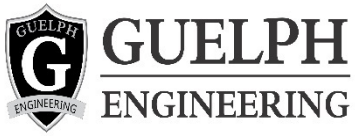
Integrating Machine Learning within FPGA Placement

Shawki Areibi, Gary Grewal
Guelph FPGA CAD Group
https://fpga.socs.uoguelph.ca

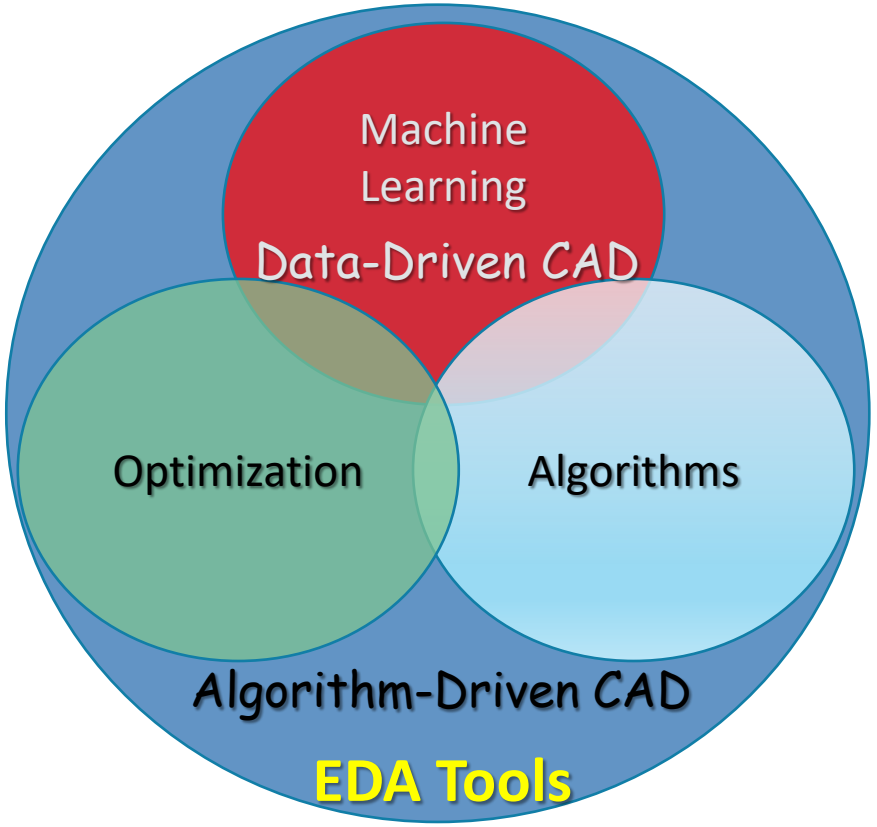


GUELPH
ENGINEERING

School of
Computer
Science

Why Machine Learning in EDA?

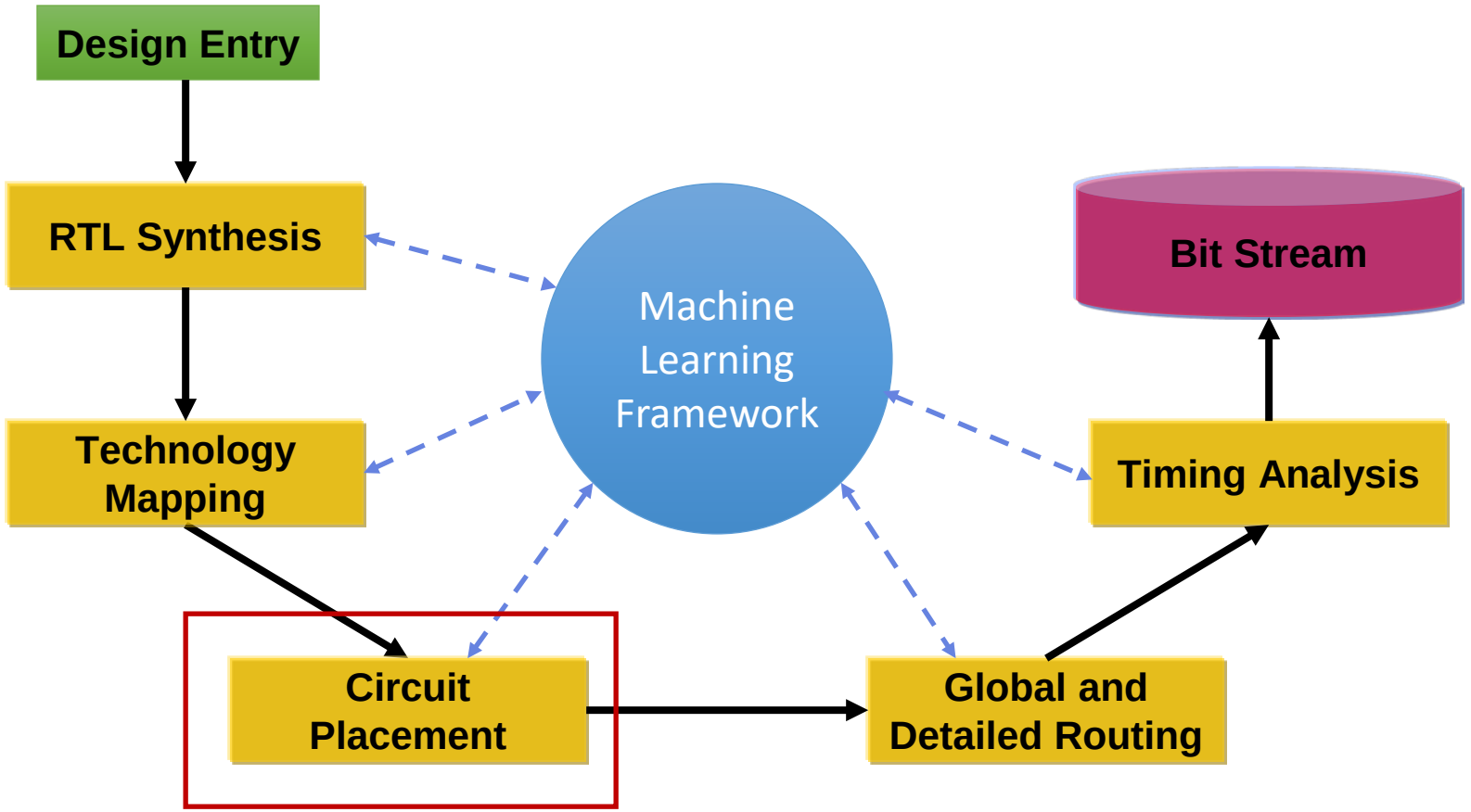
- Limitations of current Algorithmic based EDA/CAD flows:
  - Some problems are too complex for handwritten rules
    - Congestion Estimation, Routing Prediction, Selection of Optimal Flow, etc.
  - The rules of a task are constantly changing
    - FPGA architectures and constraints imposed, etc.
  - Nature of the data itself keeps changing
    - Increasing size and complexity of architectures and applications
- Benefits of Machine Learning for CAD:
  - ✓Data-driven
  - ✓No explicit programming
  - ✓May assist in cutting CPU time & improve QOR
  - ✓Provides guidance to the flow



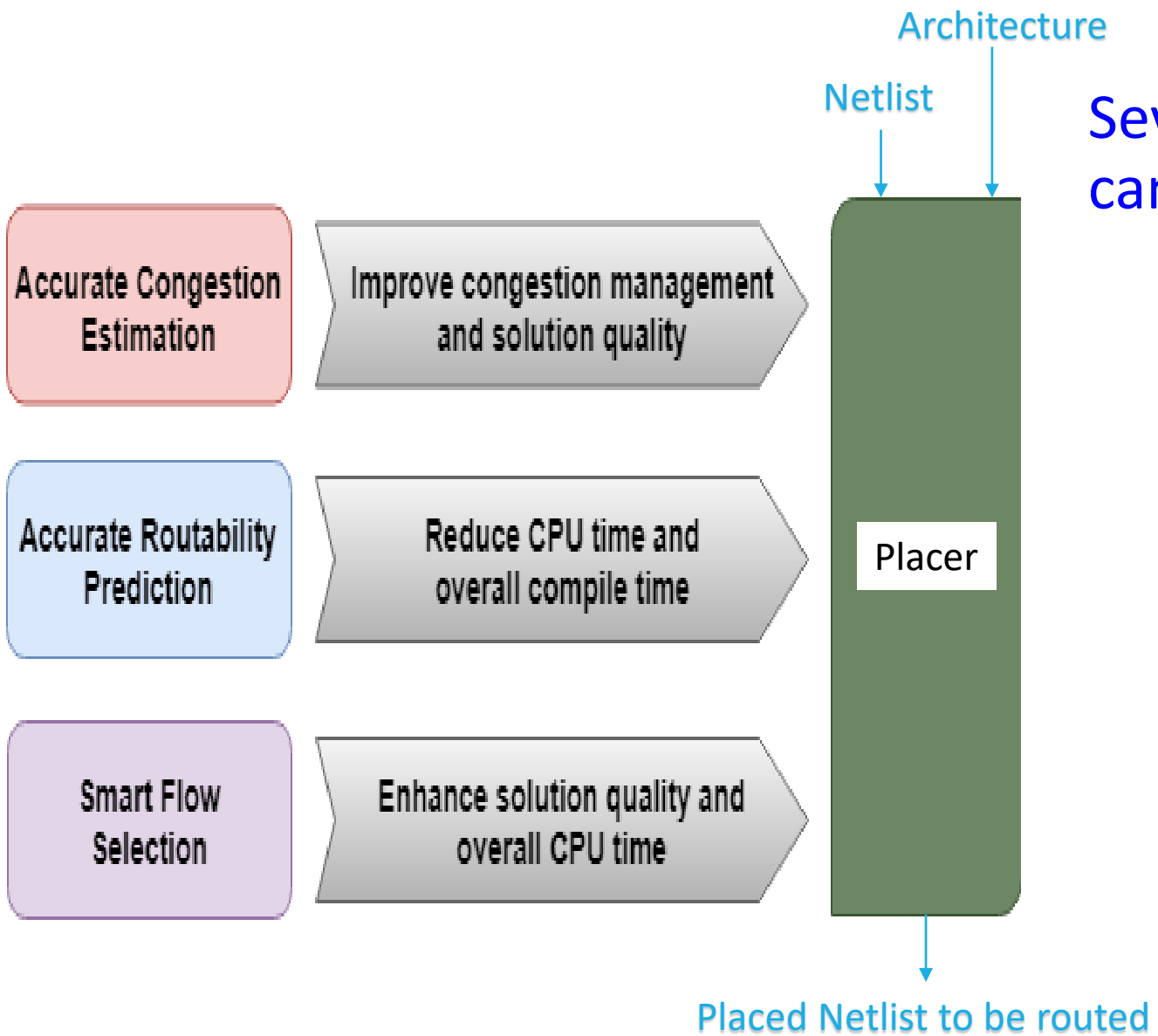
ML for CAD seems like a perfect match!

Machine Learning in FPGA CAD Flow

Machine Learning + FPGA CAD Algorithms = Smart Flows



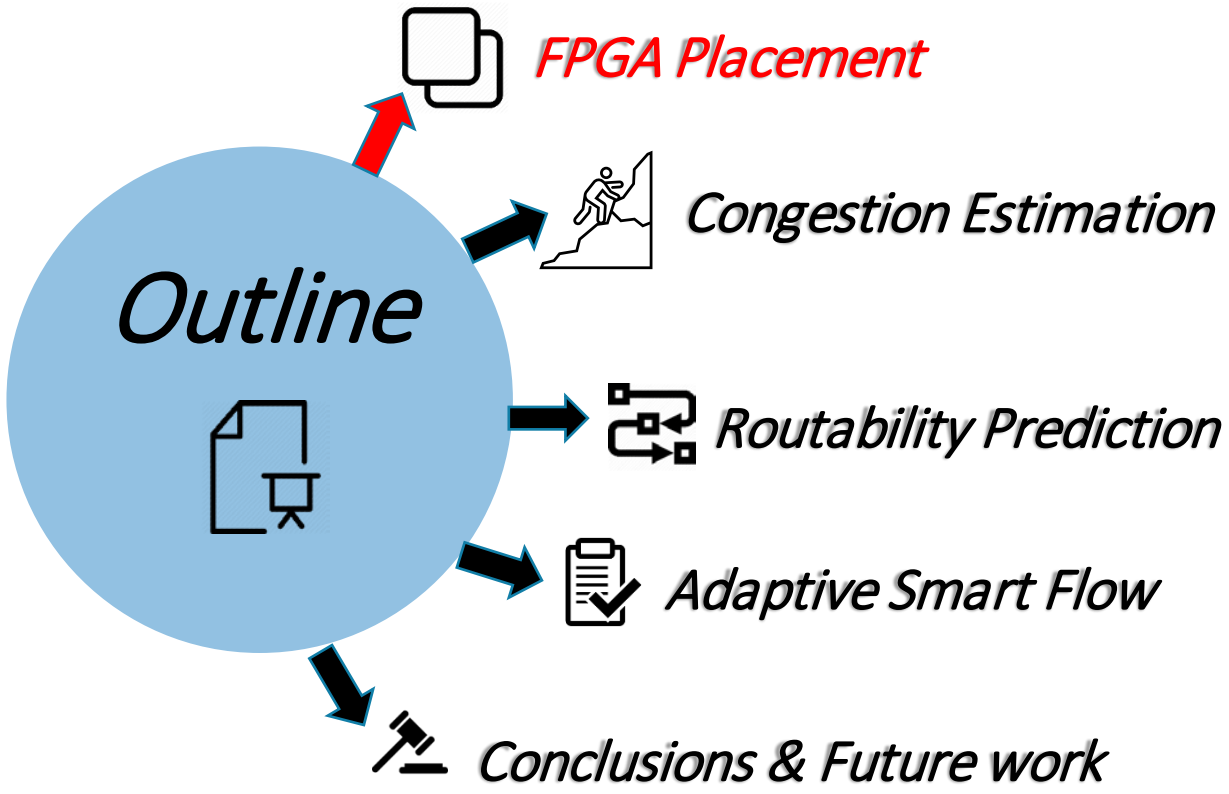
Examples of ML in FPGA Placement



Several problems in FPGA placement can be targeted using ML including:

1. Congestion:
   - Estimation,
   - Forecasting,
   - Management
2. Routability prediction
3. Flow selection/recommendation
4. Timing estimation
5. …….

