RouteNet: Routability Prediction for Mixed-Size Designs Using Convolutional Neural Network

Zhiyao Xie[1], Yu-Hung Huang[2], Guan-Qi Fang[2], Haoxing Ren[3], Shao-Yun Fang[2], Yiran Chen[1], Jiang Hu[4]

[1]Duke University, Durham, NC, USA
[2]National Taiwan University of Science and Technology, Taipei, TW
[3]Nvidia Corporation, Austin, TX, USA
[4]Texas A&M University, College Station, TX, USA

ICCAD 2018

Background

- Challenges in Routability Prediction:
  - Predict routability at placement stage
    - Predict by ‘fast trial global routing’
    - Not fast enough
  - Predict locations of Design Rule Checking (DRC) hotspots
    - Predict by global routing
    - Not accurate enough

Background

- Our Attempt on Such Challenges:
  - Fast routability forecast for placement
    - In terms of number of Design Rule Violations (#DRV)
    - To identify more routable placements among many candidates
  - Prediction of DRC hotspot locations
    - To proactively modify solutions to prevent design rule violations

Background

- Previous solutions:
  - Many fail to consider macros
  - Some require Global Routing information for #DRV prediction

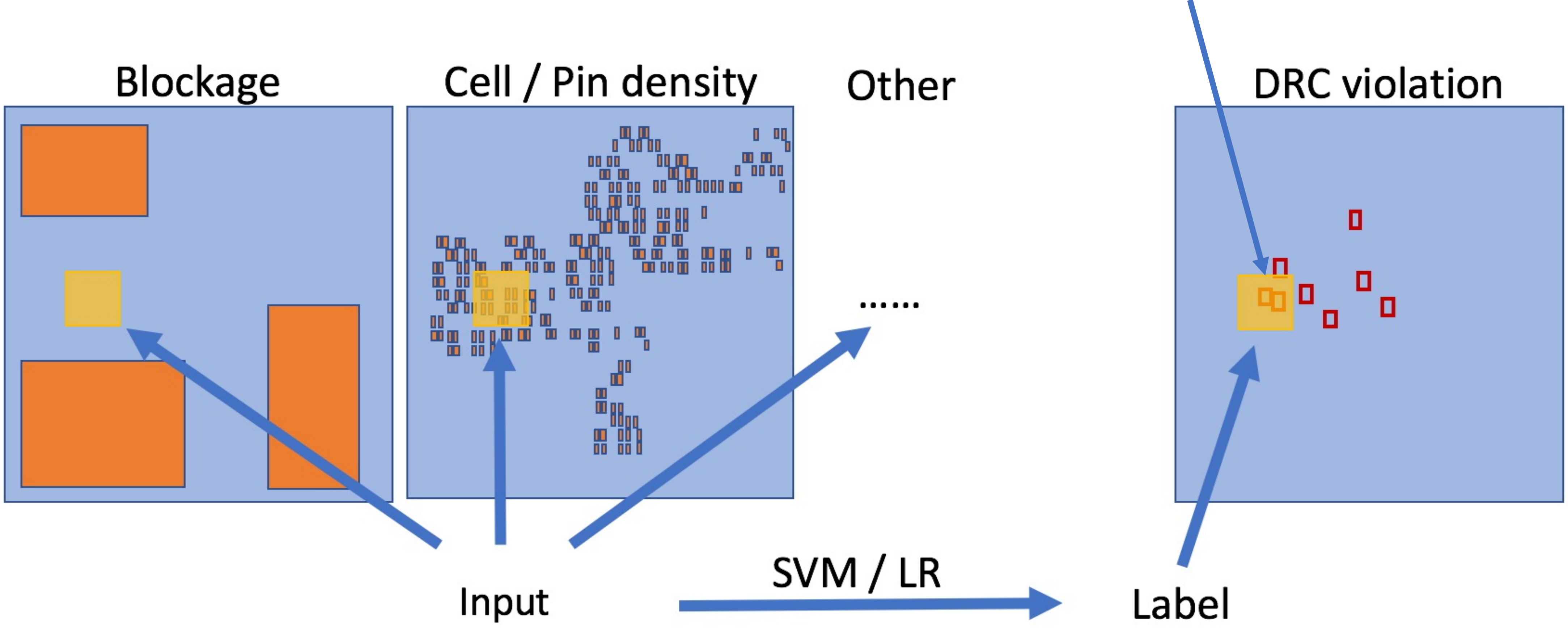
	Methods	Use GR?	Predict #DRV?	Predict hotspot?	Handle macros?
	[18] (Qi, et al., ICCD14)	Y	Y	N	Y
	[26] (Zhou, et al., ASQED15)	Y	Y	N	N
	[3] (Chan, et al., ICCD16)	N	Y	N	N
	[4] (Chan, et al., ISPD17)	Y	N	Y	N
Our Method	RouteNet #DRV prediction	N	Y	N	Y
	RouteNet hotspot prediction	Y	N	Y	Y

GR means global routing

#DRV means number of DRC violations

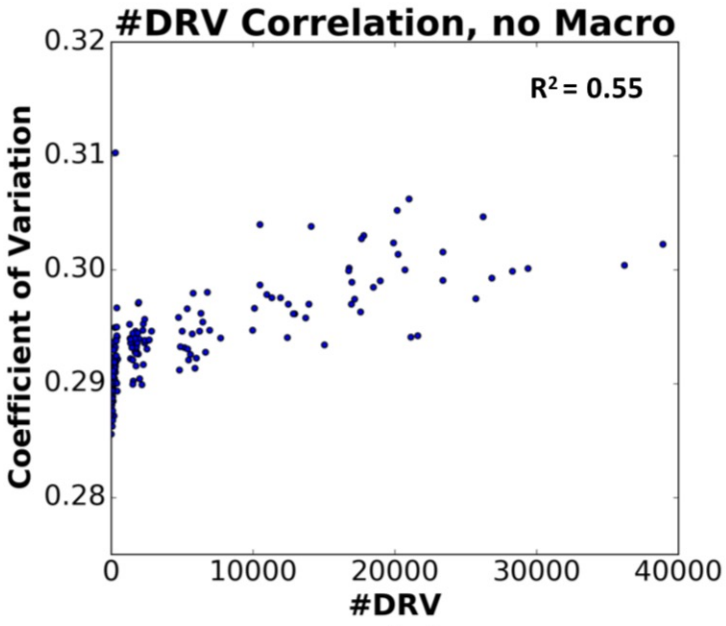
Background

- Previous solutions:
  - Many apply machine learning on every ***small*** *cropped region*

