 < d do
    s ← 1 ∈ ℝ^{d_0}, i ← argmax_i{δ_i | i = 1, 2, ..., d_0 − 1};
    s_i ← 2, δ_i ← δ_i/2, ∀i;
    δ ← RepeatElements(δ, s, 1);
    T ← RepeatElements(T, s, a);
    d_0 ← d_0 + 1;
end while
```

Pattern $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$

$$\text{deltaX} = \begin{bmatrix} 28 & 18 & 2 \end{bmatrix}$$
$$\text{deltaY} = \begin{bmatrix} 16 & 16 & 16 \end{bmatrix}$$

Result $\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$

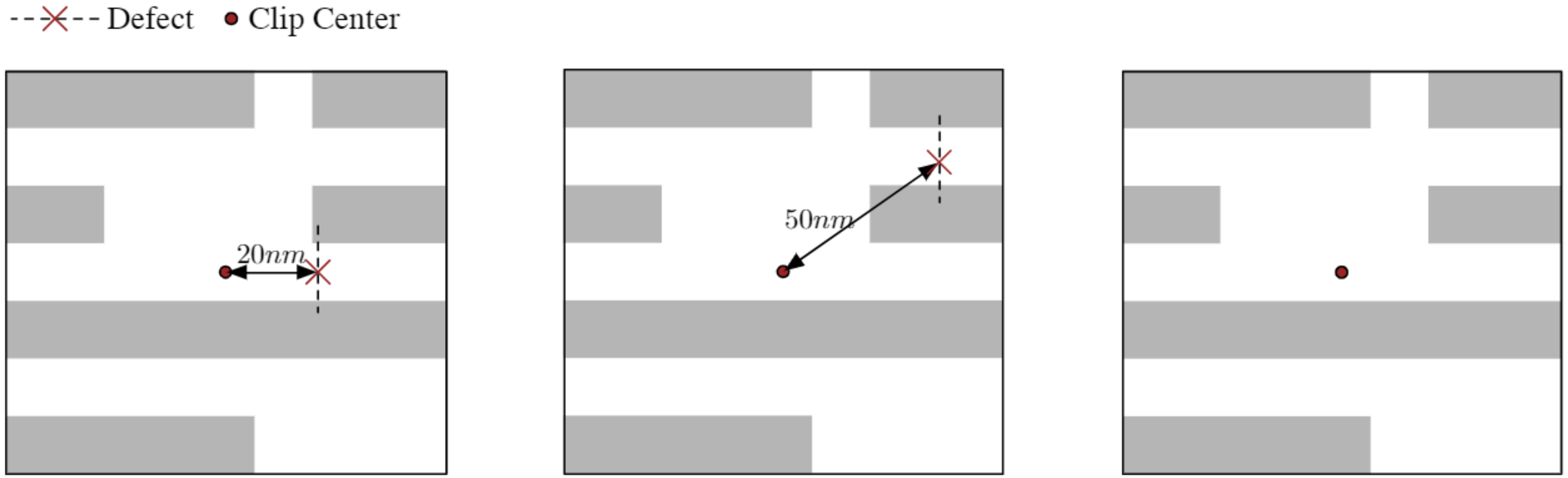
$$\text{deltaX} = \begin{bmatrix} 7 & 7 & 14 & 9 & 9 & 2 \end{bmatrix}$$
$$\text{deltaY} = \begin{bmatrix} 16 & 16 & 16 \end{bmatrix}$$

cadence

The Squish-Net Framework

Pattern Labeling

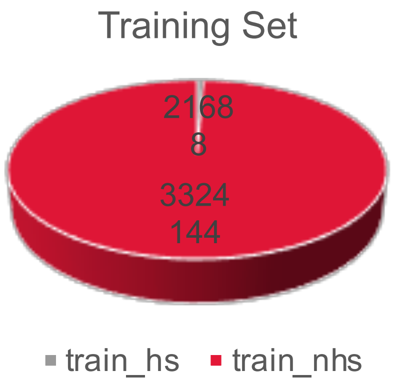
- Should a pattern be labeled as hotspot as long as defects occur?
  - Hotspots are caused by a larger pattern context.
  - Risky if defects are located near the boundaries of the clip.
