Hotspot Detection via Attention-based Deep Layout Metric Learning

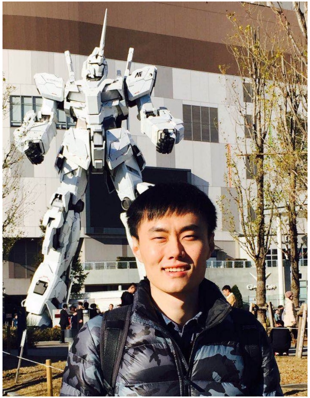
Hao Geng[1], Haoyu Yang[1], Lu Zhang[1], Jin Miao[2], Fan Yang[3], Xuan Zeng[3], Bei Yu[1]

[1]The Chinese University of Hong Kong
[2]Synopsys
[3]Fudan University

Biography

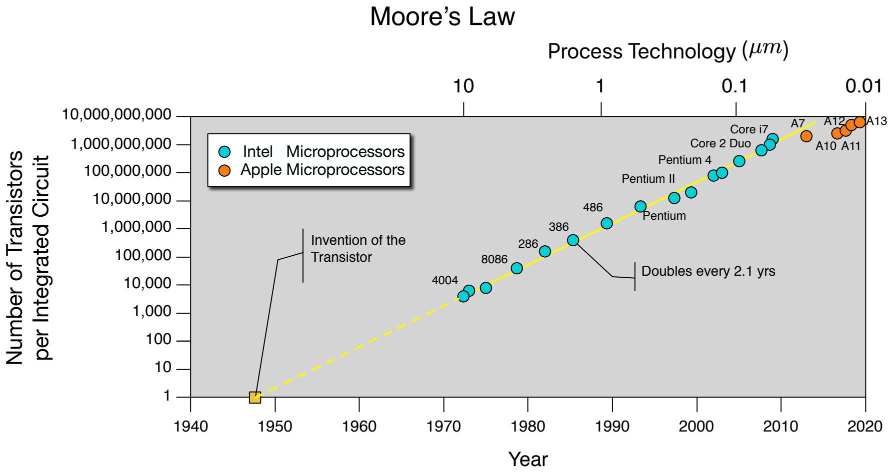
Hao GENG

Ph.D. candidate
The Chinese University of Hong Kong (CUHK)
hgeng@cse.cuhk.edu.hk

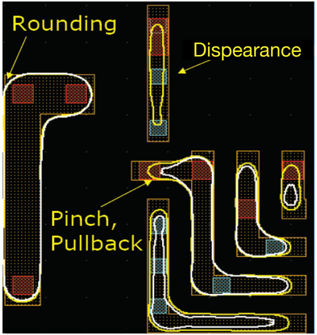


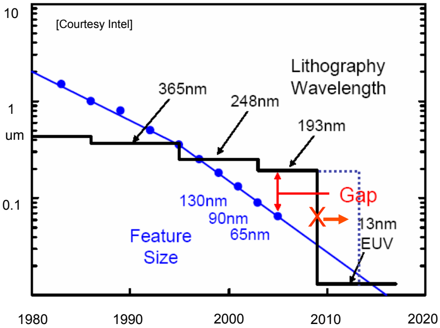
Hao is a year-4 Ph.D. student at the Department of Computer Science and Engineering, under the supervision of Prof. Bei YU since Fall 2017. His research interests include optimization, machine learning and deep learning techniques in VLSI CAD. He has received one best paper award nomination from ASPDAC 2019. Recently he has worked on design space exploration in VLSI CAD.

