Context-Aware DFM Rule Analysis and Scoring Using Machine Learning

Vikas Tripathi, Valerio Perez, Yongfu Li, Zhao Chuan Lee, I-Lun Tseng, and Jonathan Ong
GLOBALFOUNDRIES, 60 Woodlands Ind. Park D Street 2, 738406, Singapore

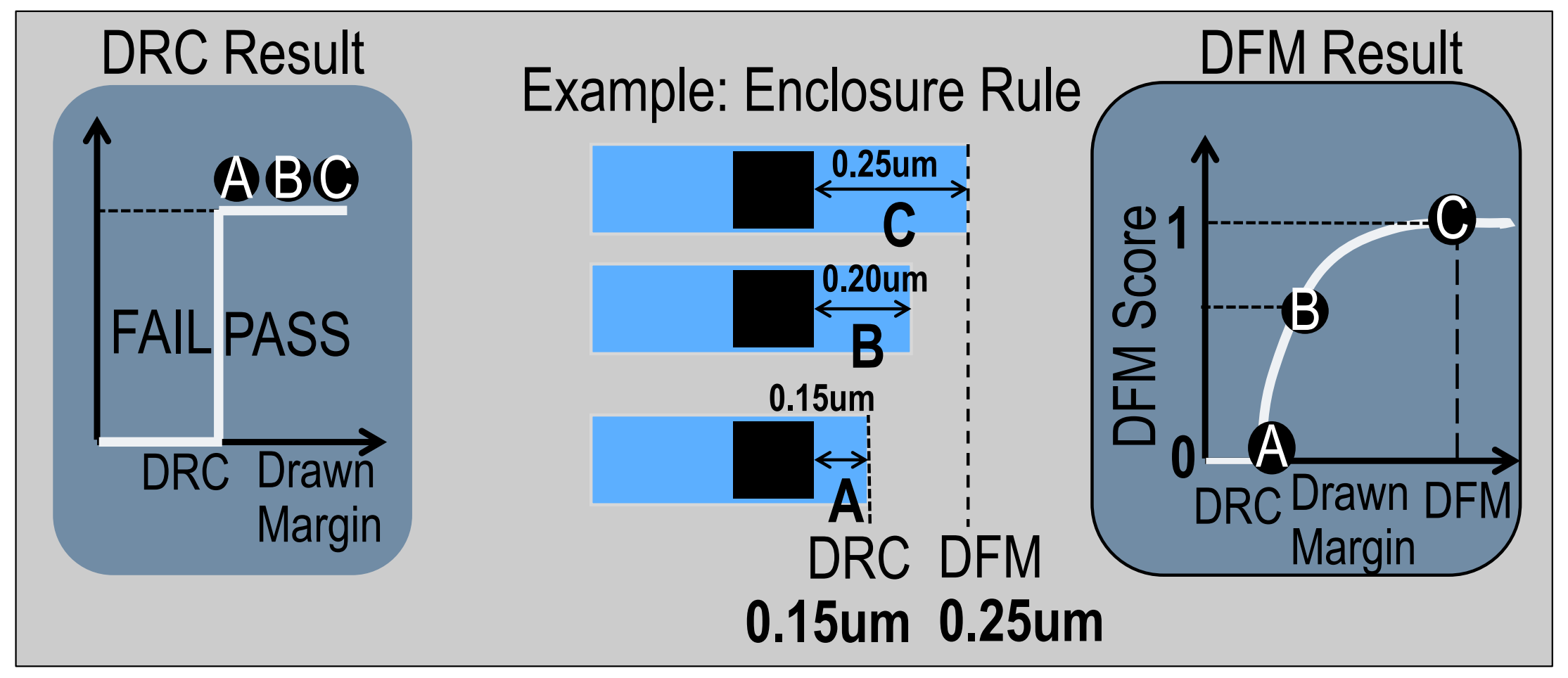
Introduction

What are DFM Rules?