- C2C distance (t_c2c=48nm)

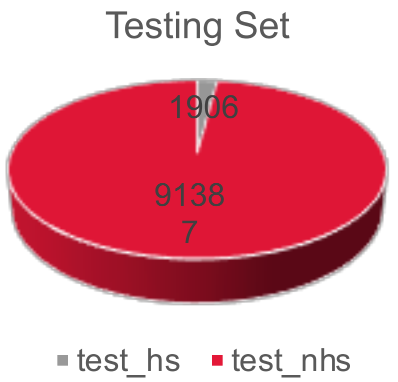


(a) Hotspot (b) Non-hotspot (c) Non-hotspot

Imbalance-aware Training

- Statistics of the dataset
  - Highly imbalanced
  - Overfitting





- Force balanced batch [Yang+, TCAD]

Algorithm 3 Batch Biased-learning

Require: $\mathbf{W}$, λ, α, k, $\mathbf{y}_h^*$, $\mathbf{y}_n^*$, β;

Initialize parameters, $\mathbf{y}_h^* \leftarrow [0, 1]$;

while not stop condition **do**

  Sample m non-hotspot instances $\{\mathbf{N}_1, \mathbf{N}_2, ..., \mathbf{N}_m\}$;

  Sample m hotspot instances $\{\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_m\}$;

  Calculate average loss of non-hotspot samples l_n with ground truth $[1, 0]$;

  $\mathbf{y}_n^* \leftarrow [1 - \epsilon(l_n), \epsilon(l_n)]$;

  for $i \leftarrow 1, 2, ..., m$ **do**

    $\mathcal{G}_{h,i} \leftarrow \texttt{backprop}(\mathbf{H}_i)$;

