Attacking a CNN-based Layout Hotspot Detector Using Group Gradient Method

Haoyu Yang[1], Shifan Zhang[1], Kang Liu[2], Siting Liu[1], Benjamin Tan[2], Ramesh Karri[2], Siddharth Garg[2], Bei Yu[1], Evangeline F.Y. Young[1]

[1]The Chinese University of Hong Kong
[2]New York University

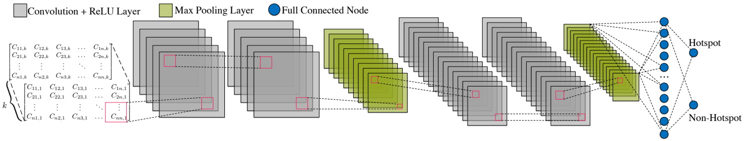
About The Speaker

Haoyu Yang
Postdoctoral Fellow
Department of Computer Science
Chinese University of Hong Kong
https://phdyang007.github.io/

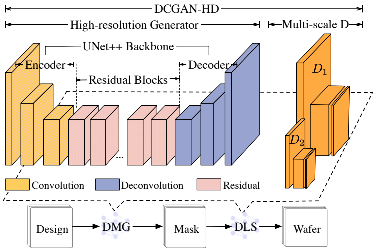
He received the Ph.D degree from the department of Computer Science and Engineering, Chinese University of Hong Kong. His research interests include (1) Machine Learning in VLSI Design for Manufacturability (2) High Performance VLSI Physical Design with Parallel Computing and (3) Machine Learning Security. He received the best paper candidate and best poster presentation from ASPDAC 2019. He is also the winner of Nick Cobb Scholarship 2019 by SPIE and Mentor Graphics.

Deep Learning Enables Intelligent DFM

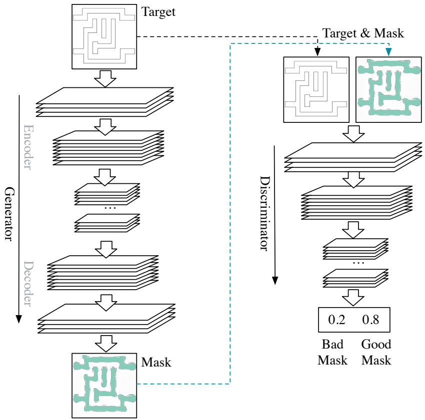
Lithography Hotspot Detection [Yang+,TCAD'19] [Jiang+,DAC'19] [Geng+,ICCAD'20]



Lithography Modeling [Ye+,DAC'19] [Ye+,ISPD'20] [Chen+,ICCAD'20]

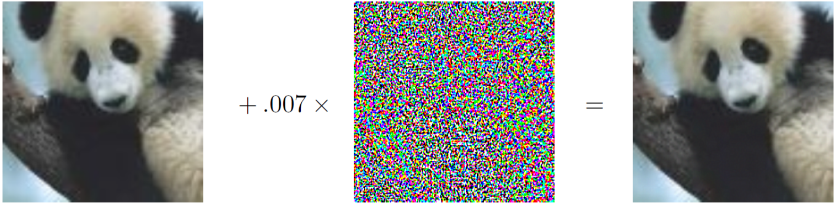


Mask Optimization [Yang+,TCAD'20] [Chen+,ICCAD'20]



Deep Neural Networks Are Fragile

Deep Neural Networks Are Vulnerable to Adversarial Examples [Goodfellow+,ICLR'15]$*$



$\boldsymbol{x}$
"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence

$\boldsymbol{x} + \epsilon \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

$*$Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples", in ICLR, 2015

Rethinking Deep Learning-based Hotspot Detection

Are DLHSDs Apparently Secure?

- Layouts are consistent with design rules and schematic designs.
- Adversarial examples are generated by pixel-wise manipulation on original image.
- DLHSDs are invulnerable to adversarial examples (generated by SOTA).