- ❑ Extension of DRC rules (recommended rules)
- ❑ Improve designs for better manufacturability

DFM Scoring Methodology

$$Score = f(Drawn_{value}, DRC_{value}, DFM_{value})$$

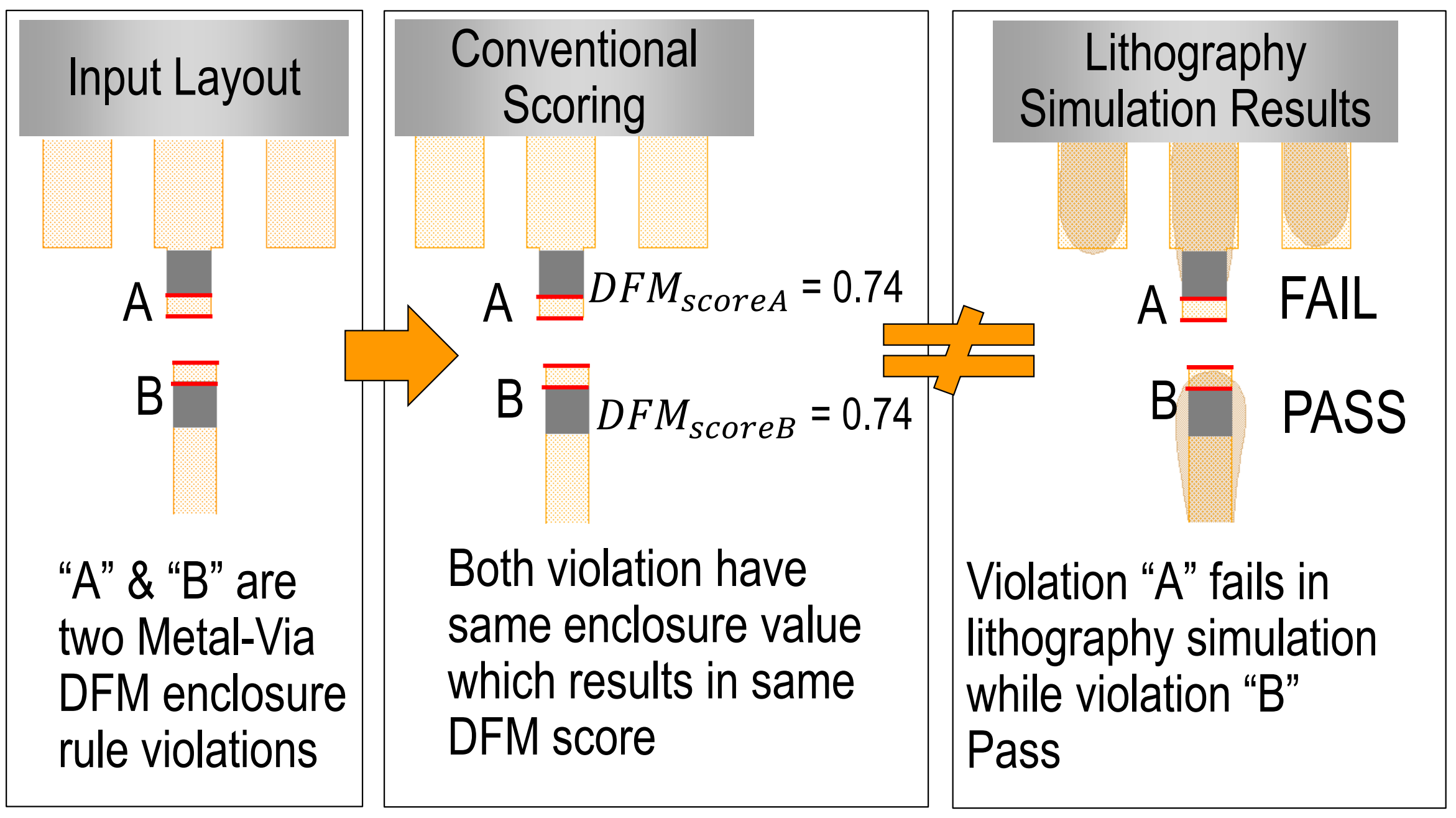


- ❑ Evaluating manufacturability of chip designs
- ❑ Based on the severity of the DFM rule violation
- ❑ Higher scores represent better Manufacturability/Yield/Quality

Problem Statement and Motivation

- ❑ The role of layout context is very critical in printability of any layout shape and can significantly impact the effective margins required to print them lithography clean
- ❑ Propose to incorporate “layout-context” analysis in our DFM scoring methodology
- ❑ To align DFM rule base score more lithography-aware
- ❑ And provide more realistic rule checking and scoring results