    $\mathcal{G}_{n,i} \leftarrow \texttt{backprop}(\mathbf{N}_i)$;

  end for

  Calculate gradient $\bar{\mathcal{G}} \leftarrow \frac{1}{2m}\sum_{i=1}^{m}(\mathcal{G}_{h,i} + \mathcal{G}_{n,i})$;

  Update weight $\mathbf{W} \leftarrow \mathbf{W} - \lambda\bar{\mathcal{G}}$;

  if $j \mod k = 0$ **then**

    $\lambda \leftarrow \alpha\lambda$, $j \leftarrow 0$;

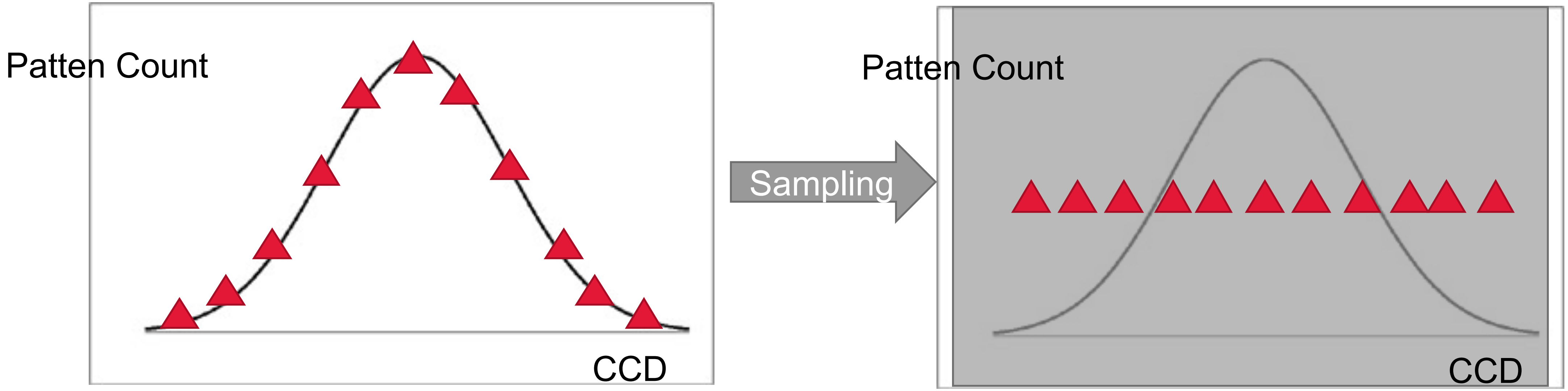
  end if

end while

cadence

Imbalance-aware Training

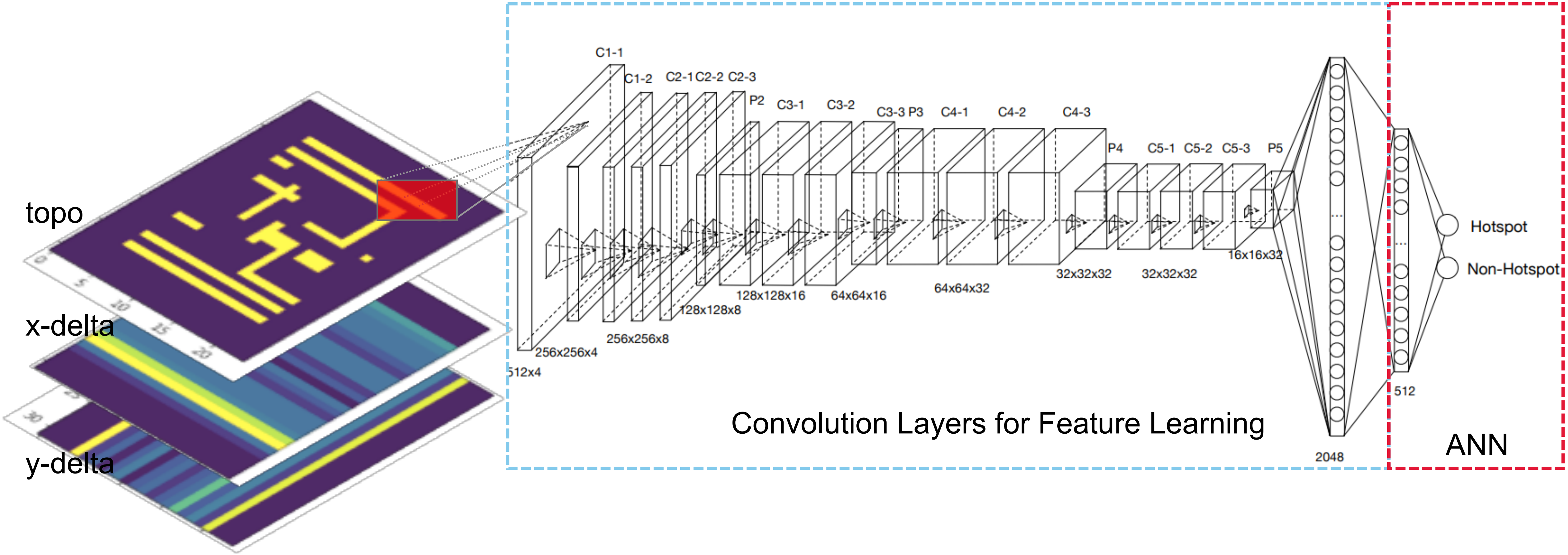
- Pre-sampling according to CCD score.



- CCD- Count of Critical Dimension

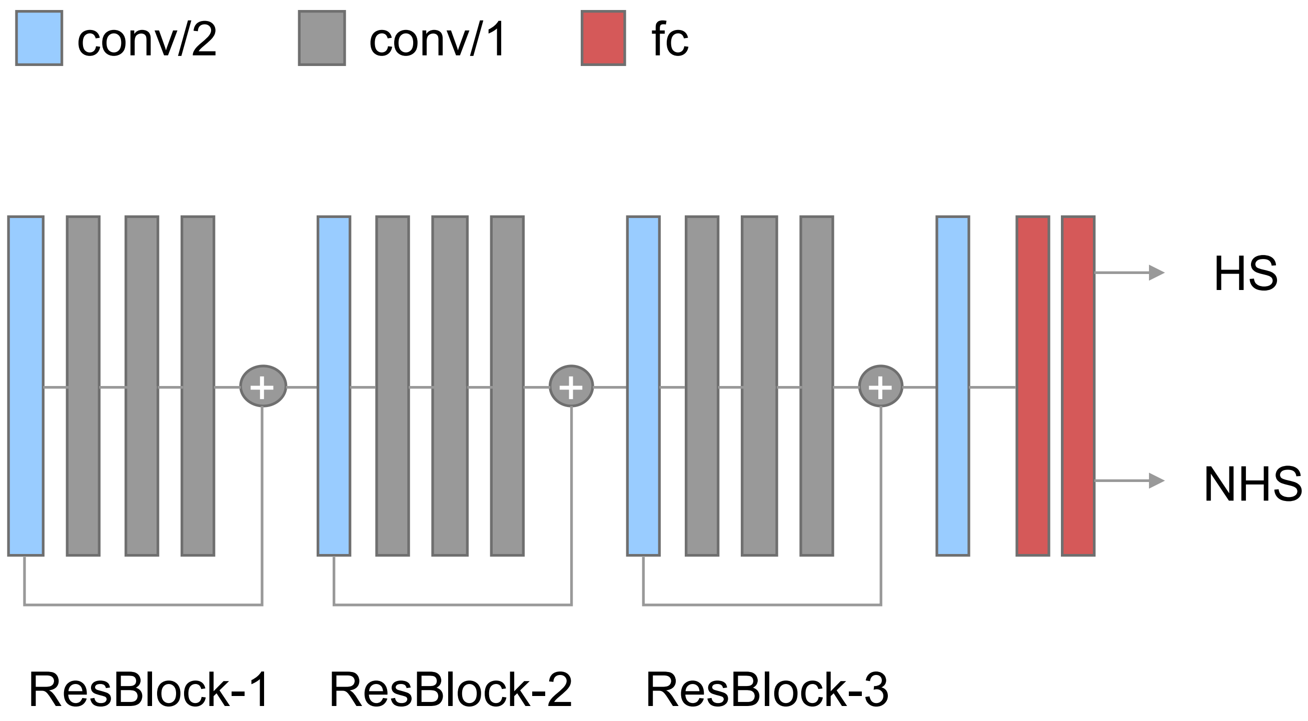
Pathak, Piyush, et al. "Methodology to extract, data mine and score geometric constructs from physical design layouts for analysis and applications in semiconductor manufacturing." *Design-Process-Technology Co-optimization for Manufacturability X*. Vol. 9781. International Society for Optics and Photonics, 2016.