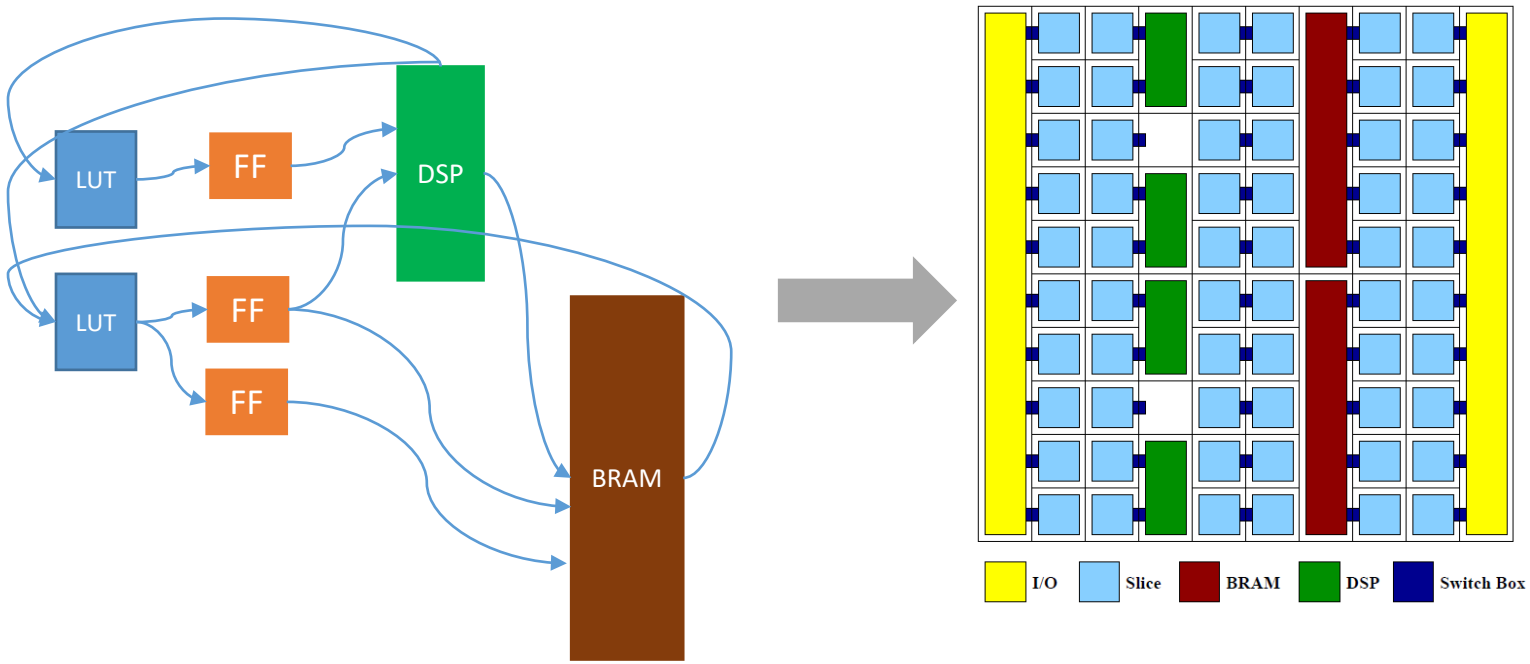
Outline


Outline
FPGA Placement
Congestion Estimation
Routability Prediction
Adaptive Smart Flow
Conclusions & Future work

FPGA Placement Problem

Given a circuit in the form of a netlist, and an FPGA architecture

- Map the components in the netlist onto locations (resources) on the FPGA such that:
  - Optimize Objectives: Minimize → wirelength, delay, congestion, etc.
  - s.t Constraints: based on FPGA Architecture (no overlap, legality control set constraints …)



Netlist **FPGA**

FPGA Analytical Placement

- Prior approaches to placement used Simulated Annealing.
- Recently, more attention has been directed towards analytic placement, which scales better on large problem instances.

- Analytic placement approach

```
1: Convert netlist to graph using Net model
2: Perform pin propagation
3: repeat
4:    solve non-linear equation system
5:    partition solution to enforce legality
      constraints
6: until termination criteria satisfied
```

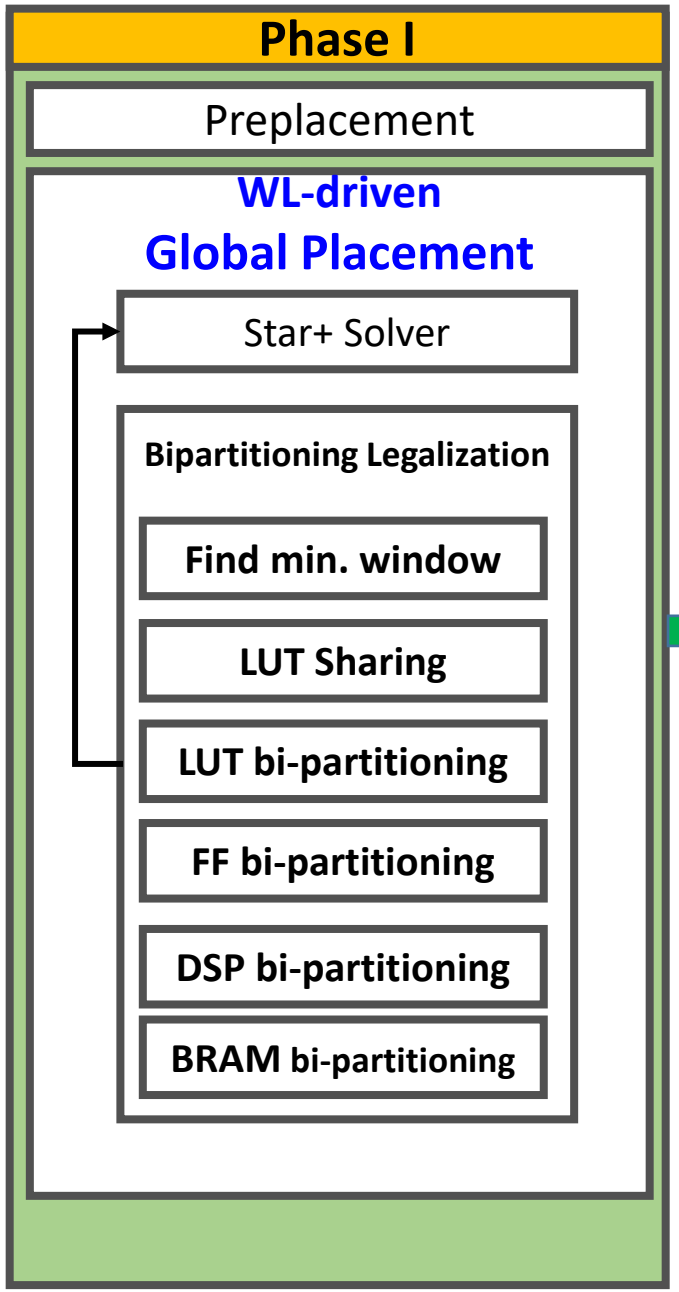
FPGA Analytical Placement

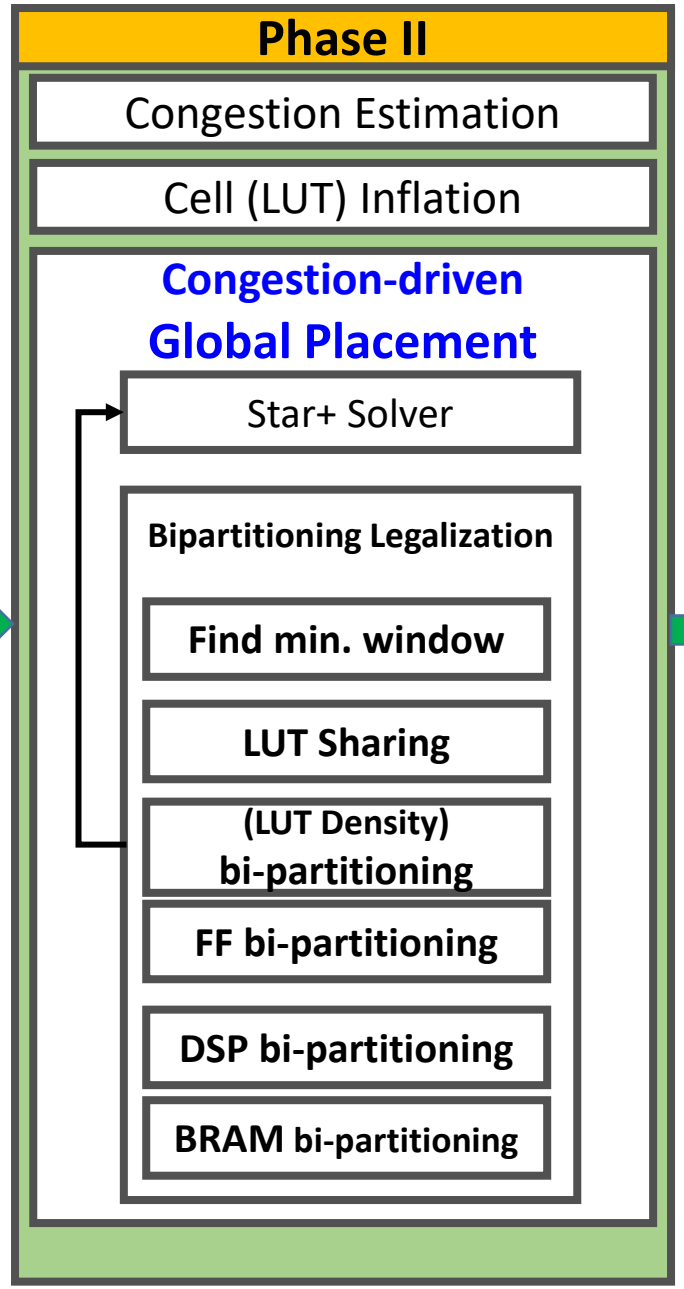
- Prior approaches to placement used Simulated Annealing.
- Recently, more attention has been directed towards analytic placement, which scales better on large problem instances.

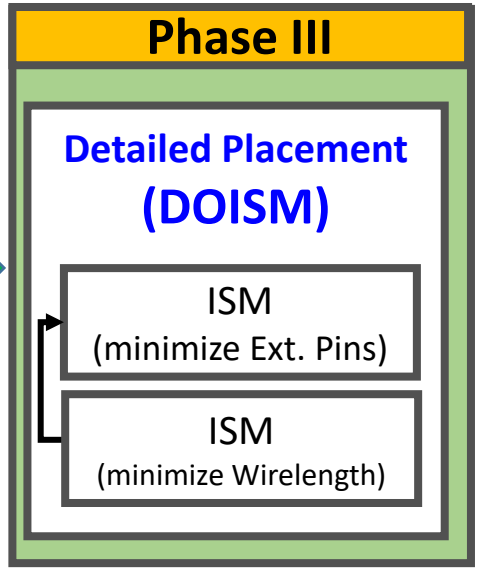
- Analytic placement approach

```
1: Convert netlist to graph using Net model
2: Perform pin propagation
3: repeat
4:    solve non-linear equation system
5:    partition solution to enforce legality
      constraints
6: until termination criteria satisfied
```