Use-Case to illustrate the problem statement



“A” & “B” are two Metal-Via DFM enclosure rule violations

Both violation have same enclosure value which results in same DFM score

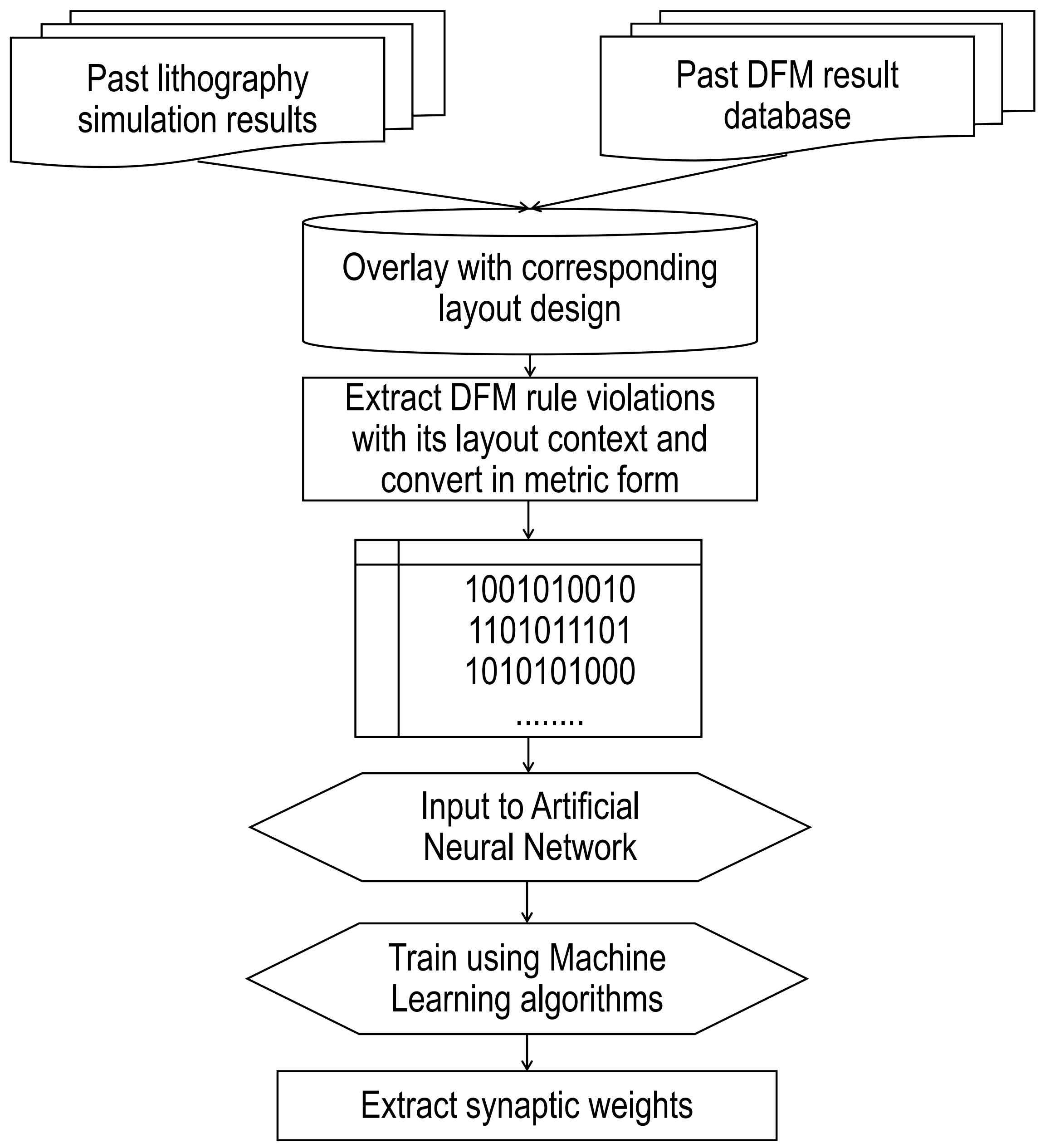
Violation “A” fails in lithography simulation while violation “B” Pass

- ❑ DFM score for violation A & B does not correlated with their lithography results
- ❑ This mismatch is due to the difference in their layout context, violation “A” printability significantly affected by neighboring line-ends and jogs while violation “B” does not have such neighboring shapes and thus has better printability
- ❑ This limitation creates a requirement of advanced techniques in which layout context can be analyzed while performing rule-based scoring

Proposed Method

- ❑ Propose a novel methodology for performing context-aware rule scoring
- ❑ This methodology is based on machine learning which uses past lithography simulation results to build training data set and use it to predict a given layout context for its probability of causing lithography failures

Flow overview



Past lithography simulation results

Past DFM result database

Overlay with corresponding layout design

Extract DFM rule violations with its layout context and convert in metric form

1001010010
1101011101
1010101000
....