Legacy CNN Structure



cadence®

The Network Architecture



He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

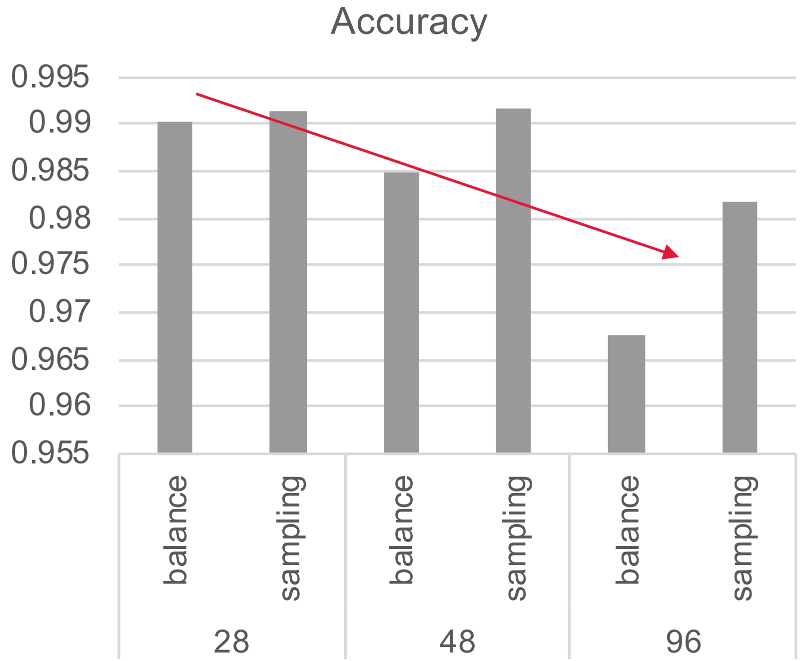
Yang, Haoyu, et al. "Lithography hotspot detection: From shallow to deep learning." *2017 30th IEEE International System-on-Chip Conference (SOCC)*. IEEE, 2017.

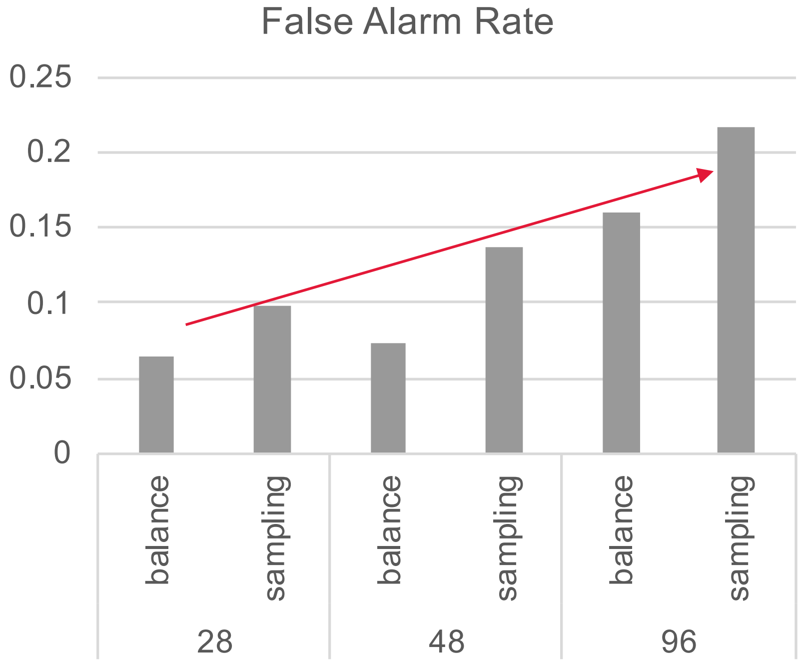
- ResBlock: Allows gradients to be easily backpropagated towards earlier layers.