GPlace3.0

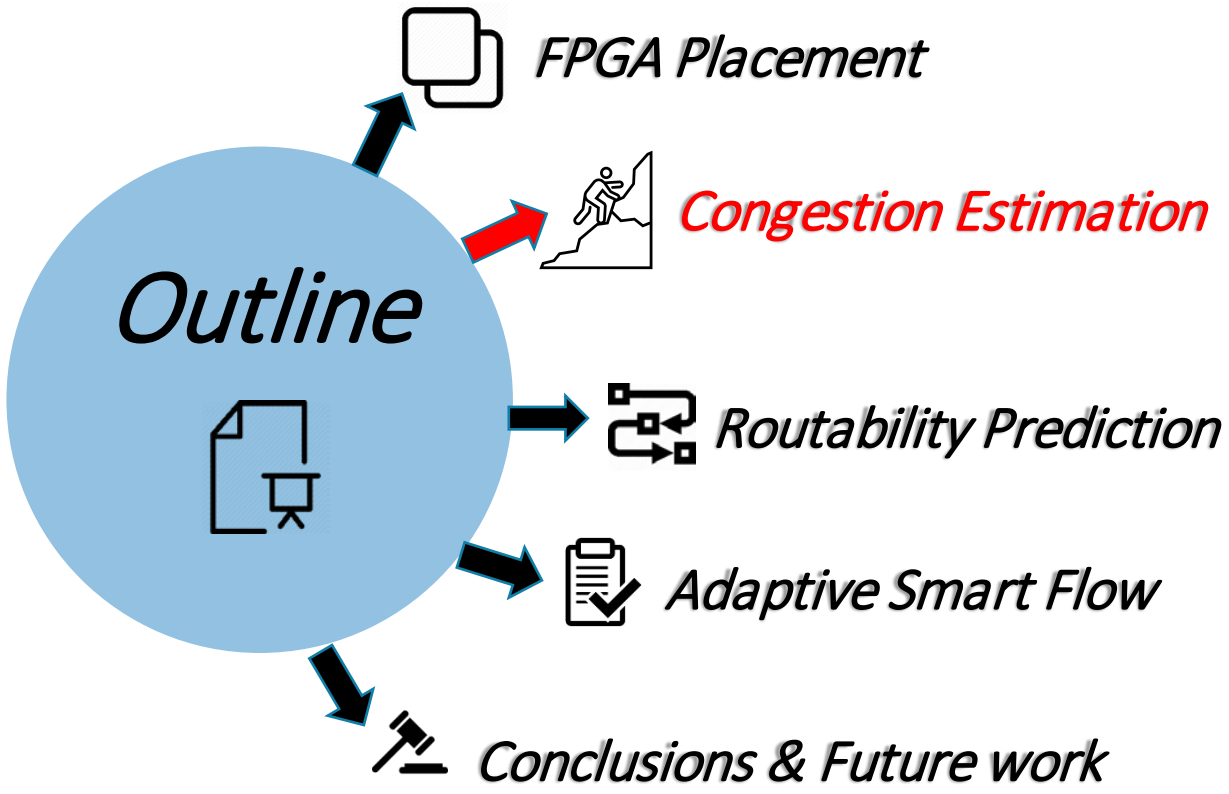






Z. Abuowaimer, D. Maarouf, T. Martin, J. Foxcroft, G. Grewal, S. Areibi, and A. Vannelli, "GPLace3.0: Routability Driven Analytic Placer for UltraScale FPGA Architectures," **ACM Transaction on Design Automation of Electronic Systems**, vol. 23, no. 5, pp. 66:1–66:3, August 2018.

Outline

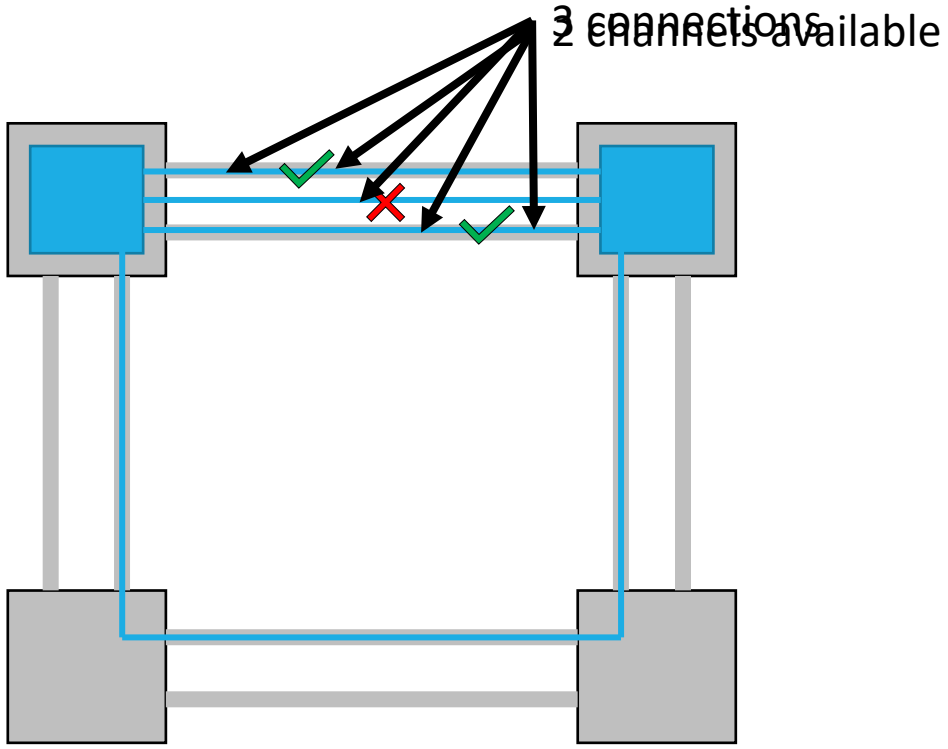

Outline
FPGA Placement
Congestion Estimation
Routability Prediction
Adaptive Smart Flow
Conclusions & Future work

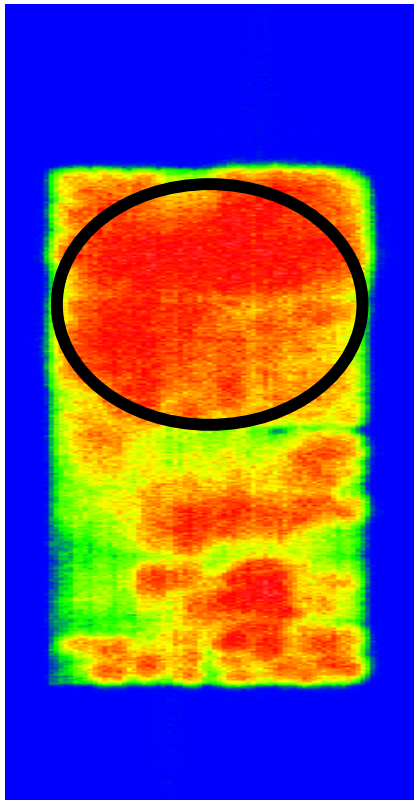
Congestion Estimation: MLCong

D. Maarouf, A. Al-hyari, Z. Abouwaimer, T. Martin, A. Gunter, G. Grewal, S. Areibi, A. Vannelli
"A Machine Learning Based Congestion Estimation for Modern FPGAs",
International Conference on Field Programmable Logic & Applications (FPL 2018), Ireland, pp. 427-434

Congestion

Congestion occurs when the demand for routing resources exceeds the supply in some region of a design



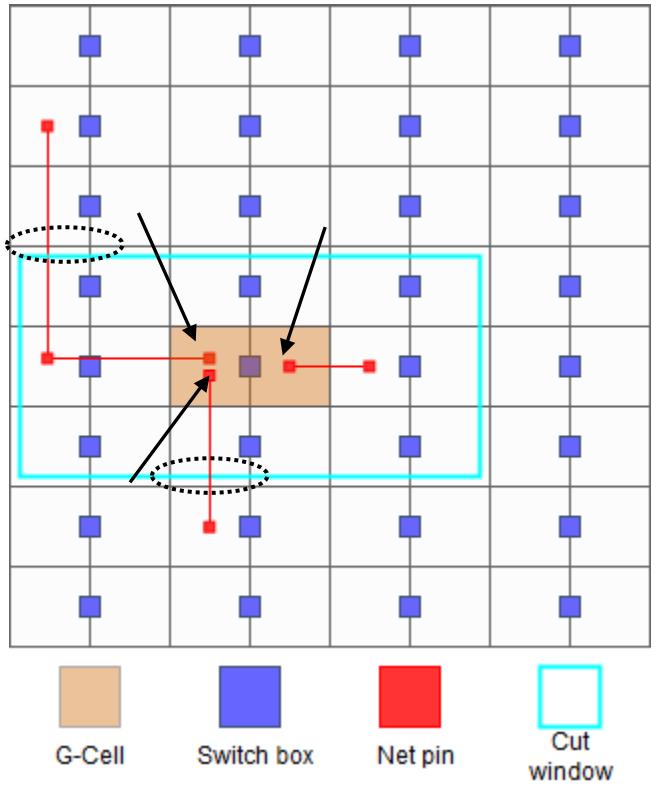


Congestion leads to:

- Placements solutions with excessive wirelength
- Degraded performance of the Router
- Subsequent routing stage may fail

Features

- Four features are calculated for each G-Cell of the FPGA
- Each feature is designed to capture routability information at each switch



$$f_1 = \sum_{n \in N_i} \frac{w_n . HPWL_n}{\#gcell_n}$$

f_1: Wire Length Per Area

$$f_2 = \sum_{n \in N_i} \#pins_{n,gcell_i}$$

f_2: Pin Count (Density)

$$f_3 = | W_{5x5} |$$

f_3: NCPR (5x5)

$$f_4 = | W_{9x9} |$$

f_4: NCPR (9x9)

Benchmarks

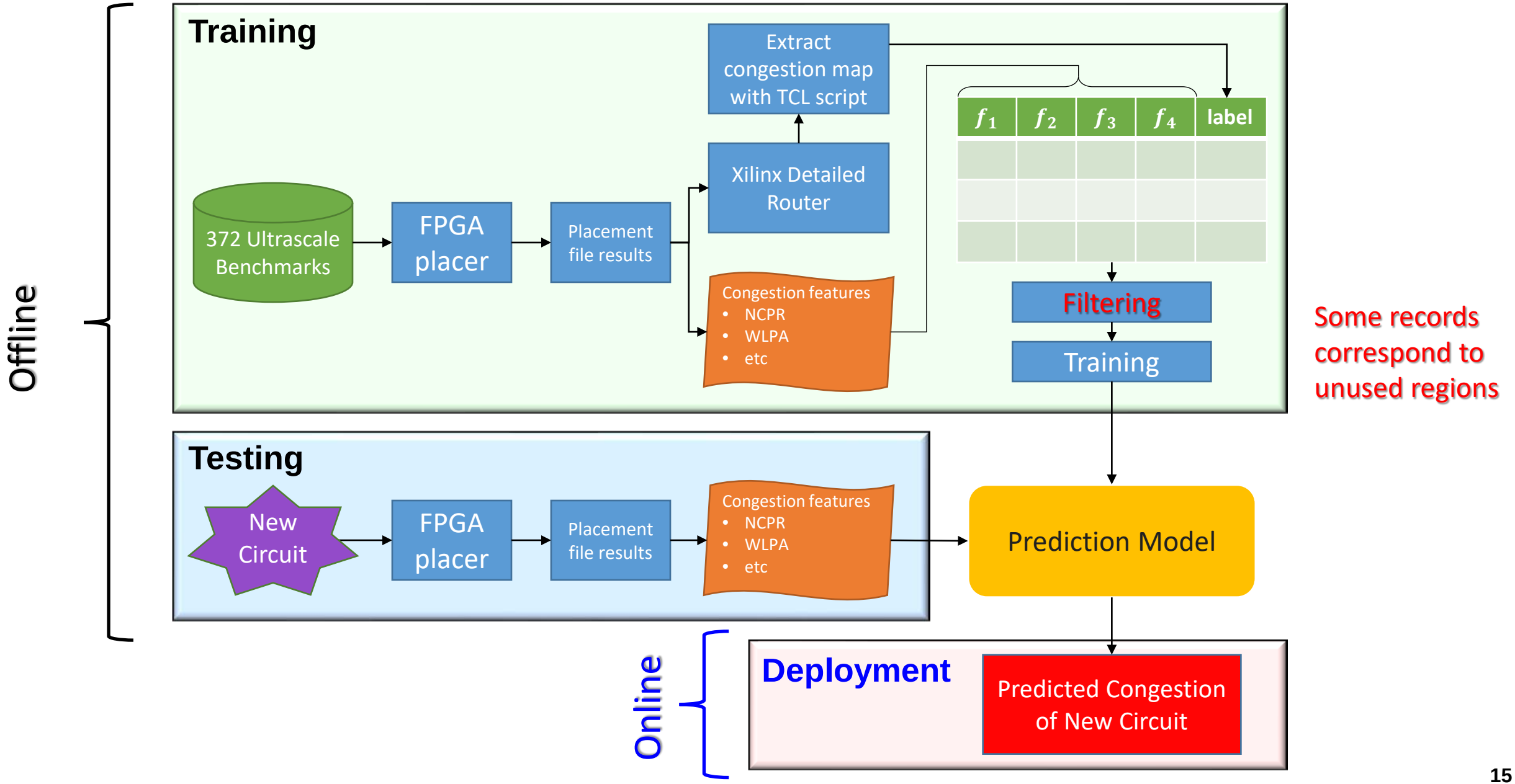
We start with 12 ISPD benchmarks:

Design	#LUTs (util)	#Flops (util)	#RAMB36	#DSPs	#Control Sets	Rent Exponent
FPGA-1	50K (9%)	55K (5%)	0 (0%)	0 (0%)	12	0.4
FPGA-2	100K (19%)	66K (6%)	100 (6%)	100 (13%)	121	0.4
FPGA-3	250K (47%)	170K (16%)	600 (35%)	500 (65%)	1281	0.6
FPGA-4	250K (47%)	172K (16%)	600 (35%)	500 (65%)	1281	0.7
FPGA-5	250K (47%)	174K (16%)	600 (35%)	500 (65%)	1281	0.8
FPGA-6	350K (65%)	352K (33%)	1000 (58%)	600 (78%)	2541	0.6
FPGA-7	350K (65%)	355K (33%)	1000 (58%)	600 (78%)	2541	0.7
FPGA-8	500K (93%)	216K (20%)	600 (35%)	500 (65%)	1281	0.7
FPGA-9	500K (93%)	366K (34%)	1000 (58%)	600 (78%)	2541	0.7
FPGA-10	350K (65%)	600K (56%)	1000 (58%)	600 (78%)	2541	0.6
FPGA-11	480K (89%)	363K (34%)	1000 (58%)	400 (52%)	2091	0.7
FPGA-12	500K (93%)	602K (56%)	600 (35%)	500 (65%)	1281	0.6

#LUTs	#FFs	#BRAMs	#DSPs	#CSETs	#IOs	Rent Exp
44K – 518K	52K – 630K	0 - 1035	0 - 620	11 - 2684	150 - 600	0.4 – 0.8

- We also used 372 benchmarks synthesized using an internal netlist-generation tool based on Generate Netlist (Gnl), and provided by our industrial partner – Xilinx Inc.