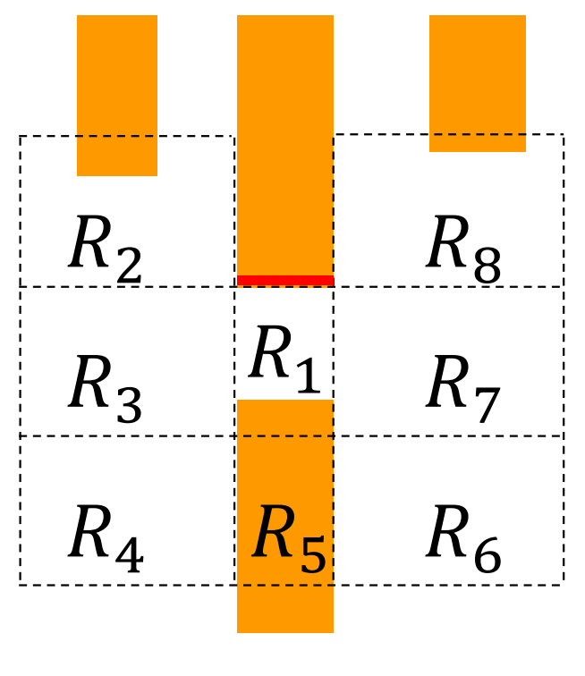
Input to Artificial Neural Network

Train using Machine Learning algorithms

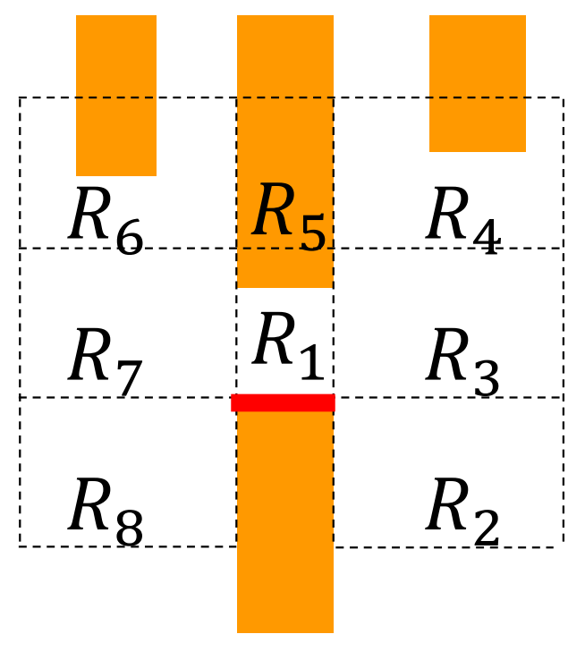
Extract synaptic weights

Data Extraction for Machine Learning

- ❑ Data is extracted from given layout context by dividing the context in 8 different regions. Each region is analyzed for layout polygon and vertices count
- ❑ These values are transformed in metric for training data set.
- ❑ If DFM violation overlay with lithography hotspot then expected outcome is “1” otherwise it will be “0”



Layout context of DFM rule violation “A”