The Answer Is No. [Liu+,TODAES'20] †

- Neural networks see limited training data.
- DRC-clean and functionality-preserving manipulation on layouts are feasible.

Why Look for Adversarial Layouts?

- Designs of Interest
- Robust ML Design

†Liu, Kang, et al. "Adversarial Perturbation Attacks on ML-based CAD: A Case Study on CNN-based Lithographic Hotspot Detection." ACM Transactions on Design Automation of Electronic Systems (TODAES) 25.5 (2020): 1-31.

Preliminaries

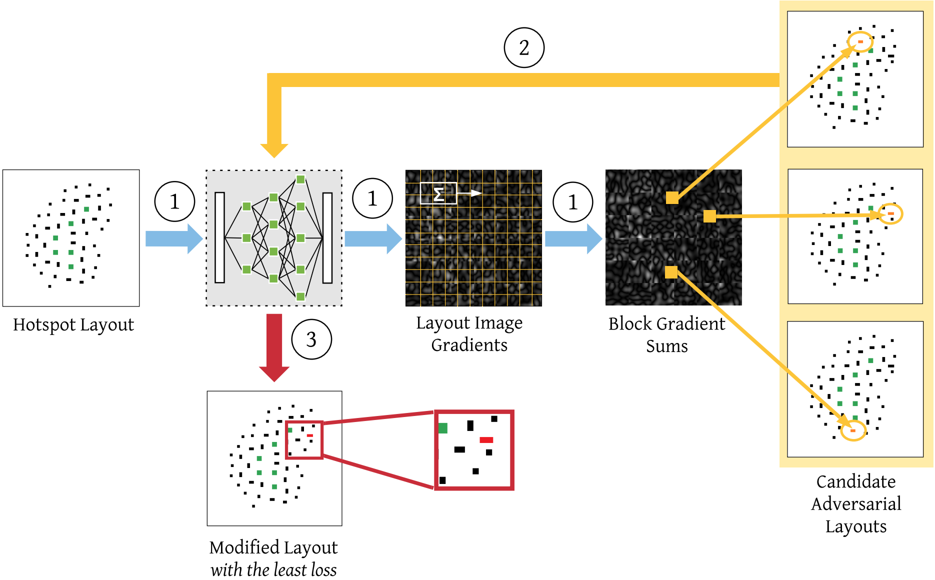
Terminologies

- $\boldsymbol{X}$: Input layout image
- $f(\cdot; \boldsymbol{W})$: Trained neural networks parameterized by $\boldsymbol{W}$
- $y^* \in \{0, 1\}$: Label of $\boldsymbol{X}$
- $y = f(\boldsymbol{X})$: Predicted logit of $\boldsymbol{X}$
- $\boldsymbol{X}' = \boldsymbol{X} + \boldsymbol{R}$: Adversarial layout image by including perturbations $\boldsymbol{R}$ on $\boldsymbol{X}$

Objective

- Given $\boldsymbol{X}$ satisfying $y^* = 1$ and $f(\boldsymbol{X}) > 0$, we want to find $\boldsymbol{R}$ such that $\boldsymbol{X}'$ is DRC-clean and as close to $\boldsymbol{X}$ as possible and in the mean time, $f(\boldsymbol{X}') < 0$.

Generating Adversarial Layouts [Liu+,TODAES'20]



Generating Adversarial Layouts [Liu+,TODAES'20]

Iterative Run Till the Label Flipped.

1. Feedforward to acquire the gradient of loss w.r.t. input.
2. Locate regions with largest gradient response.
3. Place perturbation.