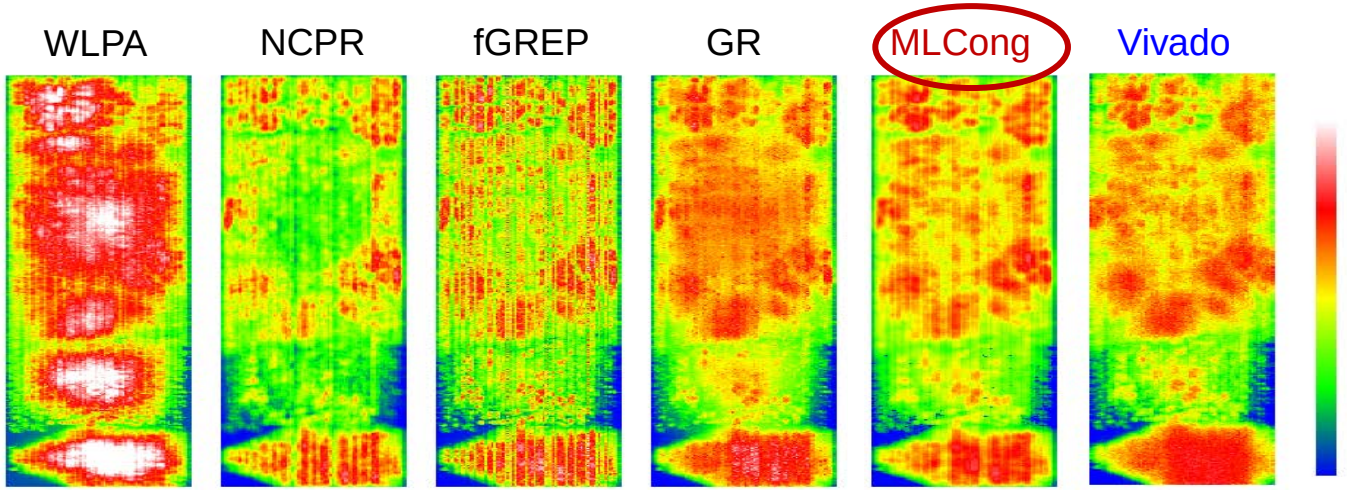
MLCong: An ML Framework for Congestion



Congestion Estimation Techniques: A Comparison

Cong. Estimation Methods	Accuracy Metrics		
	SAD	AANE	RMSE
MLCong	**2891.28**	**6.73**	**5.93**
GR	3126.27	7.34	6.41
fGREP	4351.59	9.66	8.93
NCPR	5254.11	11.47	10.07
WLPA	6185.44	14.20	12.30

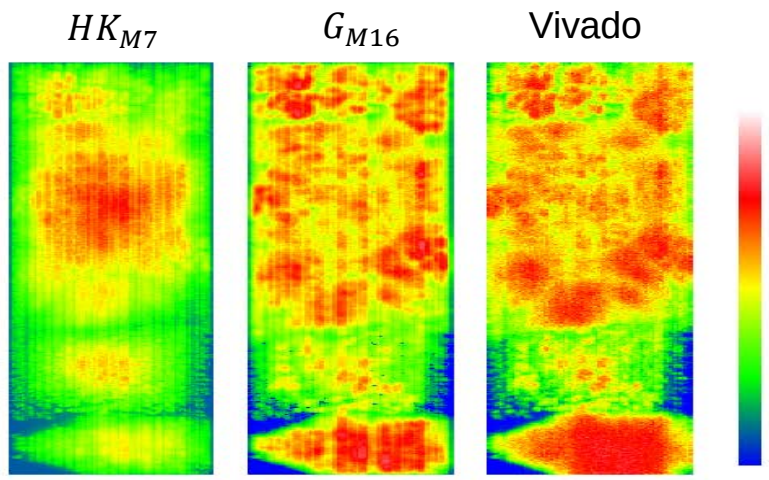
- Sum of Absolute Error (SAD)
- Average Absolute Normalized Error (AANE)
- Root Mean Square Error (RMSE)



MLCong produces congestion heatmaps that are close to those produced by Vivado detailed router

Comparison of Machine Learning Models

Cong. Estimation Methods	Congestion Metrics		Prediction Accuracy Measures	
	SAD	AANE	RMSE	R^2
MLCong	2891.28	6.73	5.93	85.24
HK_{M7} [1]	7983.41	19.23	15.29	**60.39**

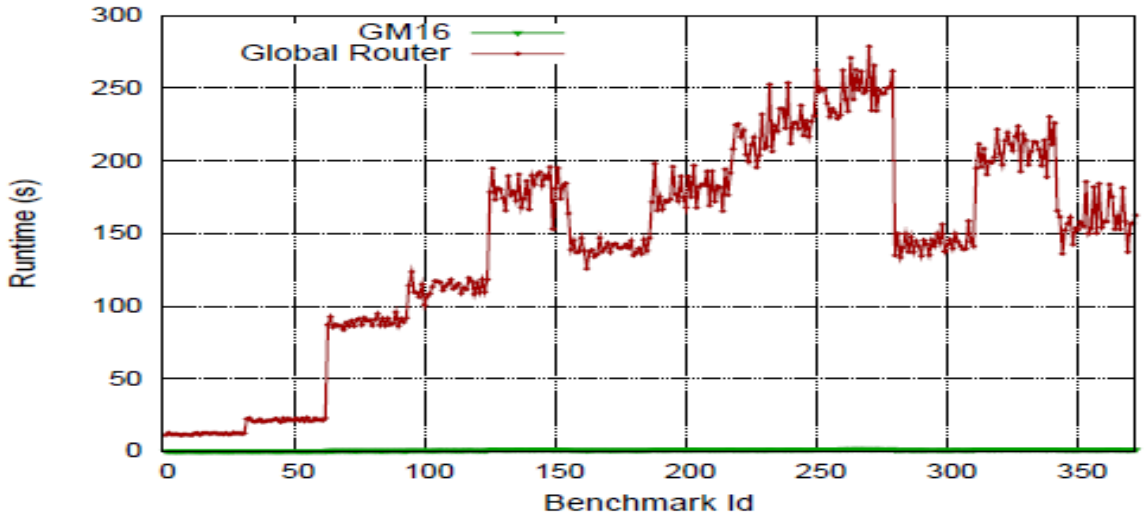


[1] C. Pui, G. Chen, Y. Ma, E. Young, and B. Yu. Clock-Aware UltraScale FPGA Placement with Machine Learning Routability Prediction. In International Conference on Computer Aided Design, pages 929–936. ACM, **2017**.

Case Study

Congestion Estimation Method	# Failures	Routed-WL (Norm.)	Router Runtime (Norm.)	Placer Runtime (Norm.)
MLCong	2	1.00x	1.00x	1.00x
mPFGR	2	1.00x	1.19x	1.17x
No Estimation	225	1.03x	3.15x	0.47x

On Average MLCong is 300x faster than the global router as a standalone congestion estimation technique



Congestion Estimation: DLCong

D. Maarouf, A. Shamli, T. Martin, G. Grewal, S. Areibi

"A Deep Learning Framework for Predicting Congestion during FPGA Placement",

International Conference on Field Programmable Logic & Applications (FPL 2020), Sweden, pp. 138-144

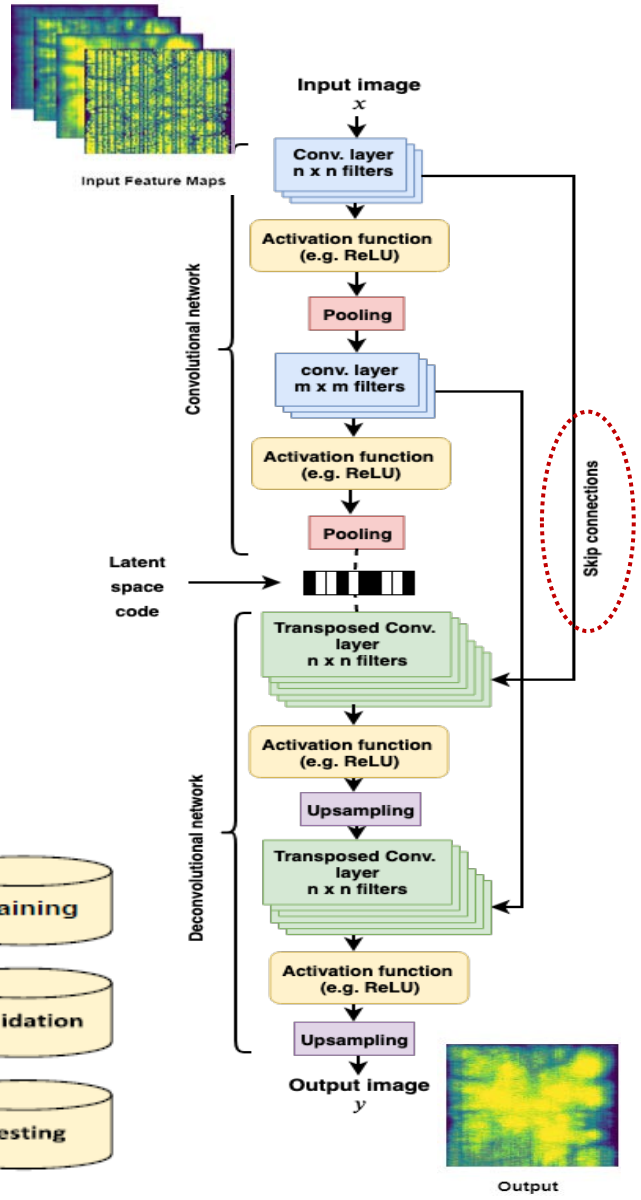
DLCong: Congestion Estimation via DL

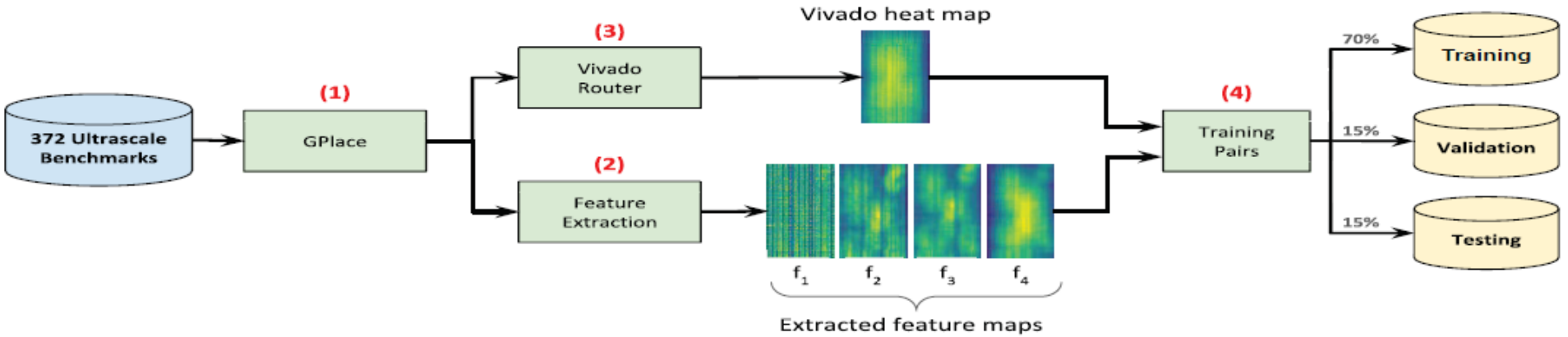
In this work, we claim that:

- The performance of our previous work (MLCong [1]) can be improved by using a deep learning Convolutional Encoder Decoder (CED), that is coined DLCong.
  - ✓ DLCong is capable of capturing global behavior of the detailed router by training on feature maps (i.e., images) rather than local individual switches.
  - ✓ It is also capable of modeling the non-linear relationships between placement features and routing resources on the FPGA.
- DLCong achieves a prediction accuracy of 94%
  - A 9% improvement in accuracy over MLCong.
- Scales well with increasing congestion.
- It's inference time is a few milliseconds.

Deep Learning Encoder-Decoder

- CED consists of five layers in the encoder and the decoder portion.
- The input to the CED are the feature maps.
- The output of the CED is the estimated congestion.
- Convolutional layers capture the spatial relation of a switch with its surrounding switches.