Moore's Law to Extreme Scaling

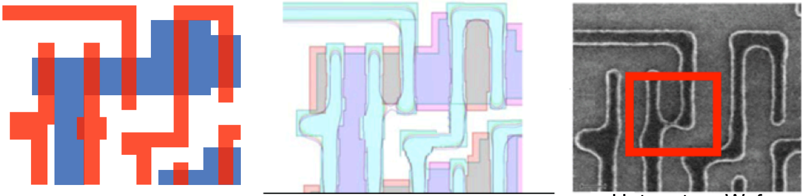


Manufacturability Status & Challenges



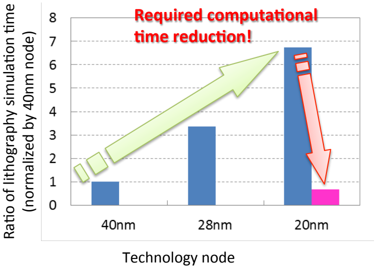


Challenge: Failure (Hotspot) Detection

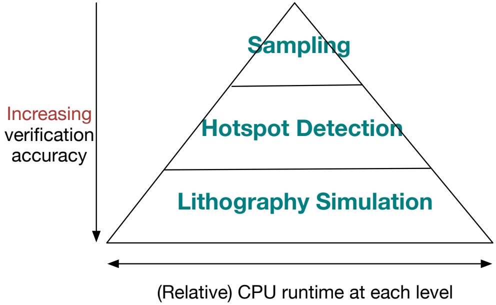


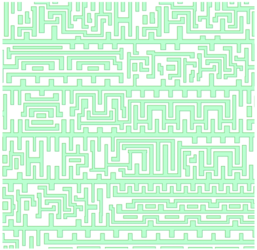
Pre-OPC Layout | Post-OPC Mask | Hotspot on Wafer

- RET: OPC, SRAF, MPL
- Still hotspot: low fidelity patterns
- Simulations: extremely CPU intensive



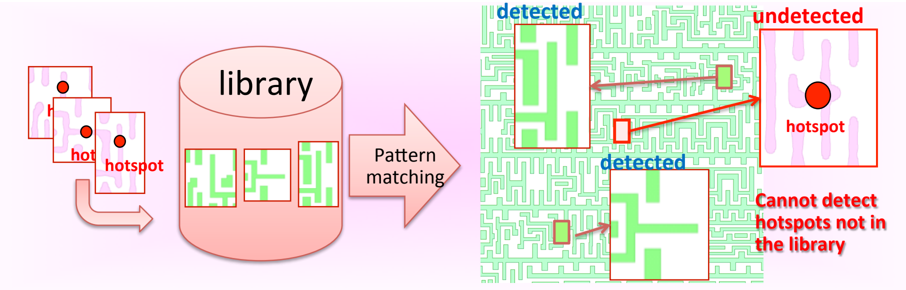
Hotspot Detection Hierarchy





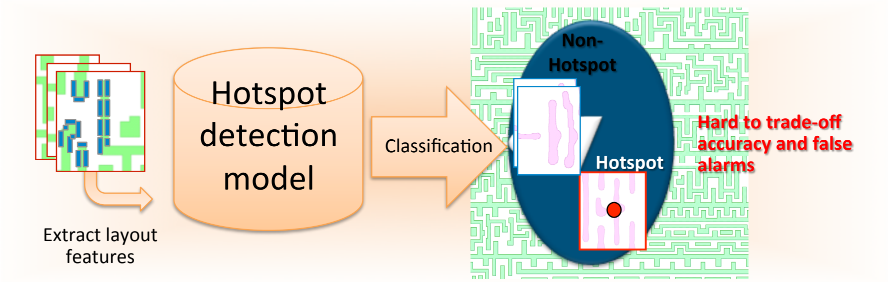
- **Sampling** (DRC Checking):
  scan and rule check each region
- **Hotspot Detection**:
  verify the sampled regions and report potential hotspots
- **Lithography Simulation**:
  final verification on the reported hotspots

Pattern Matching based Hotspot Detection



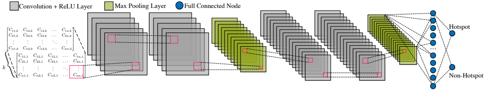
- Fast and accurate
- [Yao+,ICCAD'06], [Lin+,DAC'2013], [Wen+,TCAD'14], [Tseng+,SPIE'19]
- Hard to detect non-seen pattern