The Procedure

$$
\begin{aligned}
\min \quad & ||\boldsymbol{R}||_F^2, \\
\text{s.t.} \quad & f(\boldsymbol{X}+\boldsymbol{R};\boldsymbol{W}) < 0, \\
& f(\boldsymbol{X};\boldsymbol{W}) > 0.
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{R} &= -\gamma \frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{X}}, \\
\boldsymbol{X} &= \boldsymbol{X} + \boldsymbol{R}.
\end{aligned}
$$

$$
i = \arg\max_k \sum_{(x,y)\in\mathcal{R}_k} \frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{X}(x,y)}.
$$

Pixel-based Gradient Method Is Not Optimal

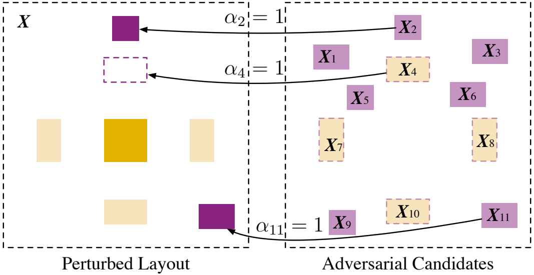
- Some perturbed pixels in the selected grid do not contribute to flip the label.
- Not designed to remove geometry as candidate perturbations.

Group Gradient Method Is Our Proposal

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \mathcal{L}(\boldsymbol{\alpha}) = || \sum_i \alpha_i \boldsymbol{X}_i ||_F^2, \\
\text{s.t.} \quad & f(\boldsymbol{X} + \sum_i \alpha_i \boldsymbol{X}_i; \boldsymbol{W}) < 0, \\
& \alpha_i + \alpha_j \leq 1, \forall i, j \in \mathcal{C}, \\
& \alpha_i \in \{0, 1\}, \forall i.
\end{aligned}
$$

- $\mathcal{X} = \{\boldsymbol{X}_i\}$: A group of perturbation candidates that do not violate design rules with existing geometry and affect design functionality.
- $\alpha_i \in \{0, 1\}$: Coefficients indicate whether $\boldsymbol{X}_i$ is selected.
- $\mathcal{L}$: The change of the layout by inserting perturbations.
- $\mathcal{C}$: Conflict set indicates whether two perturbations can be selected simultaneously.

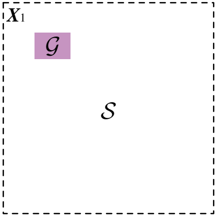
The Group Gradient Method



- Illustration of the proposed attack scheme, with a solution of $\alpha_2 = 1, \alpha_4 = 1$ and $\alpha_{11} = 1$.

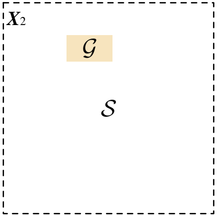
Perturbation Candidate Enumeration

Positive Candidate



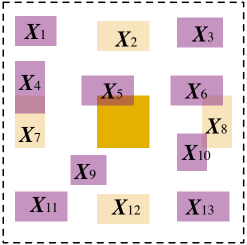
$$\boldsymbol{X}_1(i,j) = \begin{cases} 1, & \text{if } (i,j) \in \mathcal{G}, \\ 0, & \text{if } (i,j) \in \mathcal{S}. \end{cases}$$

Negative Candidate

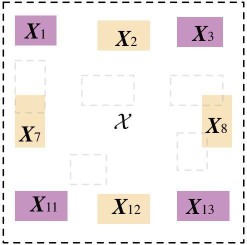


$$\boldsymbol{X}_1(i,j) = \begin{cases} -1, & \text{if } (i,j) \in \mathcal{G}, \\ 0, & \text{if } (i,j) \in \mathcal{S}. \end{cases}$$

Perturbation Candidate Enumeration



(a)



(b)

- (a) Massive perturbation candidates, (b) Legal perturbation candidates.
- Visualization of perturbation candidate generation $\mathcal{X} = \{X_1, X_2, X_3, X_7, X_8, X_{11}, X_{12}, X_{13}\}$. Due to design rule violation with existing shapes $\{X_4, X_5, X_6, X_9, X_{10}\}$ will not be included in the perturbation candidate set $\mathcal{X}$.