- NoPool: Replace all pooling layers with strided convolutions to alleviate information loss.

Results

cadence

Study of C2C Threshold

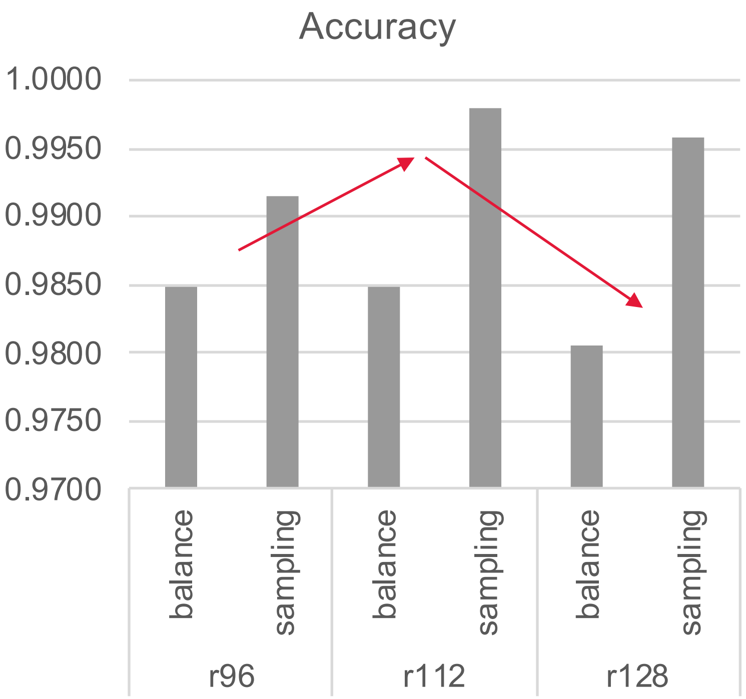


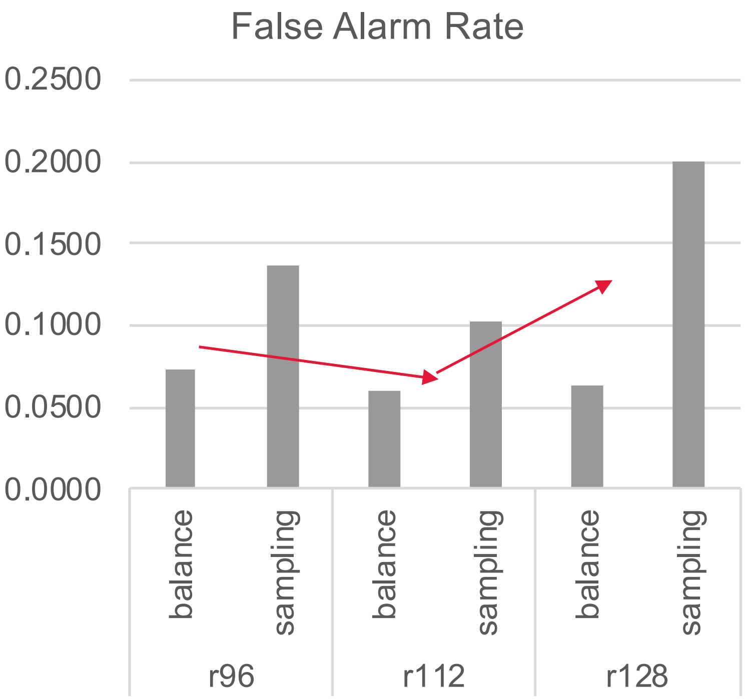


- Smaller C2C threshold: Low label noises, high imbalanced distribution.
- Larger C2C threshold: High label noises, even data distribution.

cādence®

Study of Pattern Radius





- Best performance occurs at r=112nm.
- Larger radius gives more pattern context information while making the imbalanced problem worse.

Conclusion

- Adaptive squish pattern to fit machine learning engines
  - Fixed representation size for fixed pattern
- Labeling of layout patterns
  - C2C threshold
- Study of pattern radius
  - Tradeoffs between context information and imbalanced data distribution
- Efficient learning model
  - ResNet+NoPooling