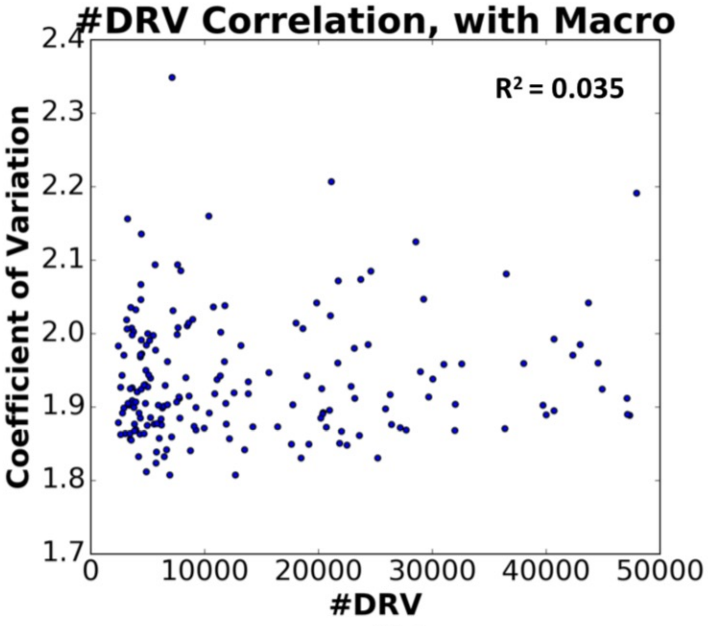


Background

- Challenges of Macros on #DRV prediction:
  - Correlation between pin density and #DRV largely disappears with macro



(a)



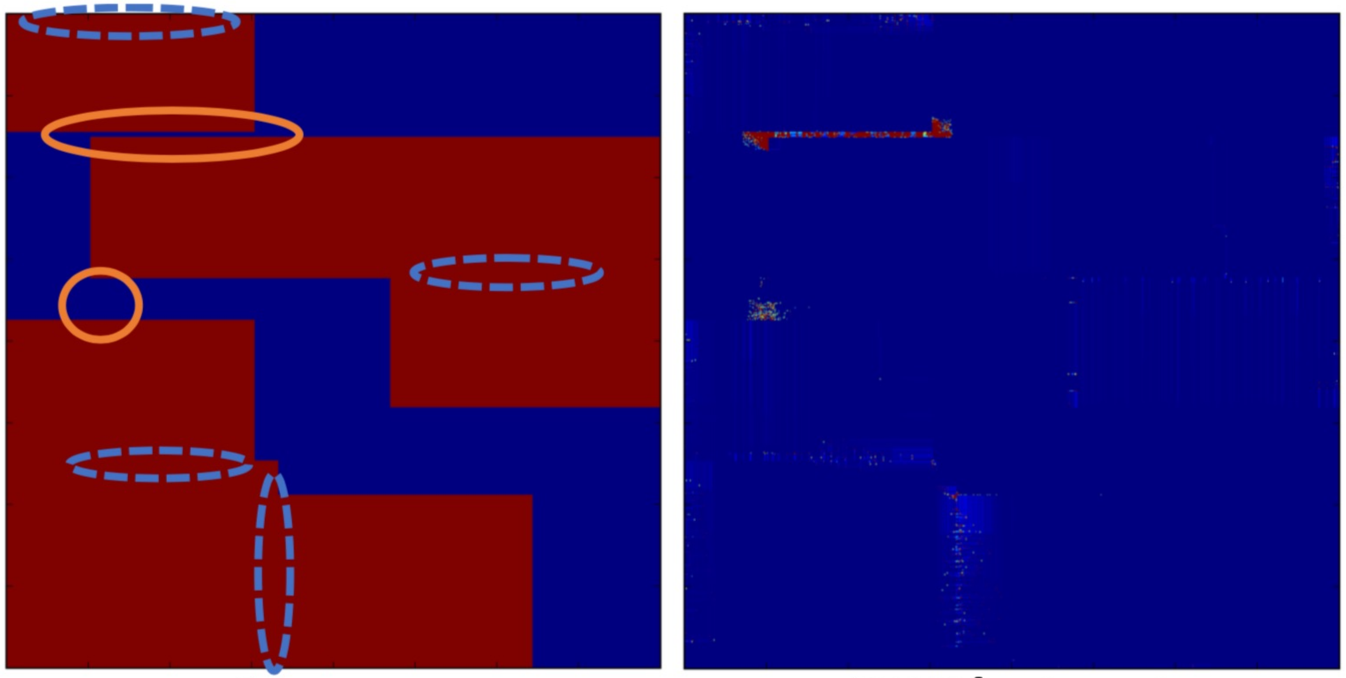
(b)

Correlation between #DRV and coefficient of variation ($\frac{\sigma}{\mu}$) of pin density.

Each point corresponds to one placement.

Background

- Challenges of Macros on hotspot detection:
  - Hotspots tend to aggregate at small gaps between neighboring macros



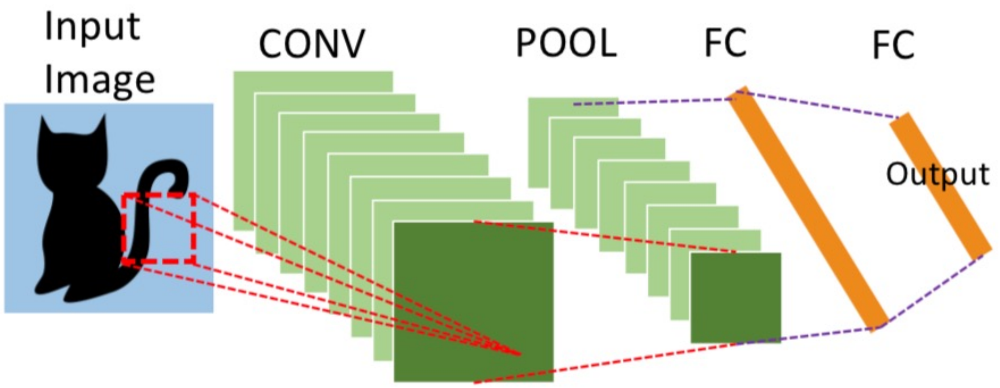
Macros **DRC hotspot**

Background

- Challenges of Macros:
  - A layout with macros is much less **homogeneous**
  - **Homogeneity** implies resemblance among different regions of layout
  - Need a larger region to capture global view
  - **Use deep neural network!**