Numerical Optimization

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \mathcal{L}(\boldsymbol{\alpha}) = ||\sum_i \alpha_i \boldsymbol{X}_i||_F^2, \\
\text{s.t.} \quad & f(\boldsymbol{X} + \sum_i \alpha_i \boldsymbol{X}_i; \boldsymbol{W}) < 0, \\
& \alpha_i + \alpha_j \leq 1, \forall i, j \in \mathcal{C}, \\
& \alpha_i \in \{0, 1\}, \forall i.
\end{aligned}
$$

- Nonlinear Integer Programming.
- Non-Convex.
- No closed form solution.

Numerical Optimization

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \mathcal{L}_{\text{cont}}(\boldsymbol{\alpha}) = ||\sum_i \alpha_i \boldsymbol{X}_i||_F^2, \\
\text{s.t.} \quad & f(\boldsymbol{X} + \sum_i \alpha_i \boldsymbol{X}_i; \boldsymbol{W}) < 0, \\
& 0 \leq \alpha_i \leq 1, \forall i.
\end{aligned}
$$

- The constraint regarding to the conflict set is processed in perturbation candidate enumeration.
- Problem relaxed to continuous.

Numerical Optimization

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \mathcal{L}_{\textsf{sim}}(\boldsymbol{\alpha}) = ||\boldsymbol{\alpha}||_2^2, \\
\text{s.t.} \quad & f(\boldsymbol{X} + \sum_i \alpha_i \boldsymbol{X}_i; \boldsymbol{W}) < 0, \\
& 0 \leq \alpha_i \leq 1, \forall i.
\end{aligned}
$$

- Objective approximation.
- Reduce computation significantly.

Numerical Optimization

Theorem

Let $\boldsymbol{\alpha}^*_{\textit{cont}}$ and $\boldsymbol{\alpha}^*_{\textit{sim}}$ be the optimal solution of $\mathcal{L}_{\textit{cont}}$ and $\mathcal{L}_{\textit{sim}}$, respectively, then we have,

$$\mathcal{L}_{\textit{cont}}(\boldsymbol{\alpha}^*_{\textit{cont}}) \leq \mathcal{L}_{\textit{cont}}(\boldsymbol{\alpha}^*_{\textit{sim}}),$$

and,

$$\begin{aligned} &\mathcal{L}_{\textit{cont}}(\boldsymbol{\alpha}^*_{\textit{sim}}) - \mathcal{L}_{\textit{cont}}(\boldsymbol{\alpha}^*_{\textit{cont}}) \\ \leq\ & ||\boldsymbol{\alpha}^*_{\textit{sim}}||_0^2 \cdot ||\boldsymbol{X}_\delta||_F^2 - ||\boldsymbol{\alpha}^*_{\textit{cont}}||_0^2 \cdot ||\boldsymbol{X}_\xi||_F^2, \end{aligned}$$

where $\delta = \mathrm{argmax}_i\ |\boldsymbol{e}^\top \boldsymbol{X}_i \boldsymbol{e}|$ and $\xi = \mathrm{argmin}_i\ |\boldsymbol{e}^\top \boldsymbol{X}_i \boldsymbol{e}|$.

Numerical Optimization

$$\min_{\boldsymbol{\alpha}} \quad \mathcal{L}_{\text{lag}}(\boldsymbol{\alpha}, \lambda) = ||\boldsymbol{\alpha}||_2^2 + \lambda f(\boldsymbol{X} + \sum_i \alpha_i \boldsymbol{X}_i; \boldsymbol{W}),$$
$$\text{s.t.} \quad \lambda \geq 0, 0 \leq \alpha_i \leq 1, \forall i,$$

- Problem simplification with Lagrangian relaxation.

$$\alpha_i = \frac{1}{1 + e^{-\beta_i}}, \beta_i \in \mathbb{R}, \forall i.$$

- Auxiliary variables introduced to keep α_i fall into $[0, 1]$ during optimization.