Conventional Machine Learning based Hotspot Detection

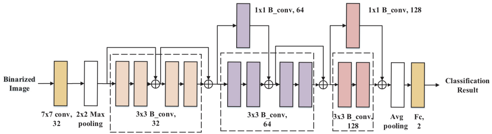


- Train a model to predict new patterns
- Decision-tree, ANN, SVM, Boosting ...
- [Ding+,TCAD'11],[Yu+,JM3'15],[Zhang+,ICCAD'16], [Ye+,DATE'19]
- Hard to balance accuracy and false-alarm

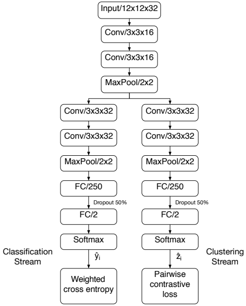
Deep Learning based Hotspot Detection



TCAD'19



DAC'19

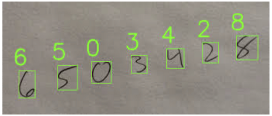


ASPDAC'19

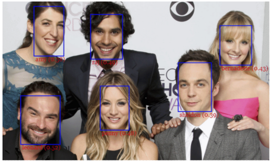
- Success has been achieved: [Yang+,TCAD'19], [Chen+,ASPDAC'19], [Jiang+,DAC'19]
- Two-stage flow and a lack of guidances from supervised information
- Question: Can we learn more informative and discriminative features?

Metric Learning in Daily Life

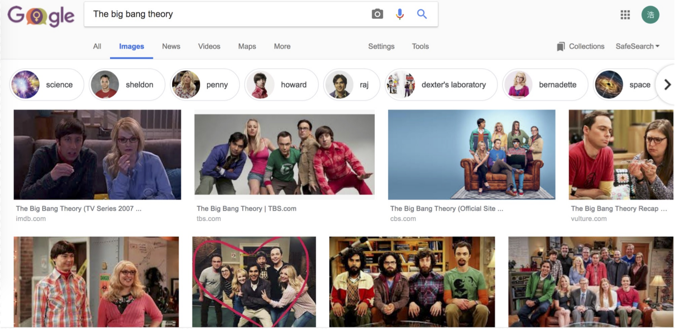
- Several applications of metric learning for similarity measures



Recognizing handwriting



Automatic detection of faces



Search Engines to match a query (could be text, image and etc.)

Basic Concepts of Metric Learning

Definition

Learning a metric that quantify the similarity or the "distance" between every pair of samples in a dataset.

- Pre-defined Metrics: Metrics which are fully specified without the knowledge of data. For example, Euclidian Distance.

$$D(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 - \mathbf{x}_2)^\top (\mathbf{x}_1 - \mathbf{x}_2)$$

- Metrics which can only be defined with the knowledge of the data. For example, Mahalanobis Distance

$$D(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 - \mathbf{x}_2)^\top \mathbf{M} (\mathbf{x}_1 - \mathbf{x}_2)$$