Background

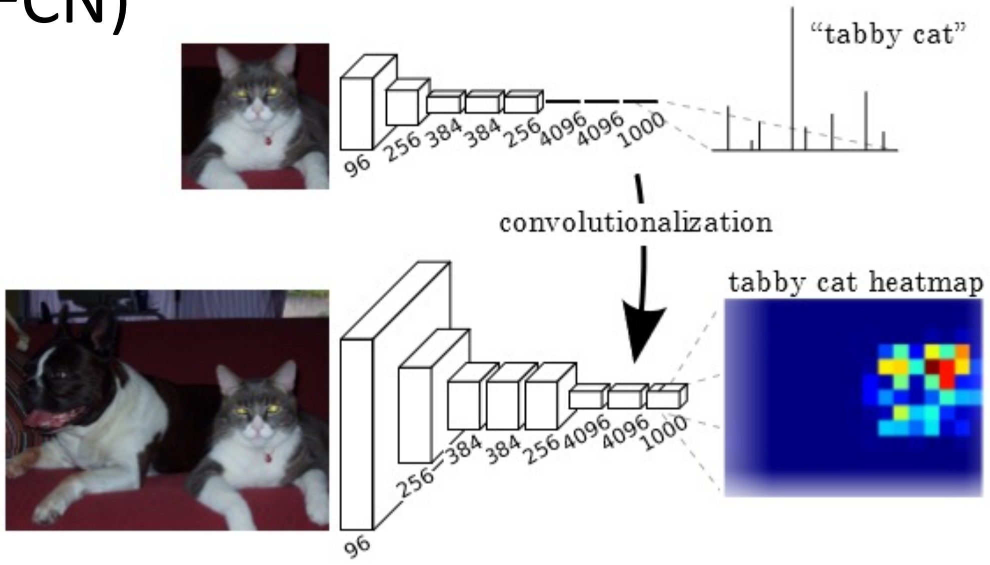
- Convolutional Neural Network (CNN)



Convolutional (CONV), Pooling (POOL) and Fully Connected (FC) layers

Widely used in image classification

- Fully Convolutional Network (FCN)



Eliminate FC layers

May use transposed-convolutional to up-sample

Used in image segmentation, object detection

Image on right side from: Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2017. Fully Convolutional Networks for Semantic Segmentation. (TPAMI)

Features Extraction

- RUDY (Rectangular Uniform wire DensitY) (P. Spinder et al. 2007)
  - RUDY is a pre-routing congestion estimator
  - At (x, y), for $k\ th$ net with bounding box $\{x_{min}^k,\ x_{max}^k,\ y_{min}^k,\ y_{max}^k\}$:

$$w^k = x_{max}^k - x_{min}^k,\ h^k = y_{max}^k - y_{min}^k$$

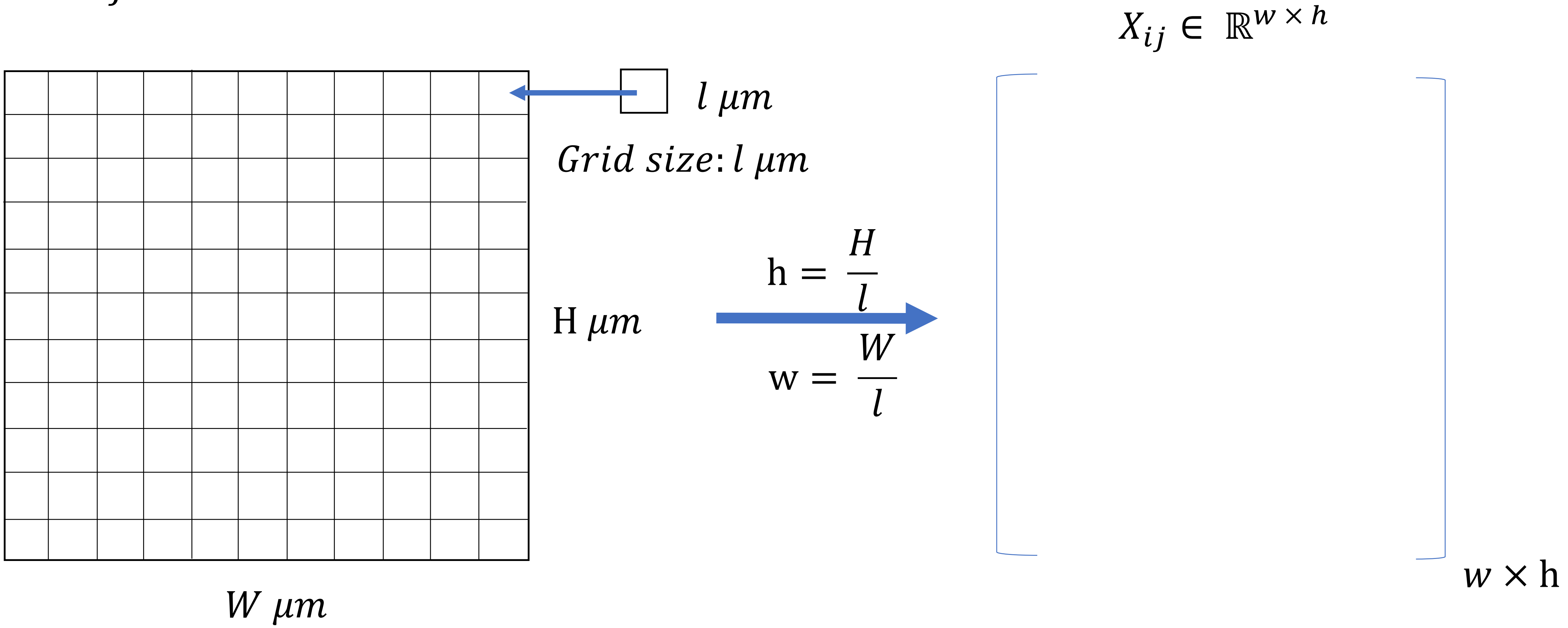
$$c^k = \begin{cases} 1 & x \in [x_{min}^k, x_{max}^k]\ ,\ y \in [y_{min}^k, y_{max}^k] \\ 0 & otherwise \end{cases}$$

$$RUDY^k(x, y) \propto c^k \frac{w^k + h^k}{w^k \times h^k} \longleftarrow \frac{Half\ perimeter}{area}$$