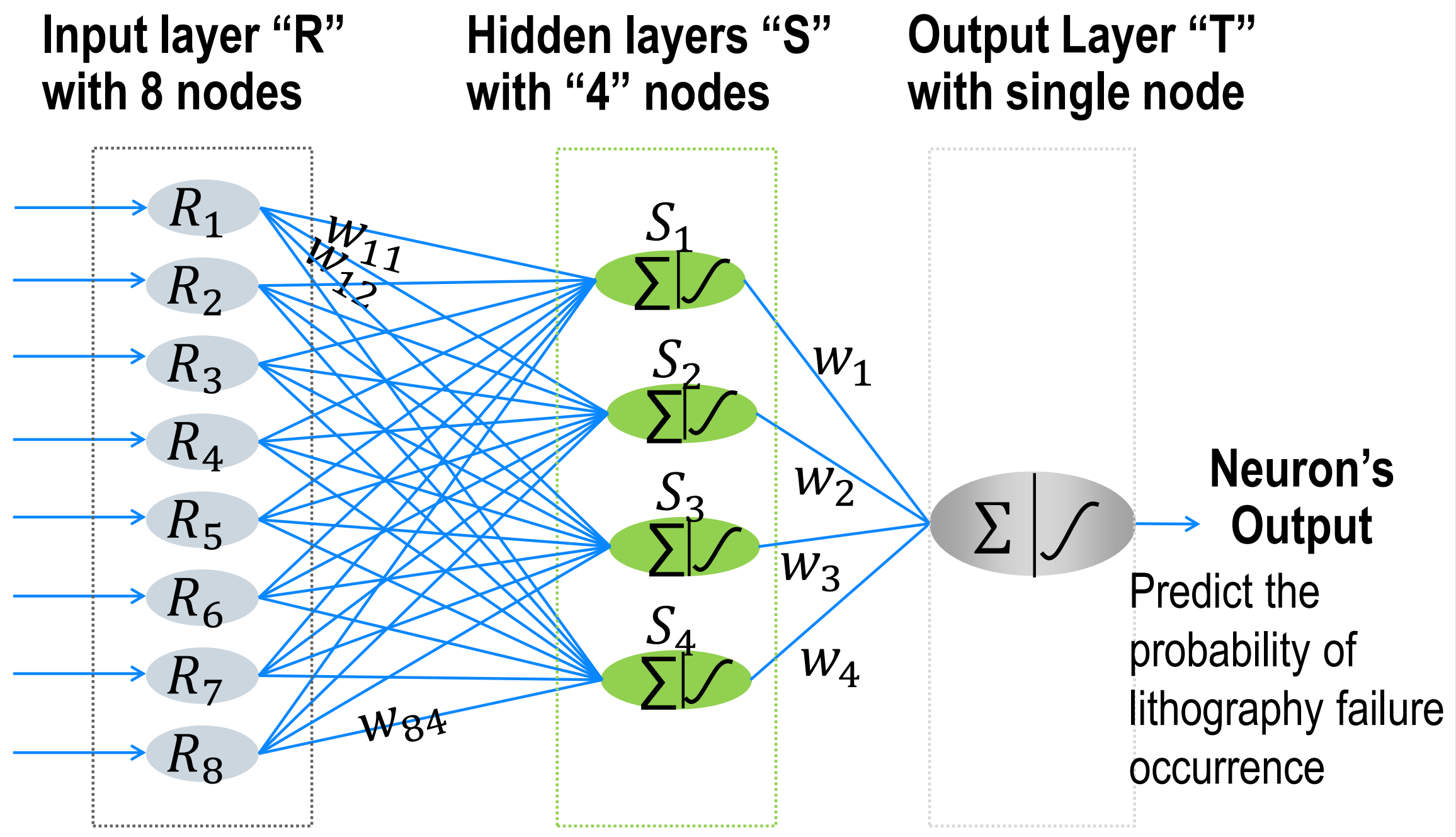


Layout context of DFM rule violation “B”

Case	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	Expected outcome
A	3	3	0	0	1	0	0	3	1
B	3	0	0	3	1	3	0	0	0

Artificial Neural Network Used

- ❑ In this methodology we have used Artificial Neural Network to predict the outcome