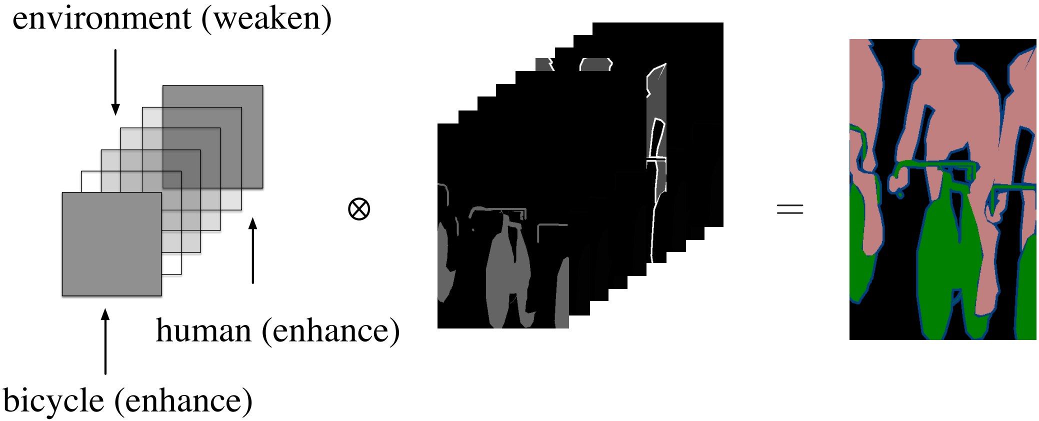
Deep Metric Learning

Deep Metric Learning

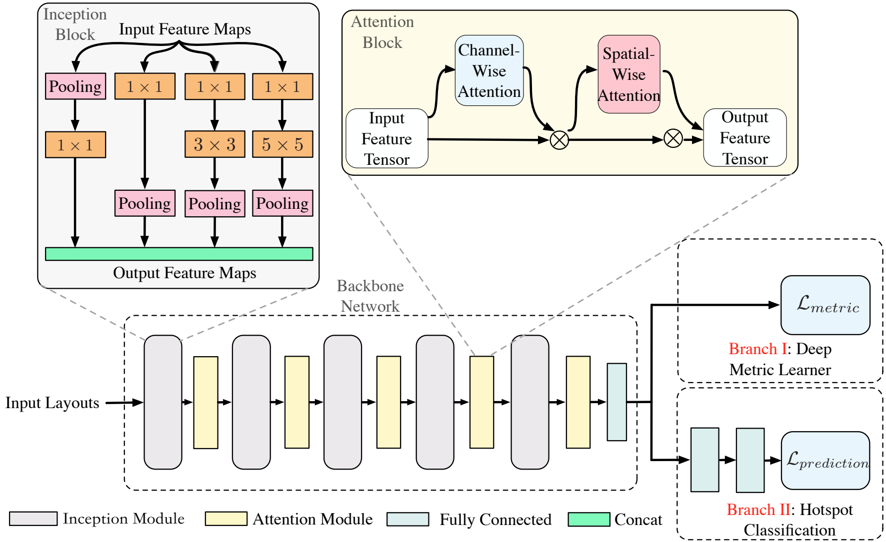
Deep metric learning **maps** an image into a feature vector in a manifold space via **deep neural networks**. In this manifold space, the **Euclidean** distance (or the **cosine** distance) can be directly used as the distance metric between two points.

- Measuring the **similarity** among different layout designs is extremely crucial
- **Supervised** Feature Extractor: learn feature embedding with label information to well represent layouts
- **Less** manually design

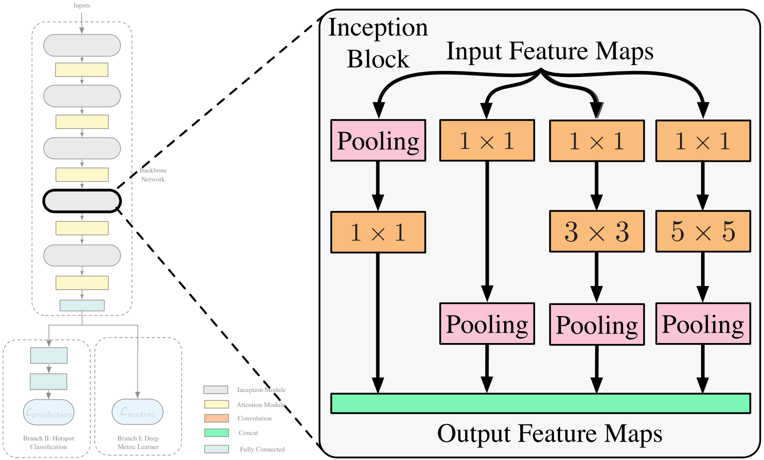
Attention Mechanism: An Example in Computer Vision