$$RUDY(x, y) = \sum_{k=1}^{K} RUDY^k(x, y) \longleftarrow Sum\ up\ all\ nets$$

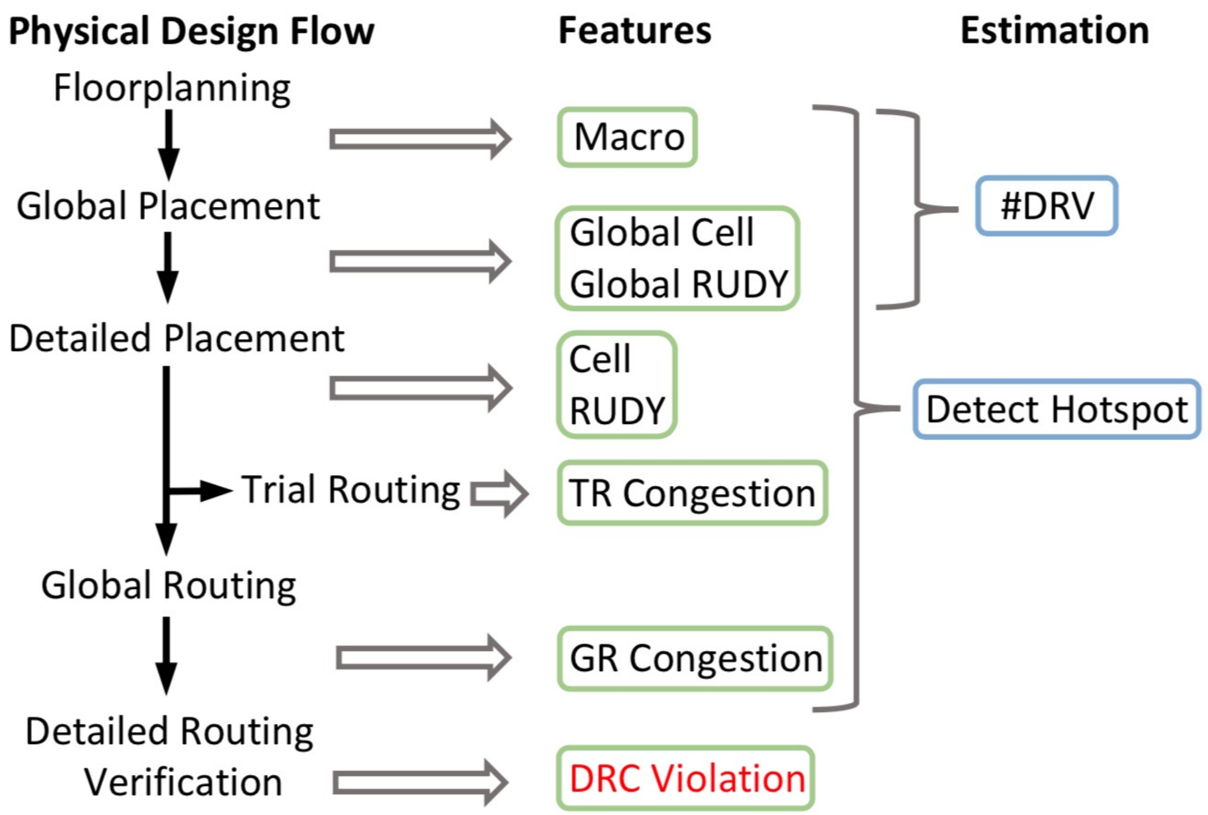
Features Extraction

- $X_{ij} = j\ th$ feature in $i\ th$ placement



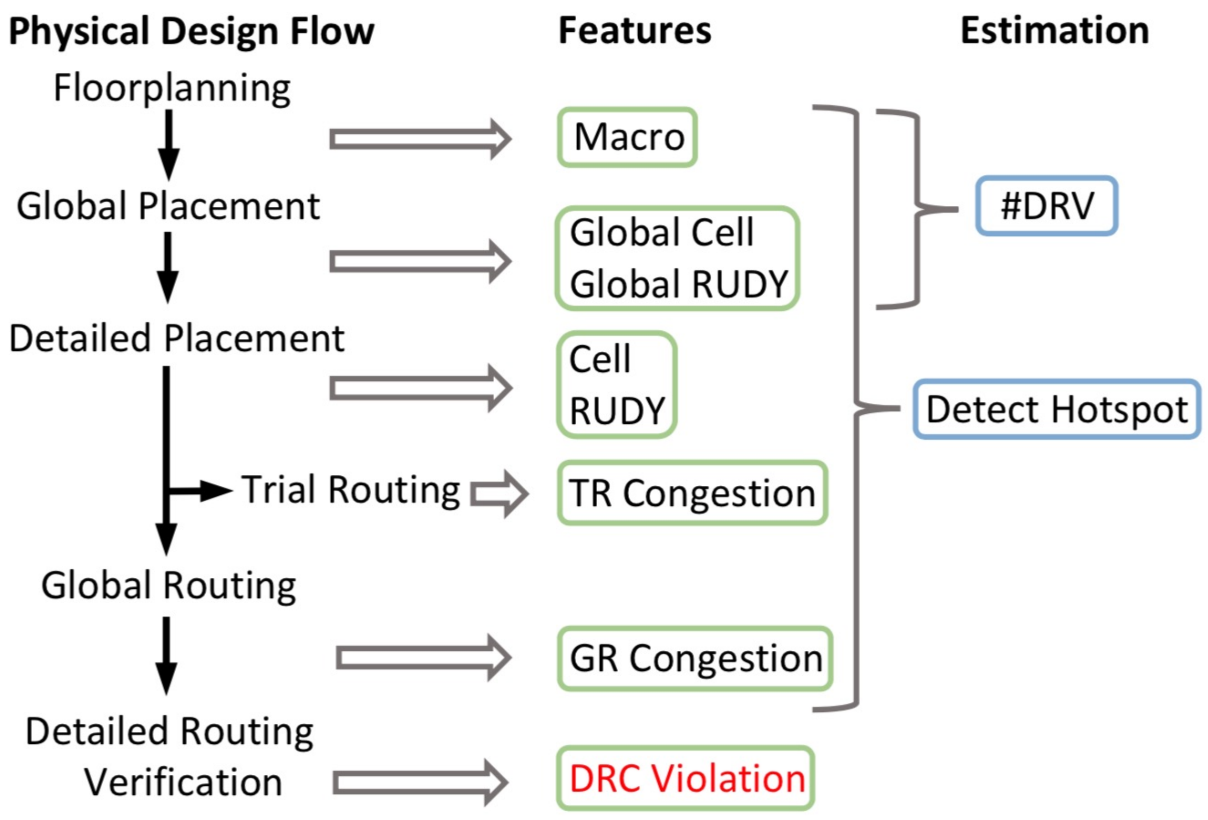
Features Extraction

- Macro:
  - region occupied by macros
  - density of macro pins in each layer
- Cell:
  - density of cells
  - density of cell pins
- Global cell:
  - cell features at global placement
- Global RUDY:
  - RUDY features calculated by global placement results