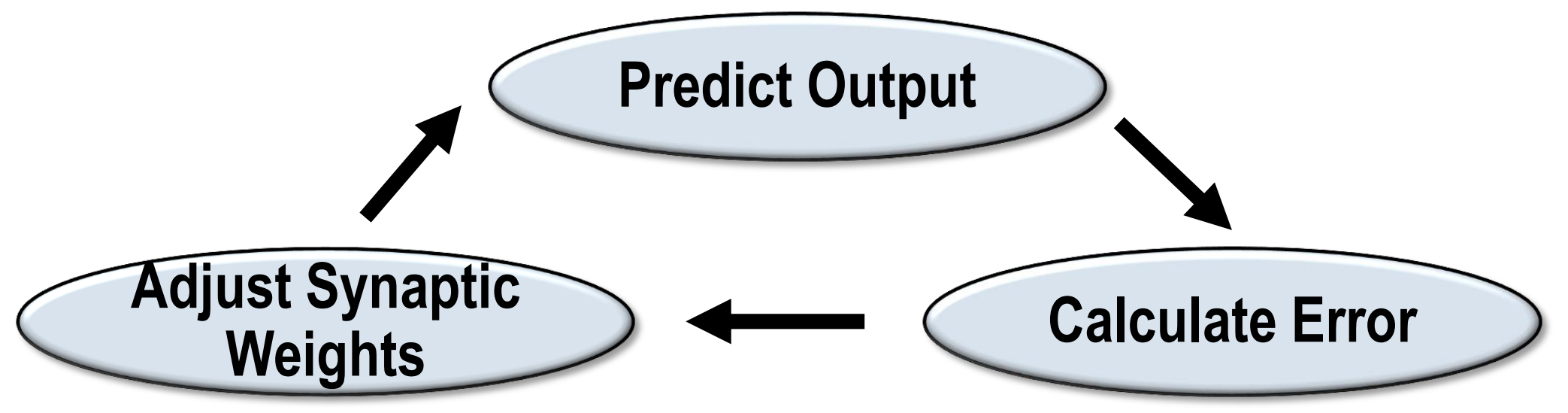


Predict the probability of lithography failure occurrence

- ❑ In this approach, We have used three layered neural network to get the optimum results
- ❑ The input layer consist of 8 input nodes which represents the value of each region extracted from layout context
- ❑ The outputs of input layer is connected with 4 hidden layer nodes using a weighted connection. These weights represent the significance of associated node in neural network
- ❑ The output of hidden layer nodes are connected to single output node using another weighted connection

Learning Cycle & Training Algorithm

- ❑ Network optimization and training process uses extracted data set with known results



1. Define ANN architecture (8:4:1) and initializes weights randomly

Provide random weights for synapsis w_{ij} connected from input to hidden layer and w_j connected from hidden layer to output neuron

2. Define initial neuron values based on sigmoid function

R_i = input layer vector , $s_{j_{out}} = \frac{1}{(1+e^{-s_{j_{in}}})}$, $s_{j_{in}} = \sum_{i=1}^{8} w_{ij} * R_i$

$t_{out} = \frac{1}{(1+e^{-t_{in}})}$, where $t_{in} = \sum_{j=1}^{4} s_j w_j$

3. Evaluate the error at the neuron output

$error_t = (x - t_{out})$

Where, x is the desired output, t_{out} is the actual output of the neuron

5. Apply gradient descent method to calculate error delta and adjust weights from hidden layer to output layer

$\delta_t = error_t * \frac{d(t_{out})}{d(t_{in})}$

Where, t_{in} is the weighted summation of all the hidden layer nodes

$\boldsymbol{\delta_{w_j} += S_{j_{out}} . \delta_t}$

6. Backward propagate the output neuron error delta to calculate error delta at hidden layer neuron and adjust weights from hidden layer to first layer

$error_{s_j} = \delta_t . w_j$, $\delta_s = error_{s_j} * \frac{d(s_{j_{out}})}{d(s_{j_{in}})}$,$\boldsymbol{\delta_{w_{ij}} += R_j . \delta_s}$

7. Go to step 4 for a certain number of iterations, or until the error is less than a pre-specified value

Context-Aware Scoring

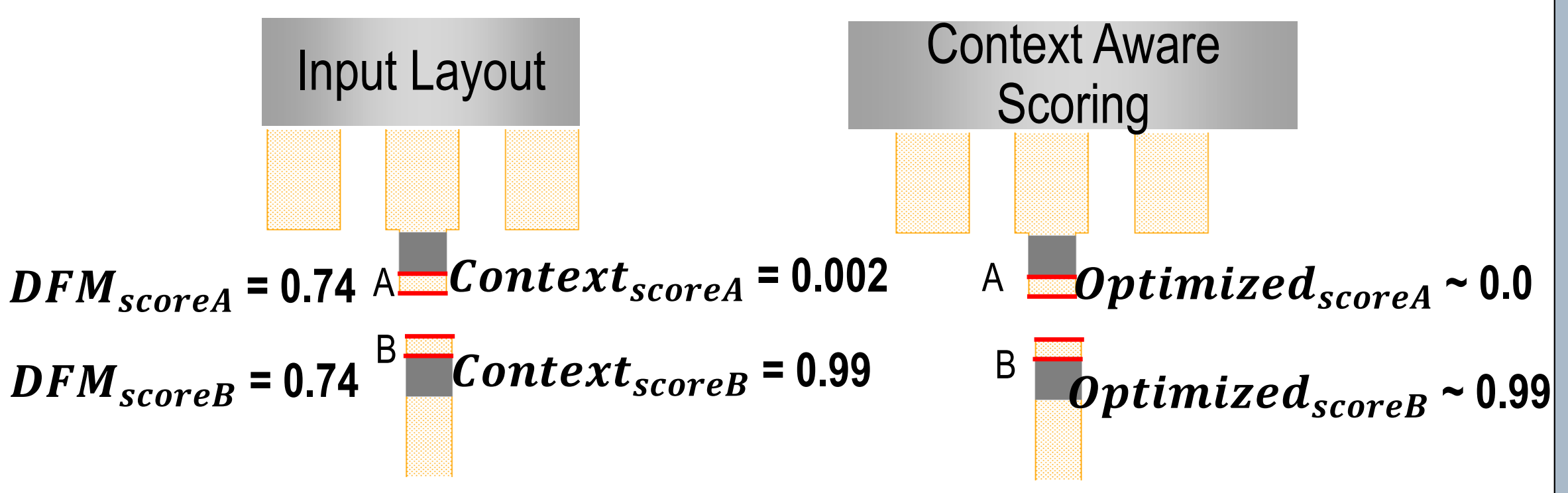
- ❑ In this implementation, a total of 36 synapses used to connect 8 input nodes to 4 hidden nodes and 1 output node
- ❑ By training the network using learning algorithm and training dataset, 36 weights for these synapses get adjusted
- ❑ These weights are extracted and later used in scoring equation during rule analysis