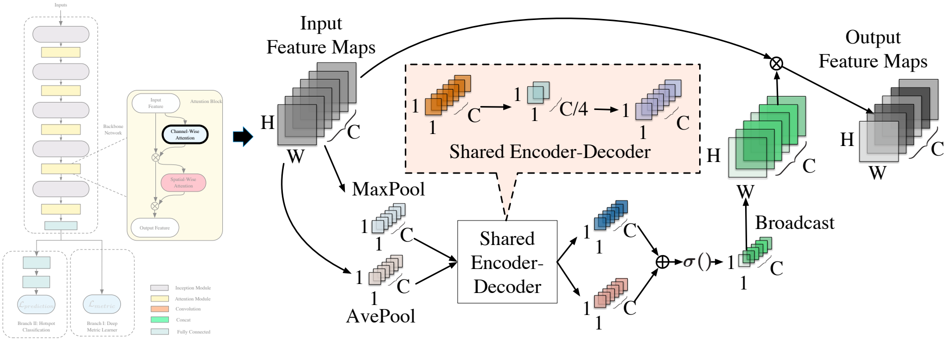
Our Solution: Overview



Our Solution: Inception Module



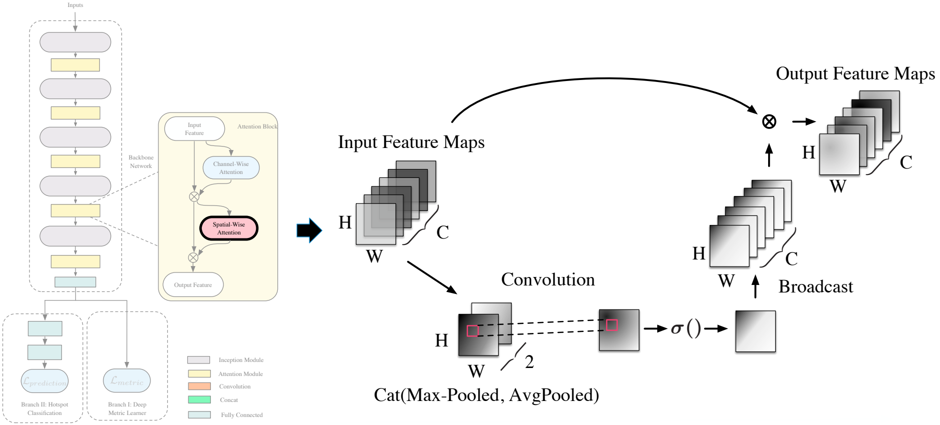
Our Solution: Channel-wise Attention Module



$$\vec{T}' = A_c(\vec{T}) \otimes \vec{T}, \tag{1}$$

$$A_c(\vec{T}) = \sigma(\mathrm{ED}(\mathrm{AvgPool}(\vec{T})) + \mathrm{ED}(\mathrm{MaxPool}(\vec{T}))). \tag{2}$$

Our Solution: Spatial-wise Attention Module



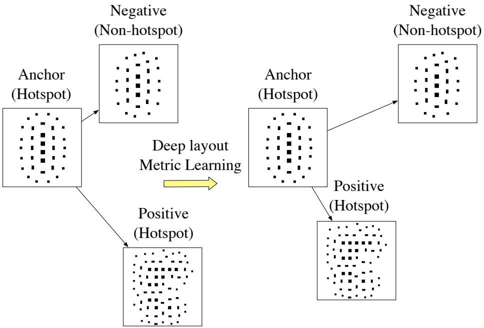
$$\vec{T}'' = A_s(\vec{T}') \otimes \vec{T}', \tag{3}$$

$$A_s(\vec{T}') = \sigma(\mathrm{Conv}(\mathrm{Cat}(\mathrm{AvgPool}(\vec{T}'), \mathrm{MaxPool}(\vec{T}')))). \tag{4}$$