Features Extraction

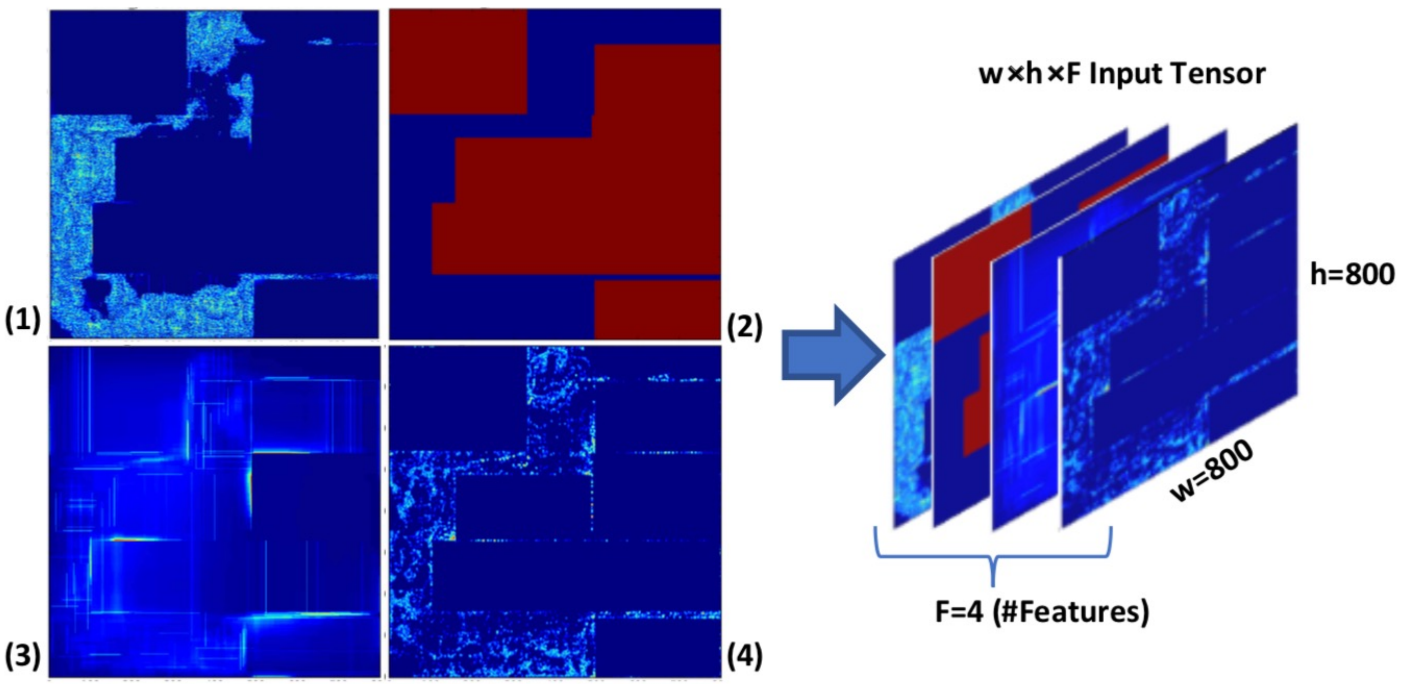
- RUDY
  - long-range RUDY
    - RUDY from long-range nets
  - short-range RUDY
    - DURY from short-range nets
  - RUDY pins
    - pins with density value equal to the RUDY value of its net
- Congestion
  - trial global routing congestion
  - global routing congestion
- DRC violation
  - prediction target / label



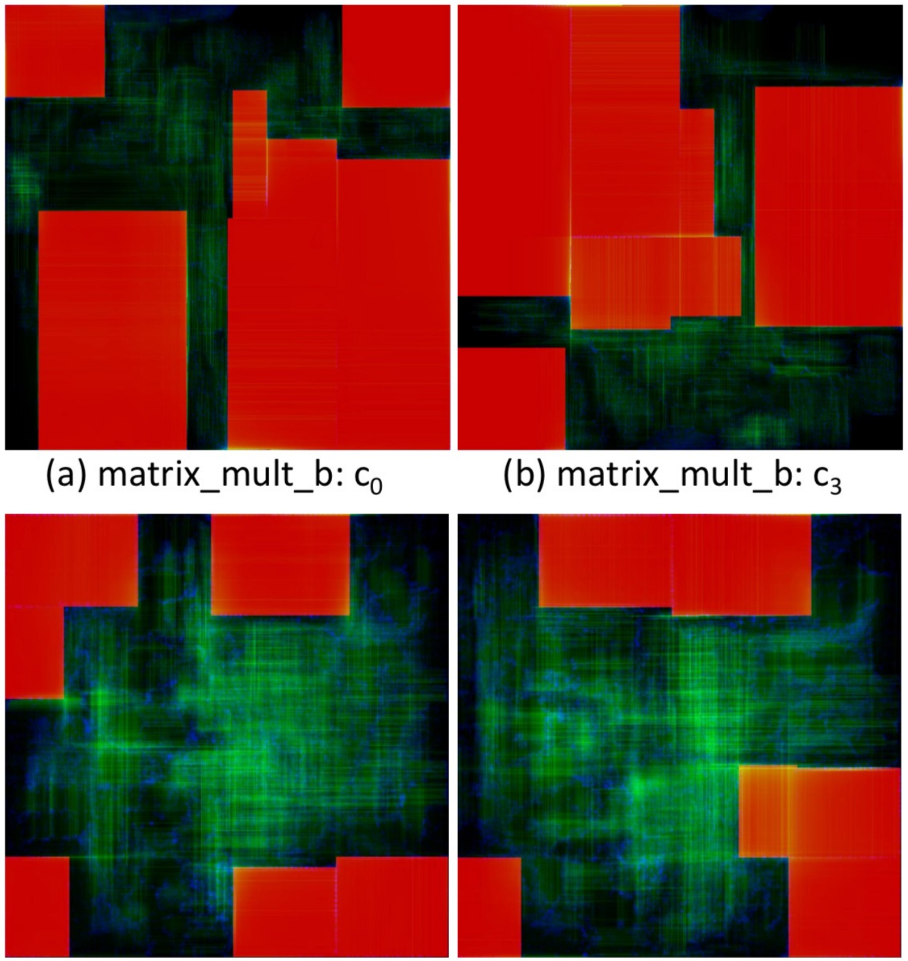
For i^{th} placement with size $w \times h$ and F features:

$$X_i \in \mathbb{R}^{w \times h \times F}$$

Features Extraction



Input tensor constructed by stacking 2D features:
(1) Pin density, (2) macro (3) long-range RUDY, (4) RUDY pins



(a) matrix_mult_b: c_0 (b) matrix_mult_b: c_3
(c) edit_dist: c_0 (d) edit_dist: c_3

Input features for #DRV prediction.
Red: macro region
Green: global long-range RUDY
Blue: global RUDY pins

Proposed Model

Problem 1 (#DRV prediction). *Find an estimator $f^*_{\#DRV}$ of DRV count in a placement:*

$$f_{\#DRV} : X_i^{(\#DRV)} \in \mathbb{R}^{w \times h \times F_1} \rightarrow y_i \in \mathbb{N}$$

$$f^*_{\#DRV} = \arg\min_f Loss(f(X_i^{(\#DRV)}),\ y_i)$$

Convolutional Neural Network (CNN)