Numerical Optimization

$$\min_{\boldsymbol{\alpha}} \quad \mathcal{L}_{\text{lag}}(\boldsymbol{\alpha}, \lambda) = ||\boldsymbol{\alpha}||_2^2 + \lambda f(\boldsymbol{X} + \sum_i \alpha_i \boldsymbol{X}_i; \boldsymbol{W}),$$
$$\text{s.t.} \quad \lambda \geq 0, 0 \leq \alpha_i \leq 1, \forall i,$$

- Problem simplification with Lagrangian relaxation.

$$\alpha_i = \frac{1}{1 + e^{-\beta_i}}, \beta_i \in \mathbb{R}, \forall i.$$

- Auxiliary variables introduced to keep α_i fall into $[0, 1]$ during optimization.

Numerical Optimization

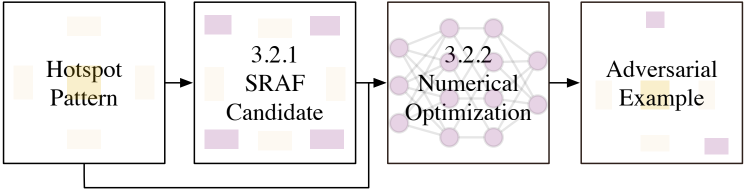
Update β

$$\begin{aligned}\beta_i^{(t+1)} &= \beta_i^{(t)} - \frac{\partial \mathcal{L}_{\text{lag}}^{(t)}}{\partial \alpha_i^{(t)}} \frac{\partial \alpha_i^{(t)}}{\partial \beta_i^{(t)}} \\ &= \beta_i^{(t)} - (2\alpha_i^{(t)} + \lambda \frac{\partial f}{\partial \alpha_i^{(t)}}) \alpha_i^{(t)} (1 - \alpha_i^{(t)}), \forall i.\end{aligned} \tag{8}$$

Update λ

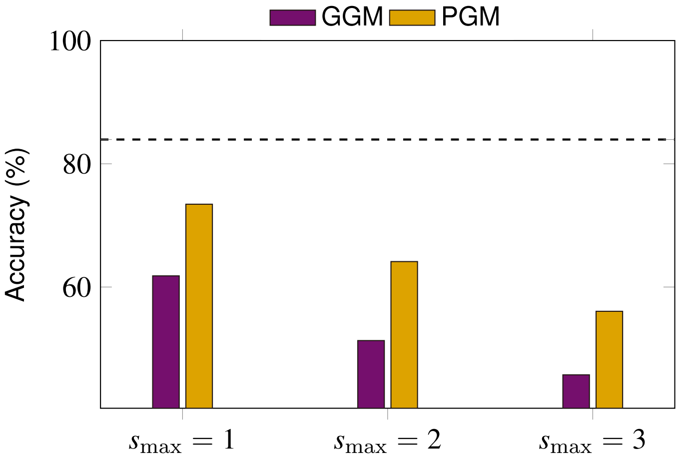
$$\lambda^{(t+1)} = \lambda^{(t)} - f(\boldsymbol{X} + \sum_i \alpha_i^{(t)} \boldsymbol{X}_i; \boldsymbol{W}).$$