	S_1	S_2	S_3	R_4
T_{Out}	w_1	w_2	w_3	w_4
R_1	w_{11}	w_{12}	w_{13}	w_{14}
R_2	w_{21}	w_{22}	w_{23}	w_{24}
R_3	w_{31}	w_{32}	w_{33}	w_{34}
R_4	w_{41}	w_{42}	w_{43}	w_{44}
R_5	w_{51}	w_{52}	w_{53}	w_{54}
R_6	w_{61}	w_{62}	w_{63}	w_{64}
R_7	w_{71}	w_{72}	w_{73}	w_{74}
R_8	w_{81}	w_{82}	w_{83}	w_{84}

Extracted synaptic weights to be used in context weightage calculation

$$\boldsymbol{Context_{Score} = 1 - Context_{Weigtage}}$$
$$\boldsymbol{Optimized_{Score} = f(DFM_{Score}, Context_{Score})}$$

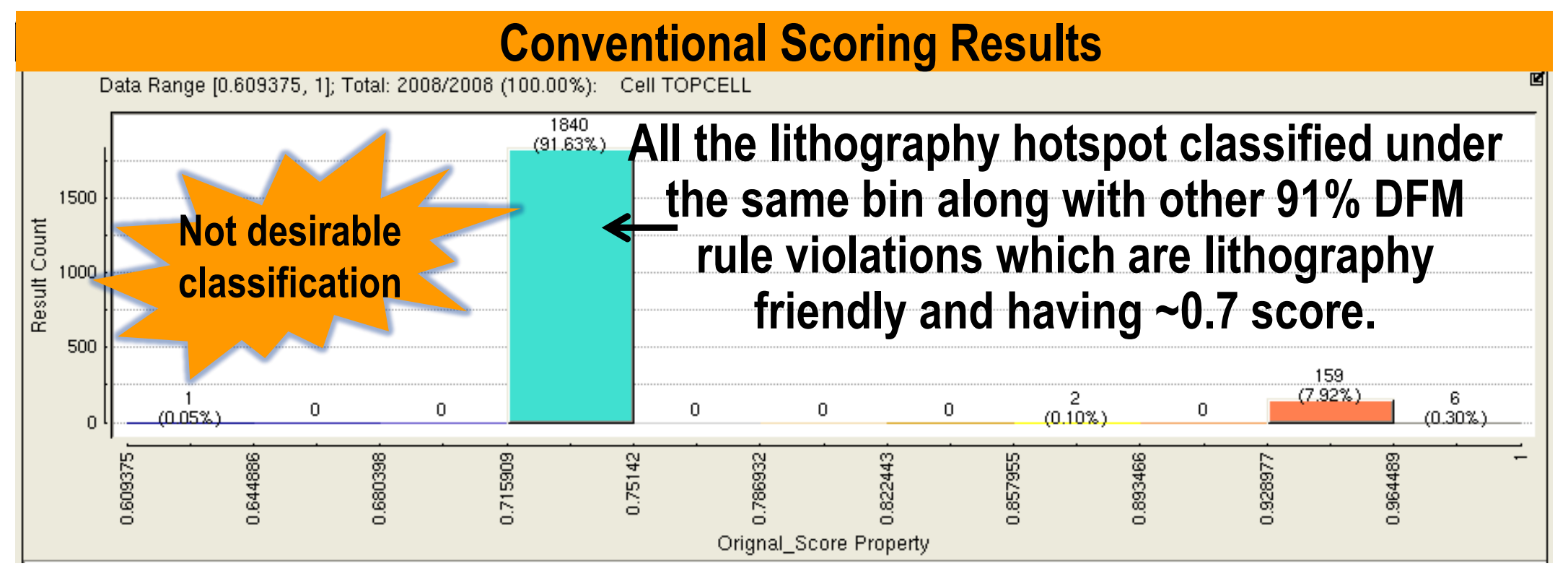
Use-Case Result


Input Layout
Context Aware Scoring
DFM_{scoreA} = 0.74
A
$Context_{scoreA}$ = 0.002
DFM_{scoreB} = 0.74
B
$Context_{scoreB}$ = 0.99
A
$Optimized_{scoreA}$ ~ 0.0
B
$Optimized_{scoreB}$ ~ 0.99

Results

- ❑ Test design with ~ 2000 DFM metal-via enclosure violations and 8 lithography hotspots is used
- ❑ Conventional scoring approach classify all the lithography hotspots with other 91.63% of total DFM rule violations which are lithography clean