Problem 2 (Hotspot prediction). *Find a detector $f^*_{hotspot}$ of hotspots. It reports locations of all DRC hotspots in a placement.*

$$f_{hotspot} : X_i^{(hotspot)} \in \mathbb{R}^{w \times h \times F_2} \rightarrow V_i \in \{0, 1\}^{w \times h}$$

$$f^*_{hotspot} = \arg\min_f Loss(f(X_i^{(hotspot)}),\ V_i)$$

$$Y_i \in \mathbb{R}^{w \times h} \qquad V_{i_{mn}} = \mathbb{1}(Y_{i_{mn}} > \epsilon)$$

Fully Convolutional Network (FCN)

Proposed Model- #DRV Prediction

Algorithm 1 Algorithm of RouteNet for #DRV Prediction

Input: Number of training placements: N, Features: $\{X_i \in \mathbb{R}^{w \times h \times 3} \mid i \in [1, N]\}$, Targets: $\{y_i \in \mathbb{R} \mid i \in [1, N]\}$

Preprocess:

for each int $i \in [1, N]$ **do**
    Resize $X_i \in \mathbb{R}^{w \times h \times 3}$ into $X_i^{\#DRV} \in \mathbb{R}^{224 \times 224 \times 3}$
Find 25%, 50%, 75% quantizes of y_i: q_1, q_2, q_3
for each int $i \in [1, N]$ **do**
    $C_i \leftarrow 0$
    for each int $t \in [1, 3]$ **do**
        if $y_i > q_t$ **then**
            $C_i \leftarrow t$, **break**
Form dataset $\{(X_i^{\#DRV}, C_i) | i \in [1, N]\}$
Training set $\{(X_i^{\#DRV}, C_i) \mid C_i = 0 \text{ or } C_i = 3\}$

Training:

Get pretrained ResNet18 $f_{Res} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}^{1000}$
Replace output layer, s.t. $f_{\#DRV} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}$
Choose MSE as loss function, SGD for optimization
Train $f_{\#DRV}$ with preprocessed dataset for ~30 epoches

Output: $f_{\#DRV}$ estimating #DRV level

Resize input to 224*224, to utilize models pre-trained on images with size 224*224

Proposed Model - #DRV Prediction

Algorithm 1 Algorithm of RouteNet for #DRV Prediction

Input: Number of training placements: N, Features: $\{X_i \in \mathbb{R}^{w \times h \times 3} \mid i \in [1, N]\}$, Targets: $\{y_i \in \mathbb{R} \mid i \in [1, N]\}$

Preprocess:

for each int $i \in [1, N]$ **do**
  Resize $X_i \in \mathbb{R}^{w \times h \times 3}$ into $X_i^{\#DRV} \in \mathbb{R}^{224 \times 224 \times 3}$

Find 25%, 50%, 75% quantizes of y_i: q_1, q_2, q_3
for each int $i \in [1, N]$ **do**
  $C_i \leftarrow 0$
  for each int $t \in [1, 3]$ **do**
    if $y_i > q_t$ **then**
      $C_i \leftarrow t$, **break**

Form dataset $\{(X_i^{\#DRV}, C_i) | i \in [1, N]\}$
Training set $\{(X_i^{\#DRV}, C_i) \mid C_i = 0 \ or \ C_i = 3\}$

Training:

Get pretrained ResNet18 $f_{Res} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}^{1000}$
Replace output layer, s.t. $f_{\#DRV} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}$
Choose MSE as loss function, SGD for optimization
Train $f_{\#DRV}$ with preprocessed dataset for ~30 epoches

Output: $f_{\#DRV}$ estimating #DRV level

Assign placements to 4 different classes (c_0, c_1, c_2, c_3) based on their level of violations (#DRV)

c_0 represents least #DRV, while c_3 represents most

Proposed Model - #DRV Prediction

Algorithm 1 Algorithm of RouteNet for #DRV Prediction

Input: Number of training placements: N, Features: $\{X_i \in \mathbb{R}^{w \times h \times 3} \mid i \in [1, N]\}$, Targets: $\{y_i \in \mathbb{R} \mid i \in [1, N]\}$

Preprocess:

1: **for** each int $i \in [1, N]$ **do**
2: Resize $X_i \in \mathbb{R}^{w \times h \times 3}$ into $X_i^{\#DRV} \in \mathbb{R}^{224 \times 224 \times 3}$
3: Find 25%, 50%, 75% quantizes of y_i: q_1, q_2, q_3
4: **for** each int $i \in [1, N]$ **do**
5: $C_i \leftarrow 0$
6: **for** each int $t \in [1, 3]$ **do**
7: **if** $y_i > q_t$ **then**
8: $C_i \leftarrow t$, **break**
9: Form dataset $\{(X_i^{\#DRV}, C_i) | i \in [1, N]\}$
10: Training set $\{(X_i^{\#DRV}, C_i) \mid C_i = 0 \text{ or } C_i = 3\}$

Training:

1: Get pretrained ResNet18 $f_{Res} : \mathbb{R}^{224 \times 224 \times 3} \to \mathbb{R}^{1000}$
2: Replace output layer, s.t. $f_{\#DRV} : \mathbb{R}^{224 \times 224 \times 3} \to \mathbb{R}$
3: Choose MSE as loss function, SGD for optimization
4: Train $f_{\#DRV}$ with preprocessed dataset for ~30 epoches

Output: $f_{\#DRV}$ estimating #DRV level

Download a pre-trained CNN model named ResNet18

Proposed Model - #DRV Prediction

Algorithm 1 Algorithm of RouteNet for #DRV Prediction

Input: Number of training placements: N, Features: $\{X_i \in \mathbb{R}^{w \times h \times 3} \mid i \in [1, N]\}$, Targets: $\{y_i \in \mathbb{R} \mid i \in [1, N]\}$

Preprocess:

```
for each int i ∈ [1, N] do
    Resize X_i ∈ R^{w×h×3} into X_i^{#DRV} ∈ R^{224×224×3}
Find 25%, 50%, 75% quantizes of y_i: q_1, q_2, q_3
for each int i ∈ [1, N] do
    C_i ← 0
    for each int t ∈ [1, 3] do
        if y_i > q_t then
            C_i ← t, break
Form dataset {(X_i^{#DRV}, C_i)|i ∈ [1, N]}
Training set {(X_i^{#DRV}, C_i) | C_i = 0 or C_i = 3}
```