DLCong: Comparison with State of the Art

GAME: Grid Average Mean Absolute Error "determines if shape is accurate"

Unstructured metrics

Method	RMSE	MAE	R^2
WLPA	9.925	7.195	65.06%
NCPR5	9.428	7.527	53.76 %
NCPR9	10.205	8.103	52.61%
fGREP	8.98	6.391	68.39%
MLCong	5.678	4.109	85.55%
DLCong	3.74	2.87	94.33%

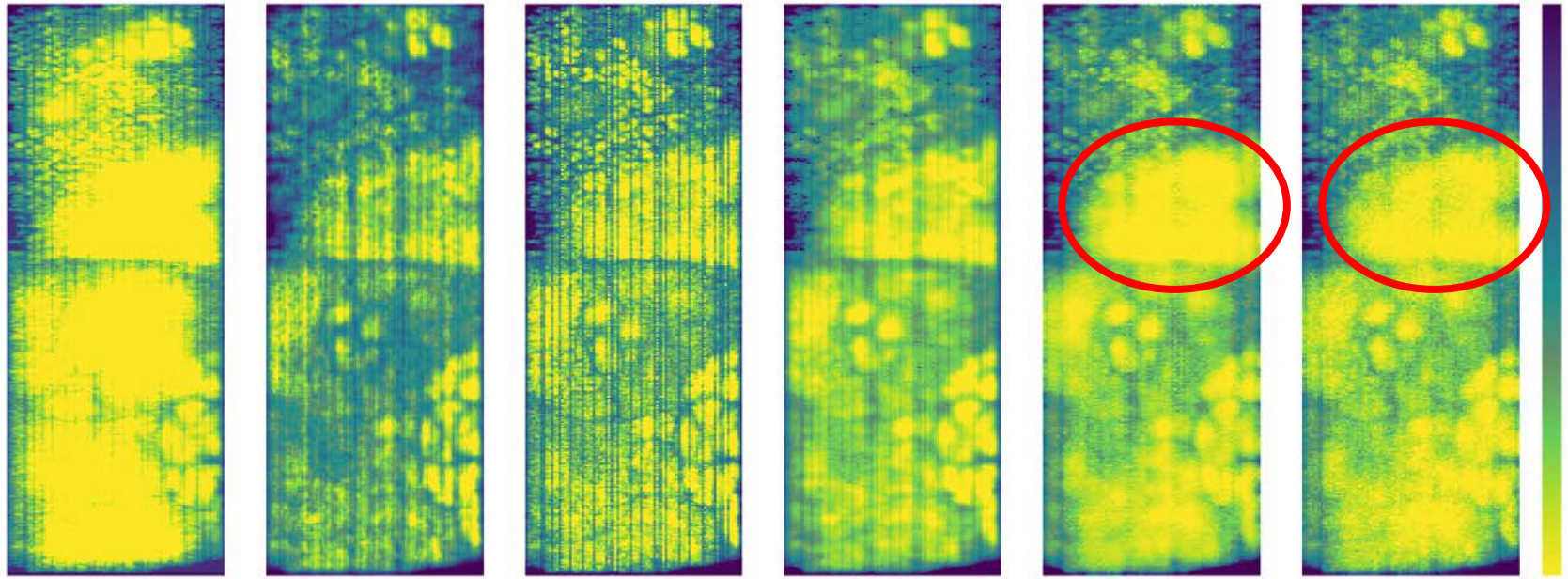
Structured metrics

Method	GAME(3)	GAME(4)
WLPA	7900.8	9242.9
NCPR5	7845.3	8547.9
NCPR9	6062.7	7112
fGREP	6179.7	6855.2
MLCong	4551.13	5117.9
DLCong	252.6	322.7

DLCong improves upon MLCong by 1.42x, 1.51x and 8.78% for RMSE, MAE and R^2

Congestion Estimation: Visual Comparison

- A visual comparison of various congestion estimation methods was used.
- The intensity of congestion in the upper part of the congestion map is underestimated by linear techniques (fGREP), while DLCong is predicting it accurately.

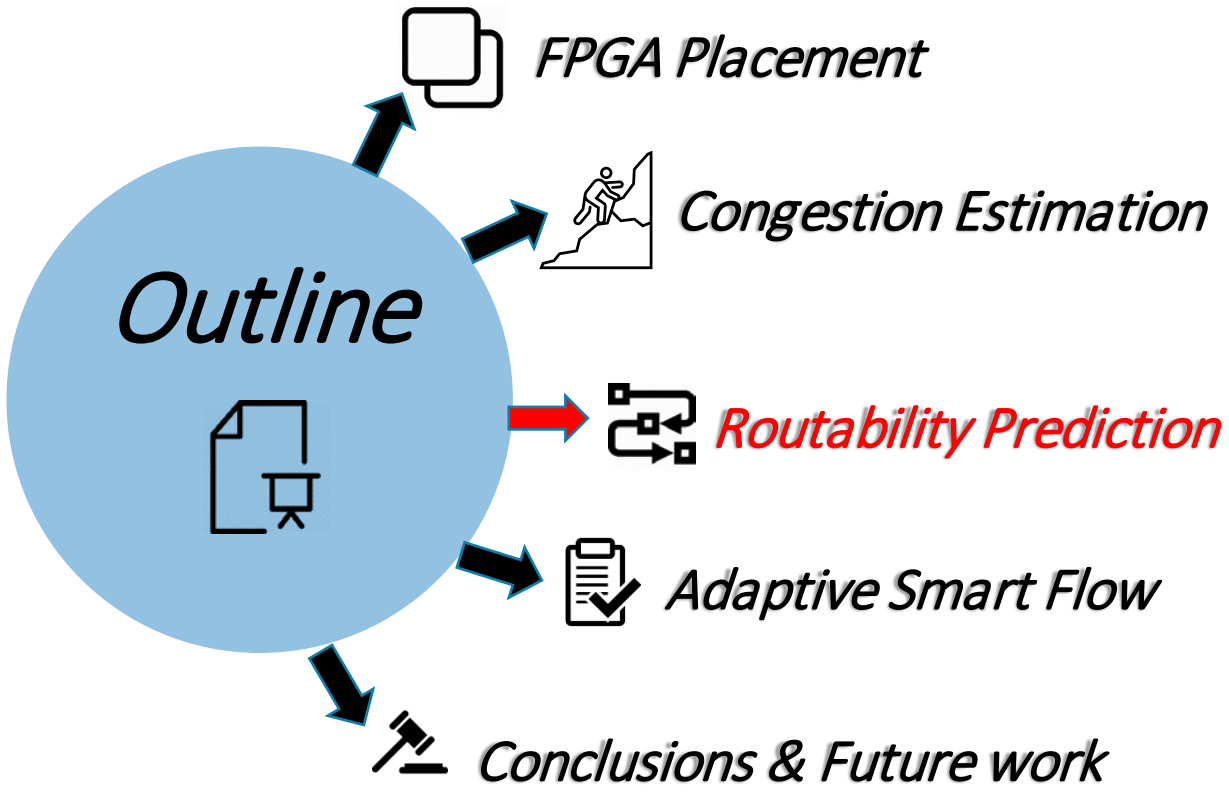


DLCong: Case Study

- DLCong was tested on placement solutions produced by other state-of-the-art academic tools and commercial placement tools other than GPlace
- The results indicate that DLCong can generalize to other placers congestion with an accuracy up to 91.28%

Placer	Routable placement	RMSE	MAE	R^2
Ripple	336	7.3	4.86	82.45%
UTplace	317	5.37	4.03	89.17%
Vivado	262	5.21	3.88	91.28%

Outline

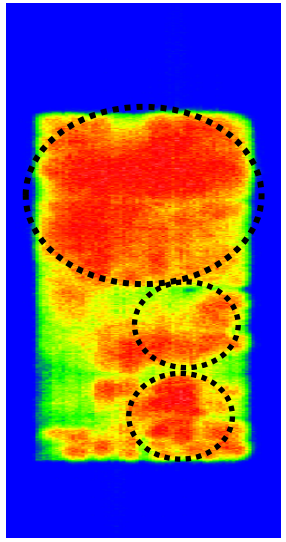

Outline
FPGA Placement
Congestion Estimation
Routability Prediction
Adaptive Smart Flow
Conclusions & Future work

Routability Prediction: DLRoute

A. Alhyari, A. Shamli, Z. Abuwaimer, S. Areibi and G. Grewal,
"A Deep Learning Framework to Predict Routability for FPGA Circuit Placement,"
International Conference on Field Programmable Logic & Applications (FPL 2019), Barcelona, Spain, pp. 334-341

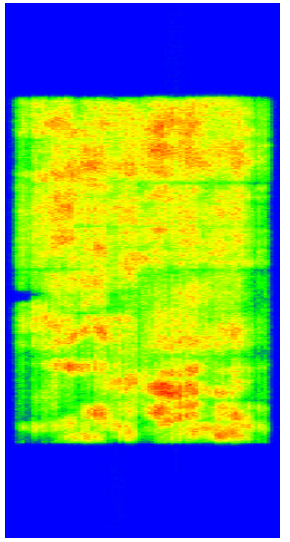
Routability Driven Placement

- In this work we present a novel, deep-learning framework based on Convolutional Neural Networks to accurately predict the routability of a placement.
- Integrating such a predictor in an FPGA placement flow can assist in improving QoR/CPU



Today's largest FPGA designs can easily take hours to place with no guarantee of routing success

It is crucial for the placement tool to know as early as possible whether a design is routable



Unroutable Placement due to Congestion

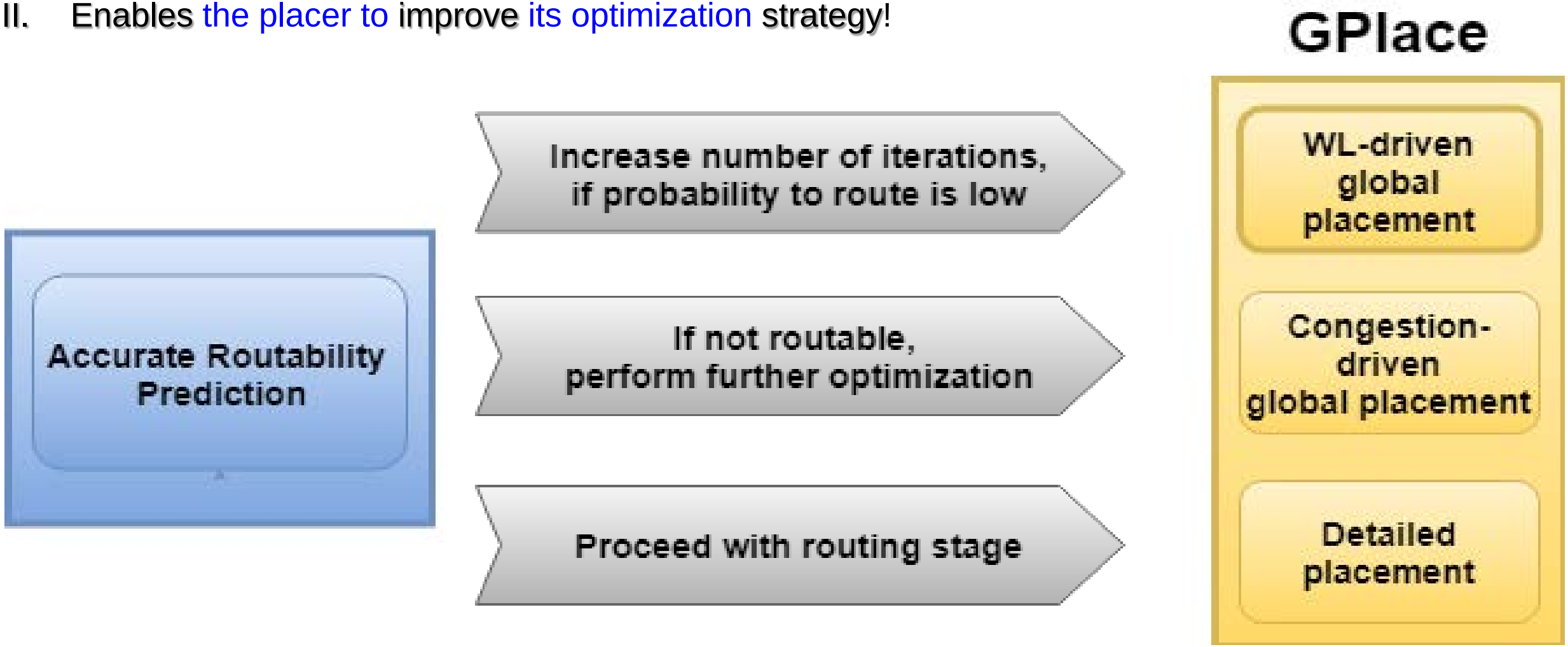
✓ Congestion free, routable placement

Routability Prediction: Benefits

An early estimate of routability can help the placer:

I. Avoid pursuing dead-end paths that do not lead to a feasible routing solution,

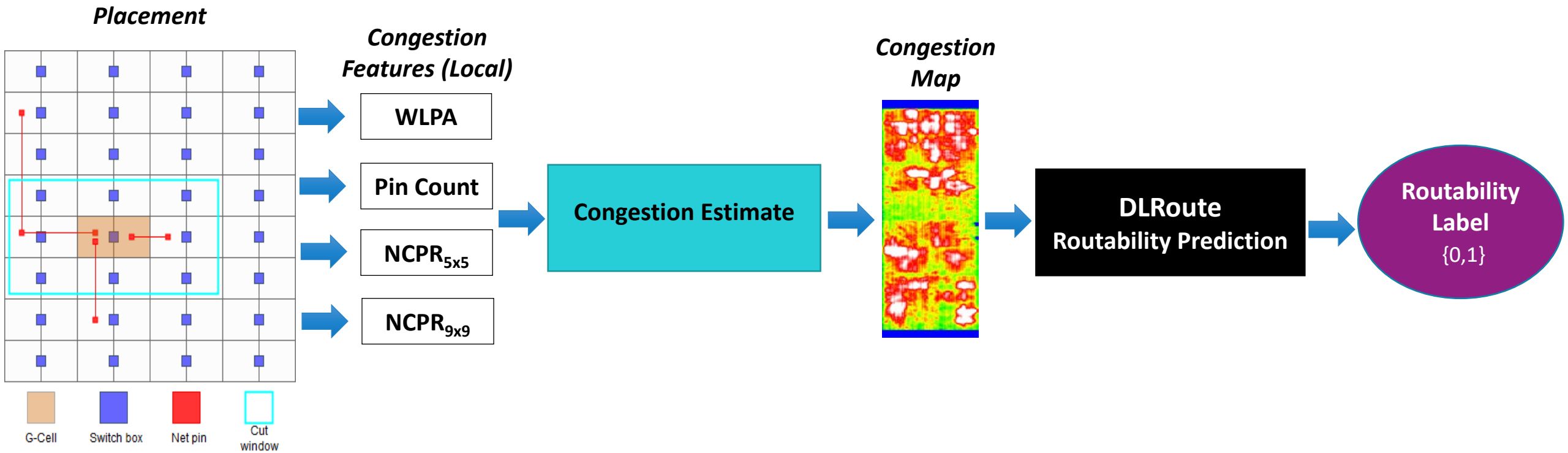
II. Enables the placer to improve its optimization strategy!