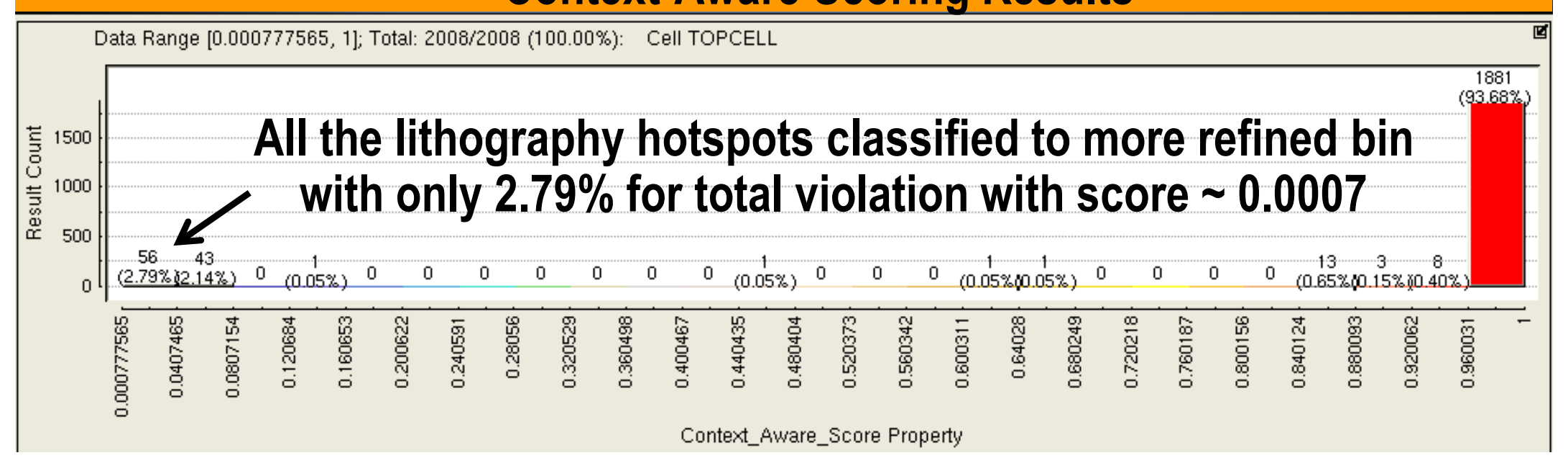
Conventional Scoring Results



All the lithography hotspot classified under the same bin along with other 91% DFM rule violations which are lithography friendly and having ~0.7 score.

- ❑ With this proposed method, score is more optimized based on its context weightage

Context-Aware Scoring Results



All the lithography hotspots classified to more refined bin with only 2.79% for total violation with score ~ 0.0007

- ❑ More refined classification of DFM rule violations. The violations with scores close to ~0.0 are more likely to have a printability issue and can be consider as potential weak points

Conclusion

The demonstrated context-aware DFM rule scoring using machine learning technique has benefited in

- ❑ Improve DFM Score accuracy and its correlation with lithography results
- ❑ Help in more desirable classification of critical DFM rule violations
- ❑ Help designers to priorities fixing of critical violations as to improve design quality for better manufacturability