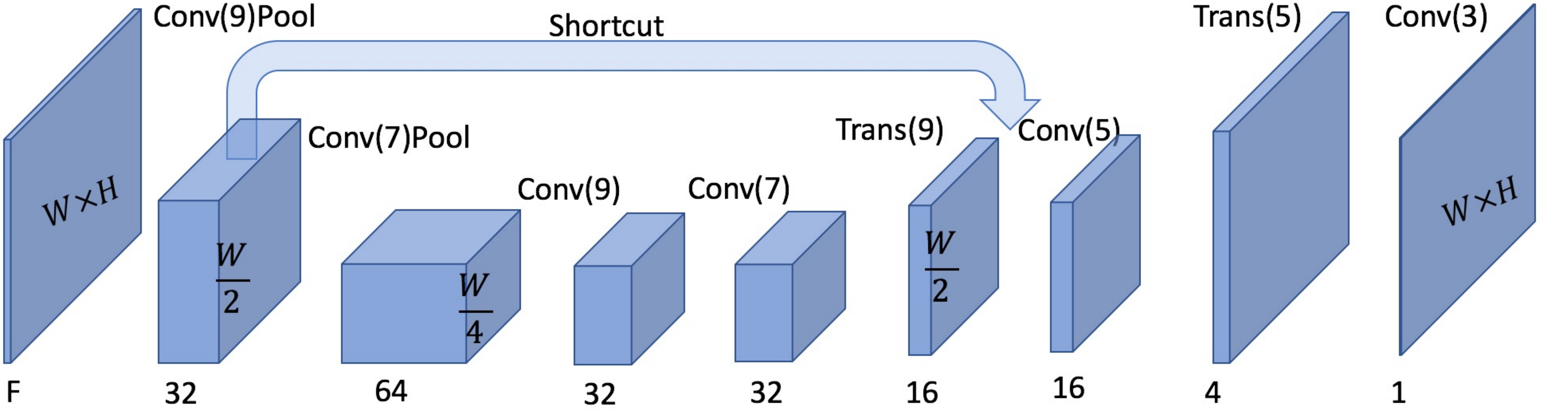
Training:

```
Get pretrained ResNet18 f_Res : R^{224×224×3} → R^{1000}
Replace output layer, s.t. f_#DRV : R^{224×224×3} → R
Choose MSE as loss function, SGD for optimization
Train f_#DRV with preprocessed dataset for ~30 epoches
```

Fine-tune CNN with preprocessed data

Output: $f_{\#DRV}$ estimating #DRV level

Proposed Model- Hotspot Detection



Filter size indicated in ()

$$Y_{i_{mn}}^{clip} = min(Y_{i_{mn}}, c)$$

$$Loss = \sum_{i=1}^{N} \sum_{m=1}^{w} \sum_{n=1}^{h} ||f_{hotspot}(X_{i_{mn}}) - Y_{i_{mn}}^{clip}||_2 + \lambda ||W||_2$$

Pixel-wise loss function

Data

- Five designs from ISPD 2015
- ~300 different placements by placing macros in different way
- When each design tested, model trained only on four other designs
- SVM and Logistic Regression (LR) methods for comparison

Circuit Name	#Macros	#Cells	#Nets	Width (μm)	#Placements
des_perf	4	108288	110283	900	600
edit_dist	6	127413	131134	800	300
fft	6	30625	32088	800	300
matrix_mult_a	5	149650	154284	1500	300
matrix_mult_b	7	146435	151614	1500	300

#DRV Prediction Evaluation

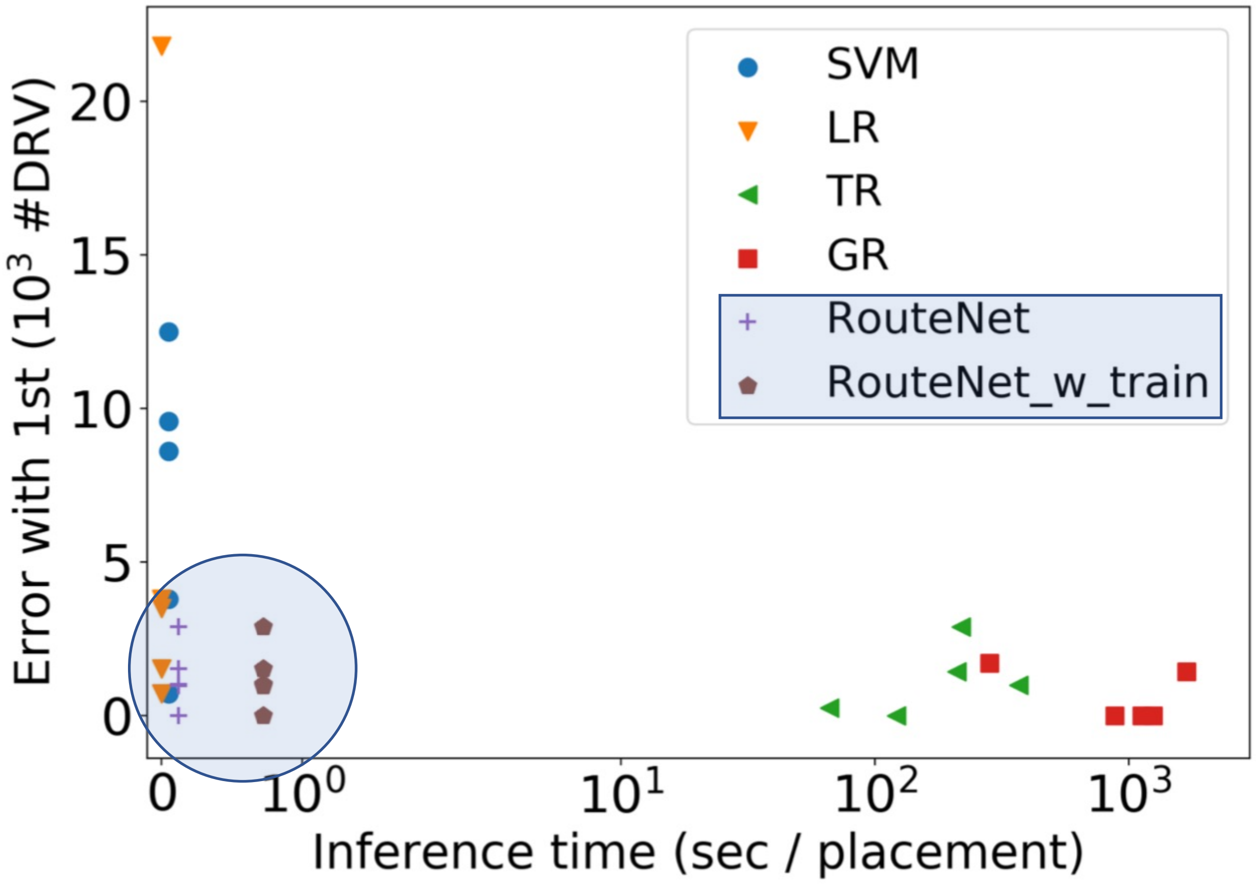
- How methods recognize placements with the lowest #DRV level (c_0)

- The quality of placements selected by each method
  - The **best rank** of top ten placements predicted to have least #DRV