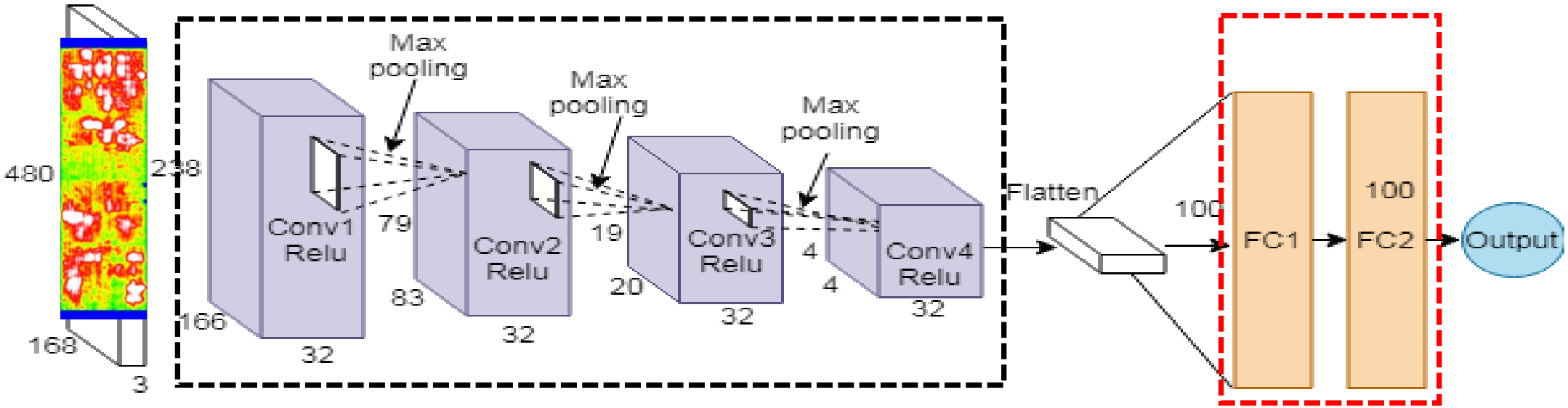
Routing Prediction: DLRoute

The proposed DL model for predicting routability is integrated within Gplace:

- Provides feedback to further improve optimization
- Avoids wasting time running the router on a placement which will fail to route

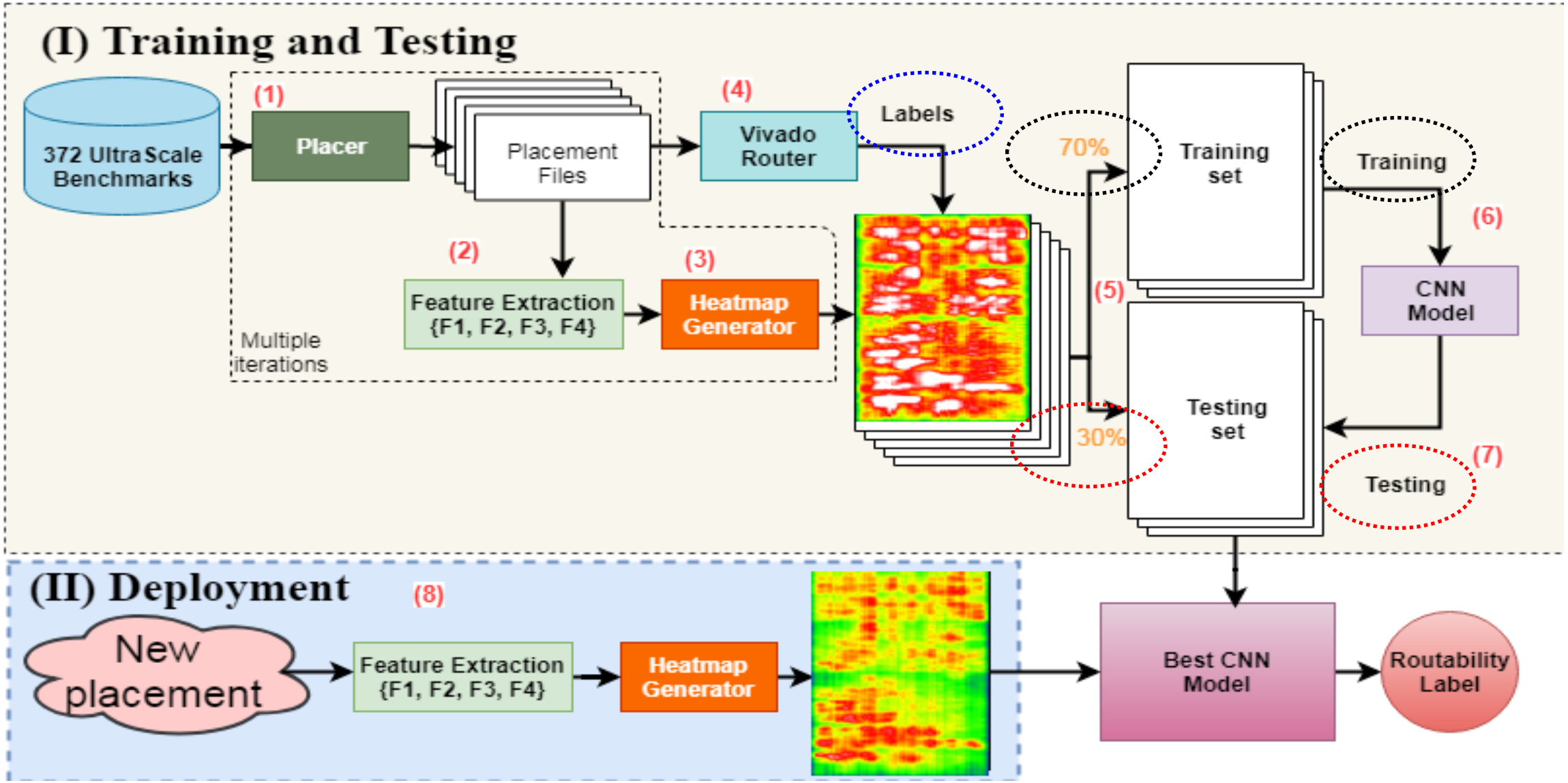


DLRoute: CNN Architecture



- The network takes a congestion heat map of size 480x168 as input
- Four convolutional layers with a depth of 32 filters are used to extract features
- Two fully connected layers are used to classify the flattened vector of features
- A sigmoid output neuron generates a binary label of {0, 1} as routability label

DLRoute Framework



DLRoute: Performance Results

Overall Performance

Accuracy	Precision	Sensitivity	Specificity	M	Train Time	Test Time
97.4%	0.961	0.980	0.970	0.876	115.8 (min)	7.8 (ms)

Performance on Each Placement Phase

Phase	Accuracy	Precision	Sensitivity	Specificity	M
Global (Wirelength) Placement	0.988	0.955	0.993	0.986	0.967
Global(Congestion) Placement	0.958	0.944	0.962	0.954	0.915
Detailed Placement	0.983	0.987	0.995	0.826	0.864

DLRoute Case Study #1: Saving Router Time

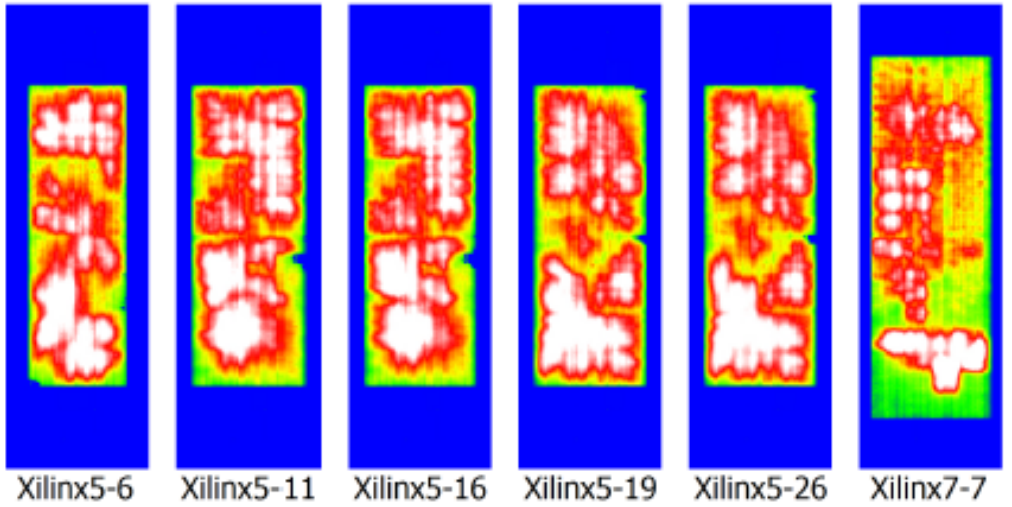
- The proposed routability predictor was tested on Xilinx Vivado placer and several state-of-the-art academic placers.
- Each placer was used to generate placements for the 372 benchmarks and routed.
- Each placer had a success/failure rate to route these benchmarks.
- These placers could have avoided performing routing if DLRoute were used.
- The savings in time ranges from 42.7% to 82.1%

Placer	Routability		CPU Time			Saving
	Routable	Non-Routable	Routed	Unrouted	Total	
UTPlace[8]	317 (85%)	55 (15%)	315473	308239	623712	49.4%
Ripple[9]	336 (90%)	36 (10%)	338626	253180	591806	42.7%
Vivado2015.4	262 (70%)	110 (30%)	209402	964381	1173783	82.1%
Vivado2018.1	327 (88%)	45 (12%)	527227	402704	929931	43.4%

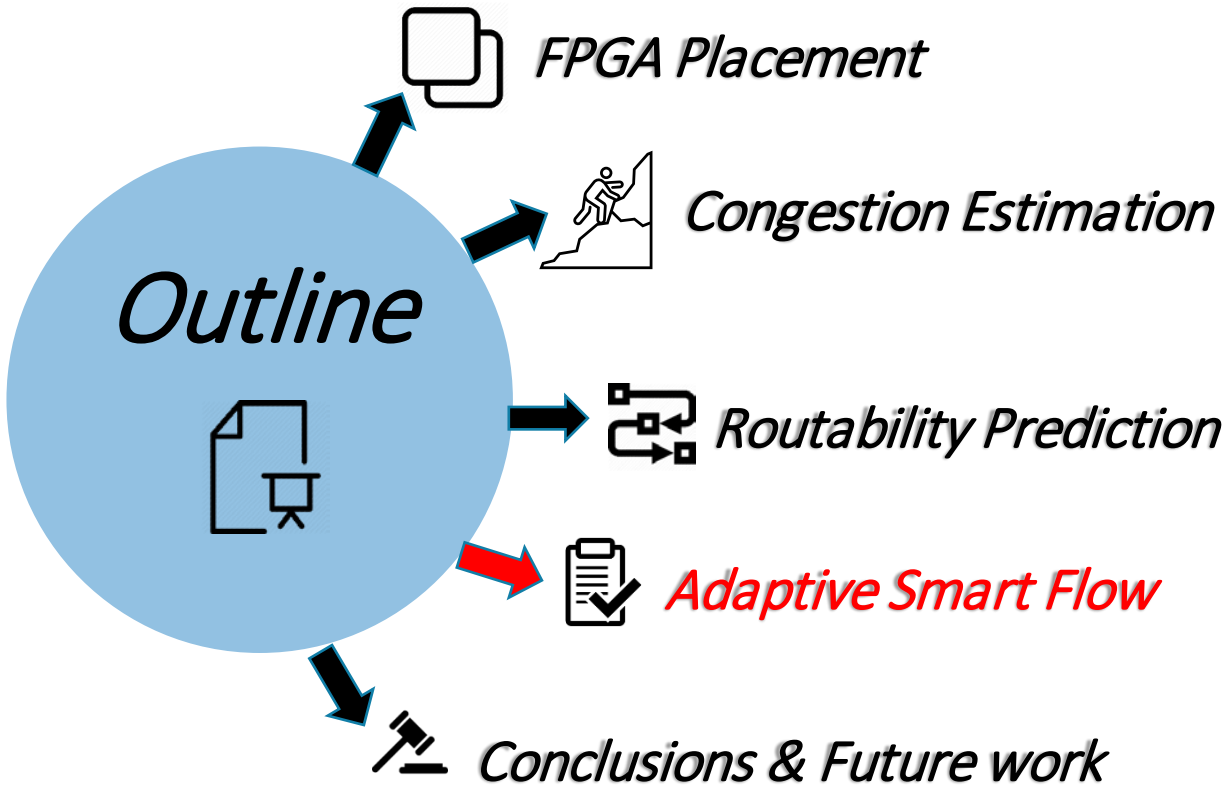
DLRoute Case Study #2: Feedback to the Placer

- Six highly-congested benchmarks were unroutable by GPlace3.0
- Integrating the CNN into a placement tool was used to adaptively improve its optimization strategy.
- To be able to route highly-congested placement, a feedback from the CNN is used to control the cell inflation parameters.
- The six highly-congested benchmarks are now routable.