Our Solution: Metric Learning Loss in Branch I (I)

Motivation

- In original space, the anchor is much similar to the negative
- After deep layout metric learning, in a new manifold, the two hotspot layout clips are kept apart from the non-hotspot clip

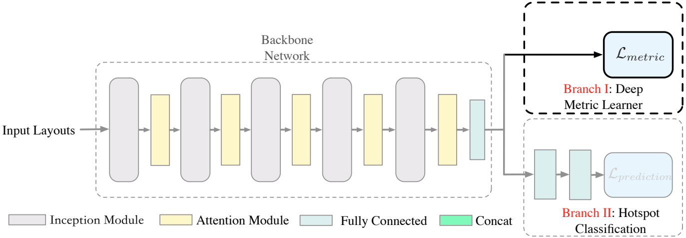


Our Solution: Metric Learning Loss in Branch I (II)

- A triplet: $f_{\vec{w}}(\vec{x}_i), f_{\vec{w}}(\vec{x}_i^+), f_{\vec{w}}(\vec{x}_i^-)$
- $f_{\vec{w}}(\vec{x}_i)$: an anchor layout clip
- $f_{\vec{w}}(\vec{x}_i^+)$: sharing the same label with the anchor
- $f_{\vec{w}}(\vec{x}_i^-)$: having the opposite label to the anchor

$$\min_{\vec{w}} \frac{1}{n}\sum_{i=1}^{N} \max(0, M + ||f_{\vec{w}}(\vec{x}_i) - f_{\vec{w}}(\vec{x}_i^+)||_2^2 - ||f_{\vec{w}}(\vec{x}_i) - f_{\vec{w}}(\vec{x}_i^-)||_2^2) \tag{5a}$$

$$\text{s.t. } ||f_{\vec{w}}(\vec{x}_i)||_2^2 = 1,\ \forall (f_{\vec{w}}(\vec{x}_i), f_{\vec{w}}(\vec{x}_i^+), f_{\vec{w}}(\vec{x}_i^-)) \in \mathcal{T}. \tag{5b}$$



Our Solution: Metric Learning Loss in Branch I (III)

Gradients Calculation:

$$\frac{\partial \mathcal{L}_{metric}(f_{\vec{w}}(\vec{x}_i), f_{\vec{w}}(\vec{x}_i^{+}), f_{\vec{w}}(\vec{x}_i^{-}))}{\partial f_{\vec{w}}(\vec{x}_i^{+})} = \frac{2}{n}\left(f_{\vec{w}}(\vec{x}_i^{+}) - f_{\vec{w}}(\vec{x}_i)\right) \cdot \mathbf{1}\left(\mathcal{L}_{metric}(f_{\vec{w}}(\vec{x}_i), f_{\vec{w}}(\vec{x}_i^{+}), f_{\vec{w}}(\vec{x}_i^{-})) > 0\right), \tag{6a}$$

$$\frac{\partial \mathcal{L}_{metric}(f_{\vec{w}}(\vec{x}_i), f_{\vec{w}}(\vec{x}_i^{+}), f_{\vec{w}}(\vec{x}_i^{-}))}{\partial f_{\vec{w}}(\vec{x}_i^{-})} = \frac{2}{n}\left(f_{\vec{w}}(\vec{x}_i) - f_{\vec{w}}(\vec{x}_i^{-})\right) \cdot \mathbf{1}\left(\mathcal{L}_{metric}(f_{\vec{w}}(\vec{x}_i), f_{\vec{w}}(\vec{x}_i^{+}), f_{\vec{w}}(\vec{x}_i^{-})) > 0\right), \tag{6b}$$

$$\frac{\partial \mathcal{L}_{metric}(f_{\vec{w}}(\vec{x}_i), f_{\vec{w}}(\vec{x}_i^{+}), f_{\vec{w}}(\vec{x}_i^{-}))}{\partial f_{\vec{w}}(\vec{x}_i)} = \frac{2}{n}\left(f_{\vec{w}}(\vec{x}_i^{-}) - f_{\vec{w}}(\vec{x}_i^{+})\right) \cdot \mathbf{1}\left(\mathcal{L}_{metric}(f_{\vec{w}}(\vec{x}_i), f_{\vec{w}}(\vec{x}_i^{+}), f_{\vec{w}}(\vec{x}_i^{-})) > 0\right), \tag{6c}$$