Overall Flow

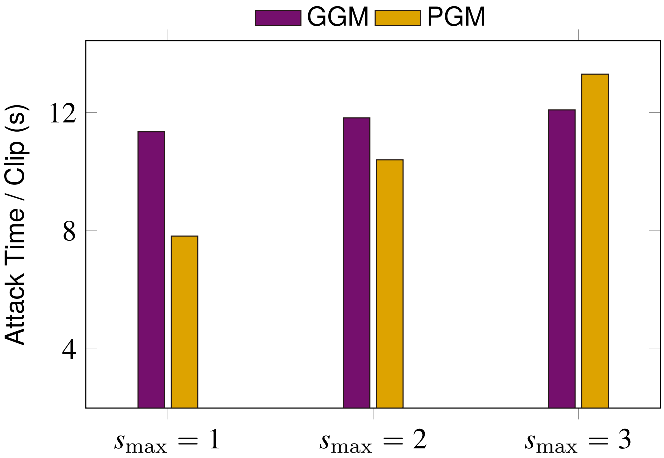


- Candidate perturbations are generated by scanning over the entire clip ensuring a comprehensive solution space.
- GGM optimizes toward DRC-clean perturbation circumventing post-processing and potential deviation from optimality.
- Gradient back-propagation and perturbation candidate determination steps make the framework robust when more changes are used to create adversarial layout examples.

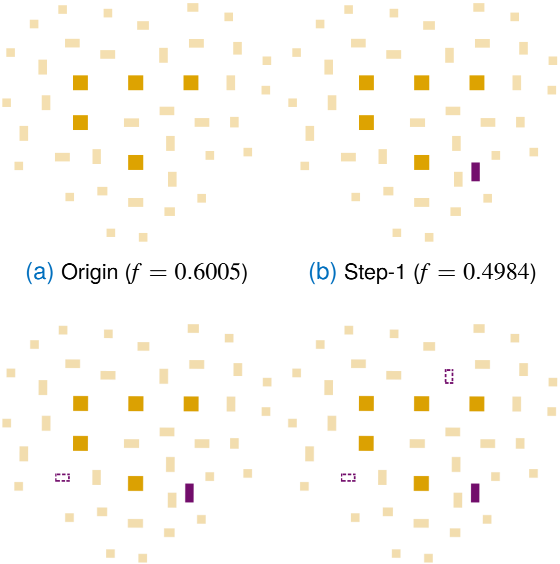
Comparison with State-of-the-Art



Comparison with State-of-the-Art



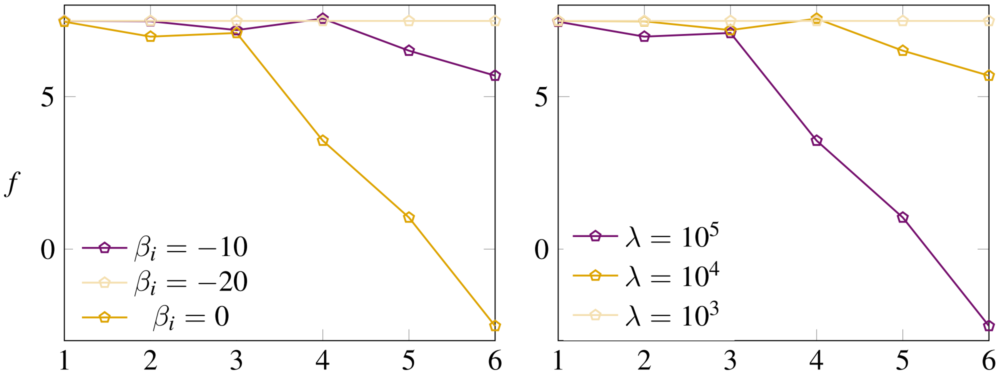
Adversarial Attack Visualization



(a) Origin ($f = 0.6005$) (b) Step-1 ($f = 0.4984$)

(c) Step-2 ($f = 0.0835$) (d) Step-3 ($f = -1.043$)

On the Importance of Hyper Parameters



(a) $\boldsymbol{\beta}$-init (b) λ-init

Conclusion

- We examine the risks of deep learning-based lithography hotspot detectors assuming a practical adversarial attack scenario, and hence motivate us the generation of adversarial layouts.
- We explain that adversarial example generation employing a conventional pixel-based gradient method deviates from the optimal when making legal perturbations.
- We recommend the group gradient method that makes DRC clean perturbations by solving an unconstrained optimization problem with an objective function that is differentiable.
- We expect this study will spur research in defenses against adversarial layout examples culminating in robust machine learning solutions in VLSI design and sign-off flow.

Thank You