Benchmark	Routing Results	
	Wirelength	CPU Time (seconds)
FPGA5-6	9900742	2639
FPGA5-11	11814937	6634
FPGA5-16	11858397	5497
FPGA5-19	12069961	4897
FPGA5-26	12035954	3235
FPGA7-7	9540692	3095



Outline


Outline
FPGA Placement
Congestion Estimation
Routability Prediction
Adaptive Smart Flow
Conclusions & Future work

Forecasting Congestion: DLForecast

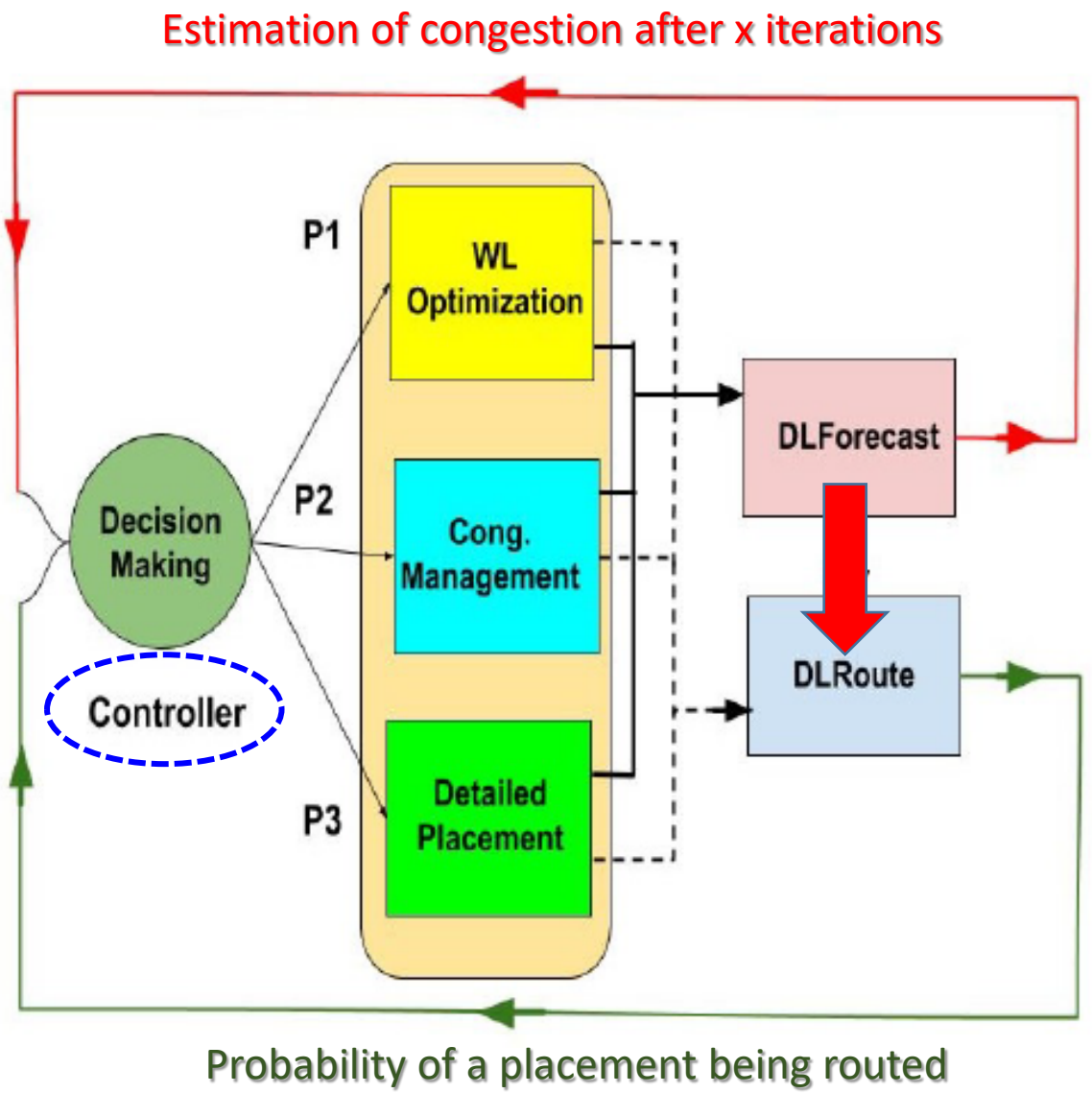
A. Al-hyari, A. Shamli, T. Martin, S. Areibi and G. Grewal
"An Adaptive Analytic FPGA Placement Framework Based on Deep-Learning",
2020, ACM/IEEE Workshop on Machine Learning for CAD (MLCAD 2020), pp. 3–8,Virtual Event, Iceland

DLForecast

- In this work we propose a deep-learning framework to accurately forecast the congestion that will be present at subsequent placement iterations based on congestion features obtained during the early phases of placement.
- We then show how this forecast can be used early in the placement flow to look ahead and make smart optimization decisions with the goal of reducing placement runtimes while maintaining quality of results.

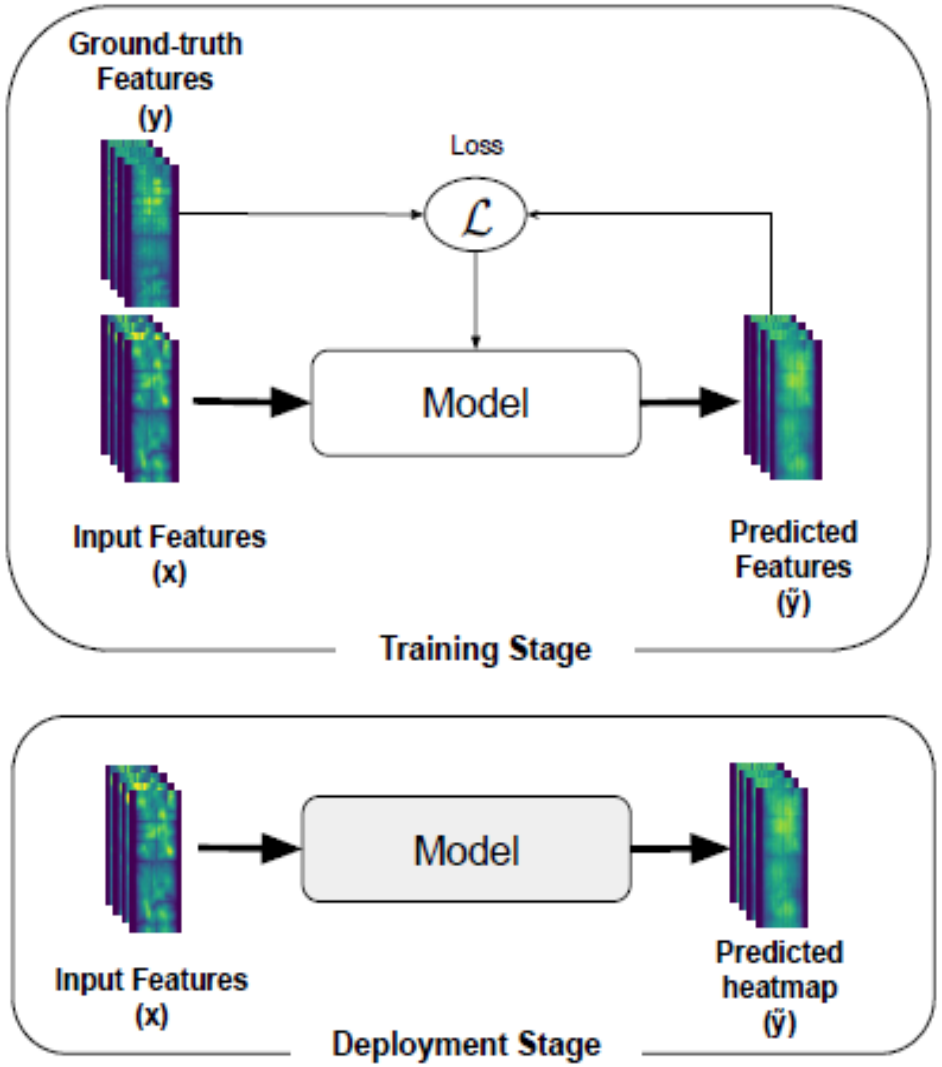
DLForecast Approach: Objective

- DLForecast is used to accurately predict the congestion that could be present at later placement iterations.
- Results are then fed back to the placer and forwarded to DLRoute to obtain the probability of achieving a feasible routing solution.
- A controller within GPlace can then use the routing probability and congestion forecast to decide between several alternative courses of action.



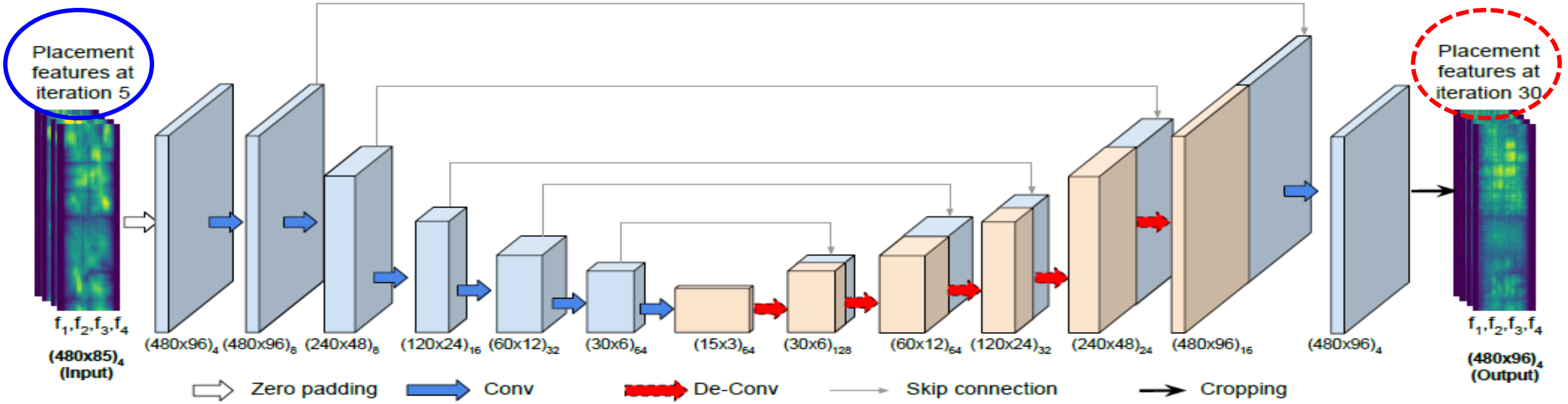
DLForecast Framework

- The input to the model is:
  I. A set of concatenated placement feature maps (images) each describing a particular feature present in the current placement.
  II. Ground-truth features used as a label.
- The Model is used to forecast the congestion feature maps of subsequent iterations including the final iteration of Phase I
- Once trained it is deployed and integrated within a placement tool.



DLForecast Architecture

- The input of the network is the current placement feature maps (size 480x85).
- The output of the network is the predicted congestion at later iterations.