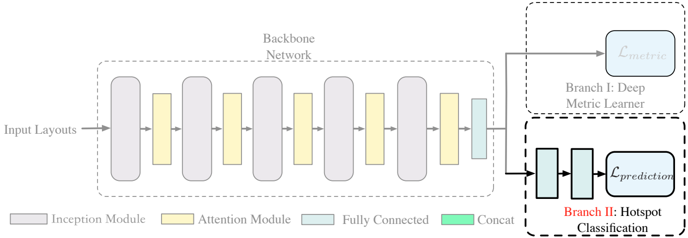
where $\mathbf{1}$ is the indicator function which is defined as:

$$\mathbf{1}(x) = \begin{cases} 1 & \text{if x is true,} \\ 0 & \text{otherwise.} \end{cases} \tag{6d}$$

Our Solution: Classification Loss in Branch II

- y^* is the predicted probability of a layout clip
- y is its ground truth (binary indicator).

$$-(y \log(y^*) + (1 - y) \log(1 - y^*)), \tag{7}$$

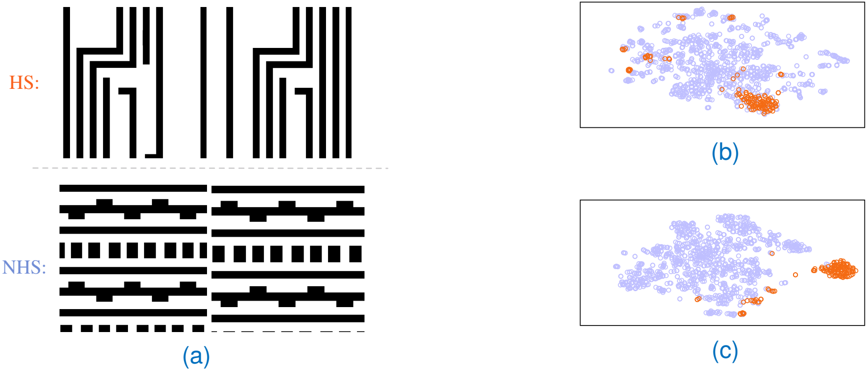


Experimental Setup

- A platform with a Xeon Silver 4114 CPU processor and Nvidia TITAN Xp Graphic card
- ICCAD 2012 contest benchmark
- More challenging via layer benchmark suite (under 45nm)
- Benchmark Statistics:

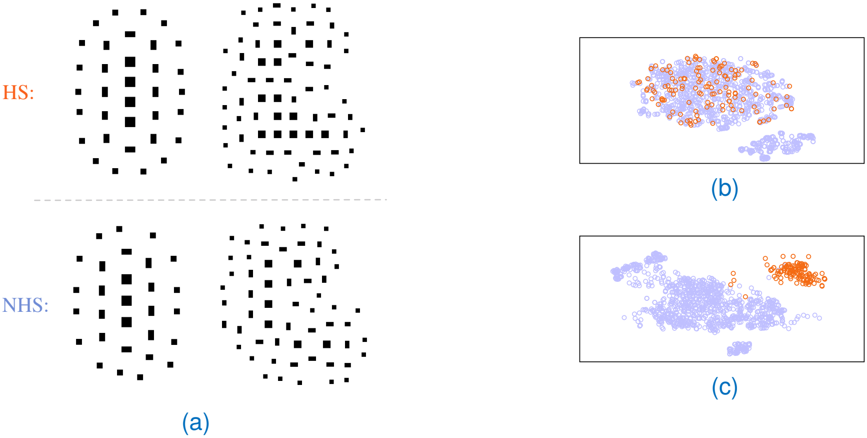
Benchmarks	Training Set		Testing Set		Size/Clip (μm^2)
	HS#	NHS#	HS#	NHS#	
ICCAD12	1204	17096	2524	13503	3.6×3.6
Via-1	3418	10302	2267	6878	2.0×2.0
Via-2	1029	11319	724	7489	2.0×2.0
Via-3	614	19034	432	12614	2.0×2.0
Via-4	39	23010	26	15313	2.0×2.0
Via-Merge	5100	63665	3449	42294	2.0×2.0