Circuit Name	$c_0/c_1+c_2+c_3$ accuracy (%)					Best rank in top 10				
	SVM	LR	TR	GR	Route Net	SVM	LR	TR	GR	Route Net
des_perf	63	74	80	77	80	87^{th}	15^{th}	2^{nd}	1^{st}	2^{nd}
edit_dist	69	68	78	77	76	17^{th}	17^{th}	3^{rd}	3^{rd}	2^{nd}
fft	66	62	73	70	75	6^{th}	6^{th}	2^{nd}	33^{rd}	1^{st}
matrix_mult_a	66	65	78	74	72	30^{th}	5^{th}	1^{st}	1^{st}	5^{th}
matrix_mult_b	63	62	76	73	76	22^{nd}	93^{rd}	4^{th}	1^{st}	4^{th}
Average	65	66	77	74	76	32^{nd}	27^{th}	2^{nd}	8^{th}	3^{rd}

#DRV Prediction Evaluation

- Y: gap between the ‘best in 10’ and the actually 1st-ranked placement with least #DRV
- X: inference time taken for each method
- RouteNet achieves low inference time and high accuracy at the same time



DRC Hotspot Detection Evaluation

- Same decision threshold is used for all designs
- Slight different FPR, but all under 1%
- RouteNet is superior to all methods and improves global routing accuracy by 50%

Circuit Name	FPR (%)	TPR (%) TR	GR	LR	SVM	RouteNet
des_perf	0.54	17	56	54	42	74
edit_dist	1.00	25	36	38	28	64
fft	0.30	21	45	54	31	71
matrix_mult_a	0.21	13	30	34	12	49
matrix_mult_b	0.24	13	37	41	20	53
Average	0.46	18	41	44	27	62

		Prediction Result: Positive	Negative
Label	Positive	TP	FN
	Negative	FP	TN

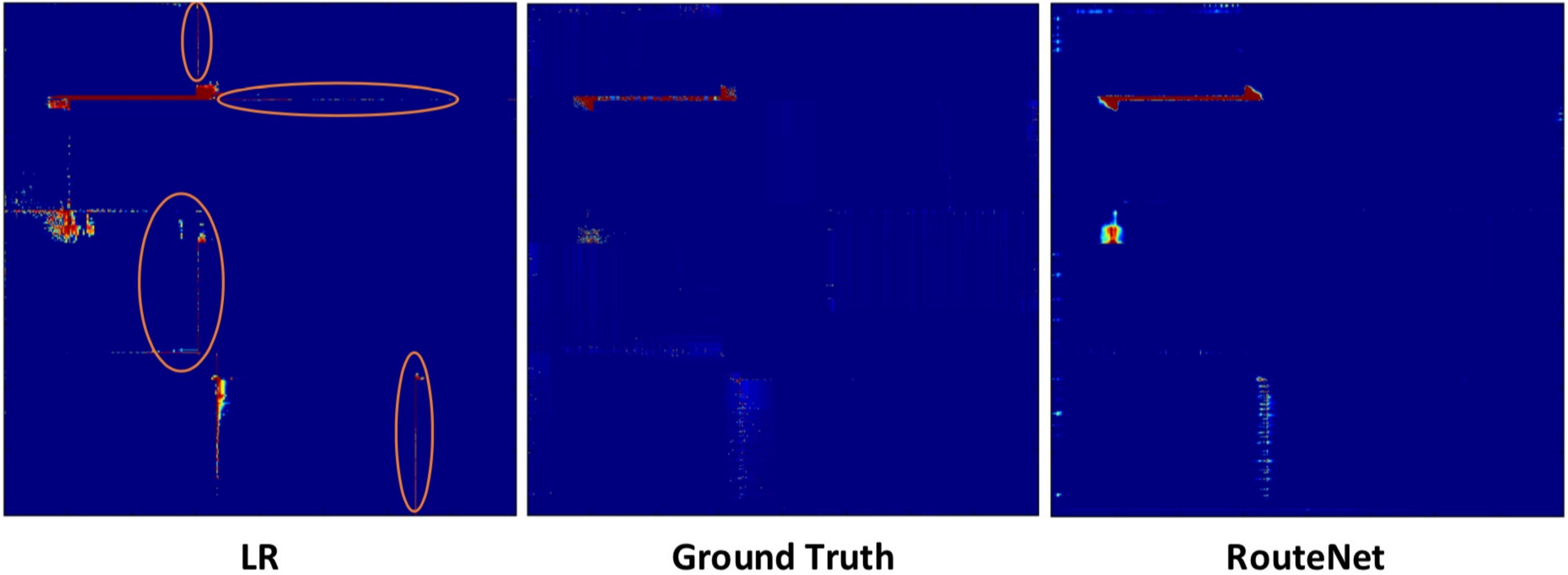
Evaluation

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

TPR (True Positive Rate)
FPR (False Positive Rate)

DRC Hotspot Detection Evaluation



DRC Hotspot Detection Evaluation

- Variations of FCN
  - Infer seen: Training and inference on different placements of the same circuit
  - Less data: Trained on data from less designs
  - No short: Shortcut structure is removed
  - Less conv: Three convolutional layers are removed
  - No pool: Pooling layers are removed

Circuit Name	FPR (%)	TPR (%)					
		Infer seen	Less data	No short	Less conv	No pool	Route Net
des_perf	0.54	77	71	71	73	68	74
edit_dist	1.00	68	61	63	62	55	64
fft	0.30	74	70	68	68	69	71
matrix_mult_a	0.21	51	46	45	45	45	49
matrix_mult_b	0.24	58	50	51	50	50	53
Average	0.46	66	60	60	60	57	62

Importance of large receptive region and global information.

DRC Hotspot Detection Evaluation

- Variations of baselines
  - 5×5: Use window size of 5×5 grid cells to capture neighboring features of each grid cell.
  - 9×9: 9 × 9 grid cells of window size.

Circuit Name	FPR (%)	TPR (%)					
		LR	5×5 LR	9×9 LR	SVM	5×5 SVM	9×9 SVM
des_perf	0.54	54	58	58	42	47	29
edit_dist	1.00	38	39	38	28	29	20
fft	0.30	54	56	54	31	41	23
matrix_mult_a	0.21	34	36	35	12	32	9
matrix_mult_b	0.24	41	44	42	20	39	16
Average	0.46	44	47	45	27	38	19

Large receptive region gives better results

But even larger window blurs local information

RouteNet is better choice

Conclusion

- We propose RouteNet:
  - Enables a global view for less homogeneous layout
  - Faster overall routability forecast at placement
  - More accurate hotspot detection at global routing

Thanks

zhiyao.xie@duke.edu
yiran.chen@duke.edu
haoxingr@nvidia.com
jianghu@tamu.edu