DLForecast Accuracy Metrics

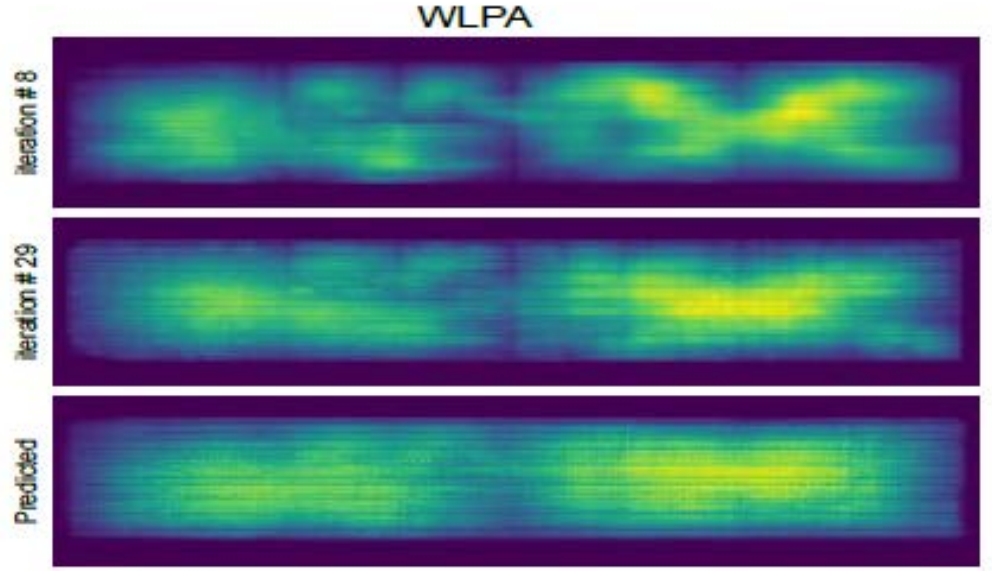
We employ two most commonly used metrics for quantifying prediction accuracy:

- Mean Absolute Error (MAE)
- Coefficient of Determination (R^2)

Metric	MAE	R^2	SSIM
Test Result	0.026	93.5%	0.898

DLForecast Accuracy Metrics

- We also employ the Structural Similarity Index Metric (SSIM)
  - SSIM is used to measure the similarity between images.
  - Unlike MAE and R^2, SSIM accounts for luminance distortion, contrast distortion, and loss of correlation.
- The DLForecast predicted WLPA feature map for FPGA5. The evaluation metrics for this feature are:
  - MAE = 0.025,
  - R^2 = 94.01% and
  - SSIM = 0.902



Runtime/QoR(Wirelength) Comparison

- Columns 2 and 3 compare the runtimes of GPLace3.0 with that of DLForecast respectively.
- Column 4 shows that GPLace3.0 with DLForecast achieves runtime improvements in the range of 27% to 40%.
- Column 5 shows the percentage increase (+) and decrease (-) in wirelength of GPLace3.0 with DLForecast compared to GPLace3.0.
- It is clear that the increase/decrease in wirelength are small, with no overall difference in wirelength across the 12 benchmarks.

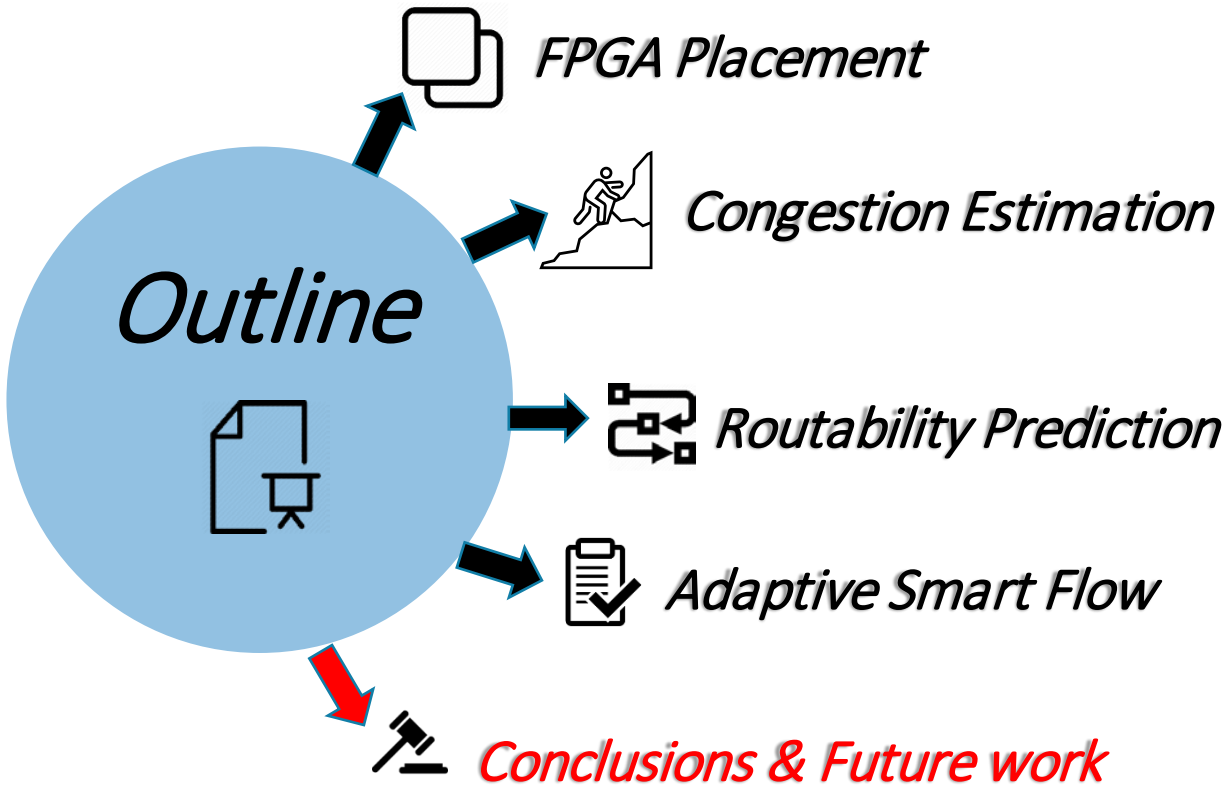
Benchmark	$T_{GPlace3.0}$	$T_{GPl+DLF or}$	% Imp.	% Wirelength
FPGA-1	76	**55**	27.98%	+2.02%
FPGA-2	134	**88**	34.10%	+0.01%
FPGA-3	424	**270**	36.35%	-0.27%
FPGA-4	432	**278**	35.70%	+1.04%
FPGA-5	508	**293**	42.30%	+2.29%
FPGA-6	901	**543**	39.72%	-2.22%
FPGA-7	948	**576**	39.22%	-0.88%
FPGA-8	991	**592**	40.30%	+0.01%
FPGA-9	1277	**768**	39.83%	-2.49%
FPGA-10	1423	**852**	40.11%	+0.98%
FPGA-11	1229	**736**	40.16%	+0.58%
FPGA-12	1767	**1043**	40.96%	-1.46%
Total	10111	**6095**	**39.72%**	-0.03%

State-of-the-art Placers: Comparison

- We next show that the final routed wirelength obtained by integrating GPlace3.0 with DLForecast is comparable to that obtained by other state-of-the-art analytic placement tools.
- Column 3 shows that GPlace3.0 with DLForecast obtains a small 1.40% overall improvement in wirelength compared to RippleFPGA.
- Column 5 shows that GPlace3.0 with DLForecast obtains an even larger 2.98% overall improvement in wirelength compared to UTPlace.

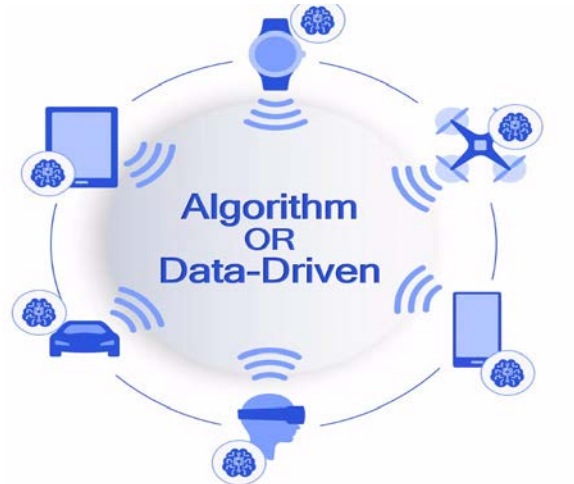
Benchmark	RippleF [3]	% Imp.	UTPlcae [10]	%Imp.	DLFor WL
FPGA-1	**352628**	-5.07%	356769	-3.85%	370503
FPGA-2	645400	-0.59%	**642108**	-1.11%	649238
FPGA-3	3262106	+3.34%	3215087	+1.93%	**3153013**
FPGA-4	5509661	+1.88%	5409765	+0.07%	**5406070**
FPGA-5	9968955	-3.85%	**9659958**	-7.17%	10352730
FPGA-6	6180104	+3.61%	6487628	+8.18%	**5956791**
FPGA-7	**9639639**	-1.39%	10104837	+3.28%	9773651
FPGA-8	8156951	+0.10%	**7879022**	-3.43%	8149034
FPGA-9	12305192	+2.62%	12369055	+3.12%	**11982531**
FPGA-10	7139694	+3.84%	8794515	+21.94%	**6865298**
FPGA-11	11022815	+6.81%	**10196038**	-0.75%	10272573
FPGA-12	**7363451**	-1.47%	7755443	+3.66%	7471911
Total	81546596	**+1.40%**	82870225	**+2.98%**	**80403343**

Outline


Outline
FPGA Placement
Congestion Estimation
Routability Prediction
Adaptive Smart Flow
Conclusions & Future work

Conclusions & Future Work

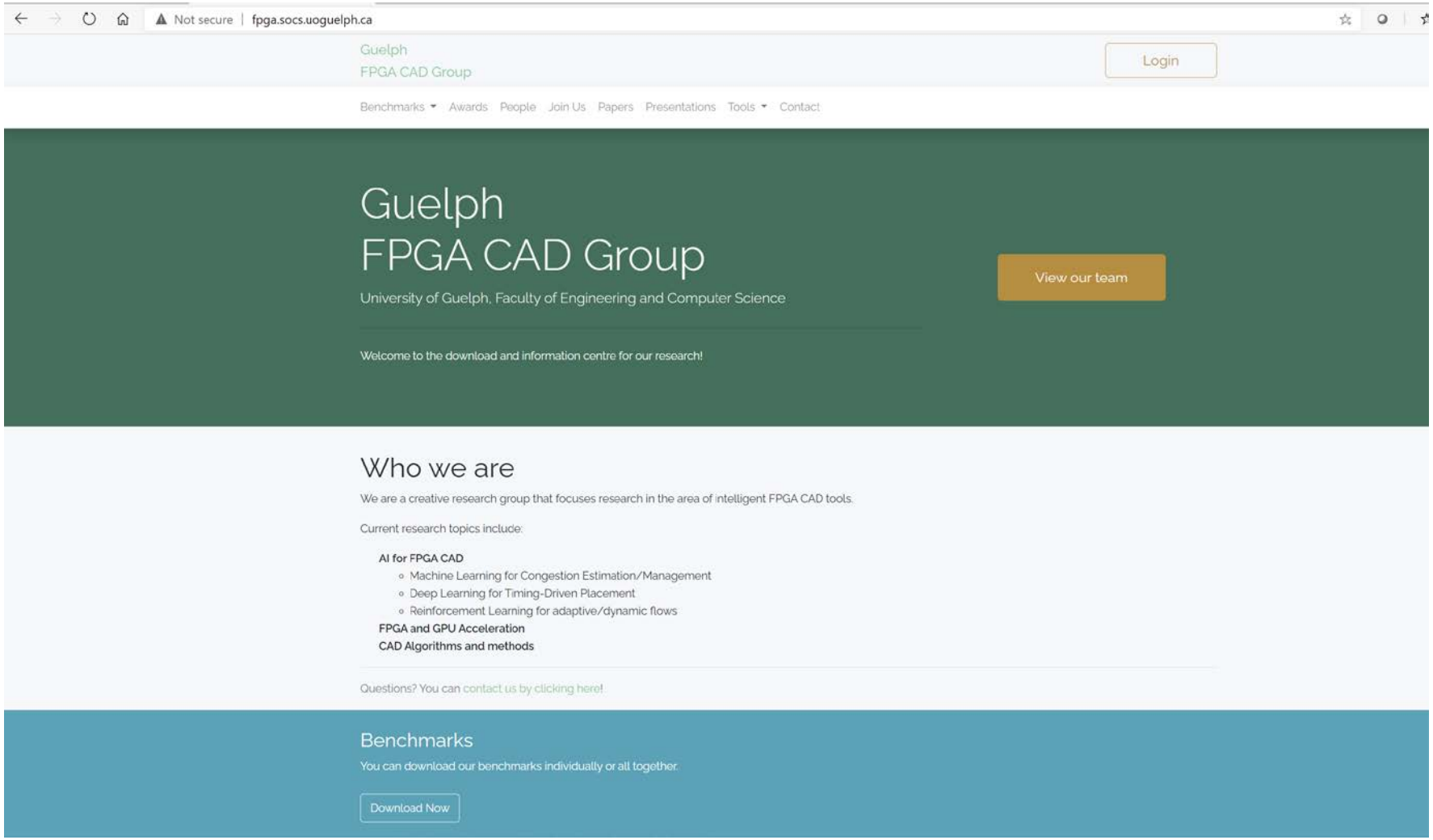
- Algorithmic CAD based on analytic solutions need human guidance
- Data-Driven ML/DL CAD can aid designers with fast QoR evaluation and guide algorithmic CAD with optimal inputs
- Machine Learning can further assist EDA in several directions
  - Adaptive Hyperparameter tuning
  - Guidance to designer to choose best options
  - Enhance productivity of optimization techniques



Thank You

Guelph FPGA CAD Group
Website: https://fpga.socs.uoguelph.ca
Email: sareibi@uoguelph.ca

Our Website



We are currently adding qualified graduate students to our team. Contact us if interested.