The t-SNE visualizations of feature embeddings on ICCAD12 benchmarks



(a) The exemplars of hotspots and non-hotspots; (b) The DCT feature embeddings of TCAD’19; (c) The feature embeddings of our proposed framework.

The t-SNE visualizations of feature embeddings on VIA benchmarks



(a) The exemplars of hotspots and non-hotspots; (b) The DCT feature embeddings of TCAD’19; (c) The feature embeddings of our proposed framework.

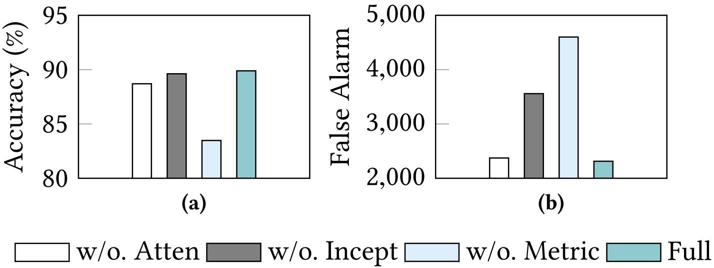
Comparison with state of the arts

- "Accu": Accuracy, the ratio between the number of correctly categorized hotspot clips and the number of real hotspot clips.
- "FA": False Alarm, non-hotspot clips that are classified as hotspots

Bench	TCAD'19			DAC'19			ASPDAC'19			JM3'19			Ours		
	Accu (%)	FA	Time (s)	Accu (%)	FA	Time (s)	Accu (%)	FA	Time (s)	Accu (%)	FA	Time (s)	Accu (%)	FA	Time (s)
ICCAD12	98.40	3535	502.70	98.54	3260	561.28	97.66	2825	441.96	97.82	2651	505.67	98.42	2481	143.79
Via-1	71.50	773	43.36	89.85	1886	57.76	64.18	1077	52.95	89.19	2624	47.87	93.42	1589	19.83
Via-2	65.06	1290	40.02	73.00	1222	21.66	30.52	372	43.21	38.81	454	43.06	86.32	1100	13.22
Via-3	48.15	760	60.23	73.38	3406	43.15	26.92	148	77.09	21.06	42	67.13	88.20	2105	20.69
Via-4	76.92	155	67.44	73.08	15288	51.98	61.54	74	87.24	46.15	21	77.15	80.77	152	20.70
Via-Merge	88.01	7633	165.85	90.42	9295	105.30	72.77	3859	228.57	84.34	6759	170.91	92.20	6453	59.74
Average	74.67	2357.67	146.60	83.06	5726.17	140.19	58.93	1392.50	155.17	62.90	2091.83	151.97	89.89	2313.33	46.33
Ratio	0.83	1.02	3.16	0.92	2.48	3.03	0.66	**0.60**	3.35	0.70	0.90	3.28	**1.00**	1.00	**1.00**

The ablation study

- 'w/o. Atten": the detector without attention modules
- "w/o. Incept": the detector with inception blocks replaced with vanilla convolution layers
- "w/o. Metric": the detector trained without layout metric learning loss
- "Full": the proposed detector



Conclusion

- A new end-to-end hotspot detection flow
- Two-branch design
- Works well on previous and more challenging benchmarks

Thank You