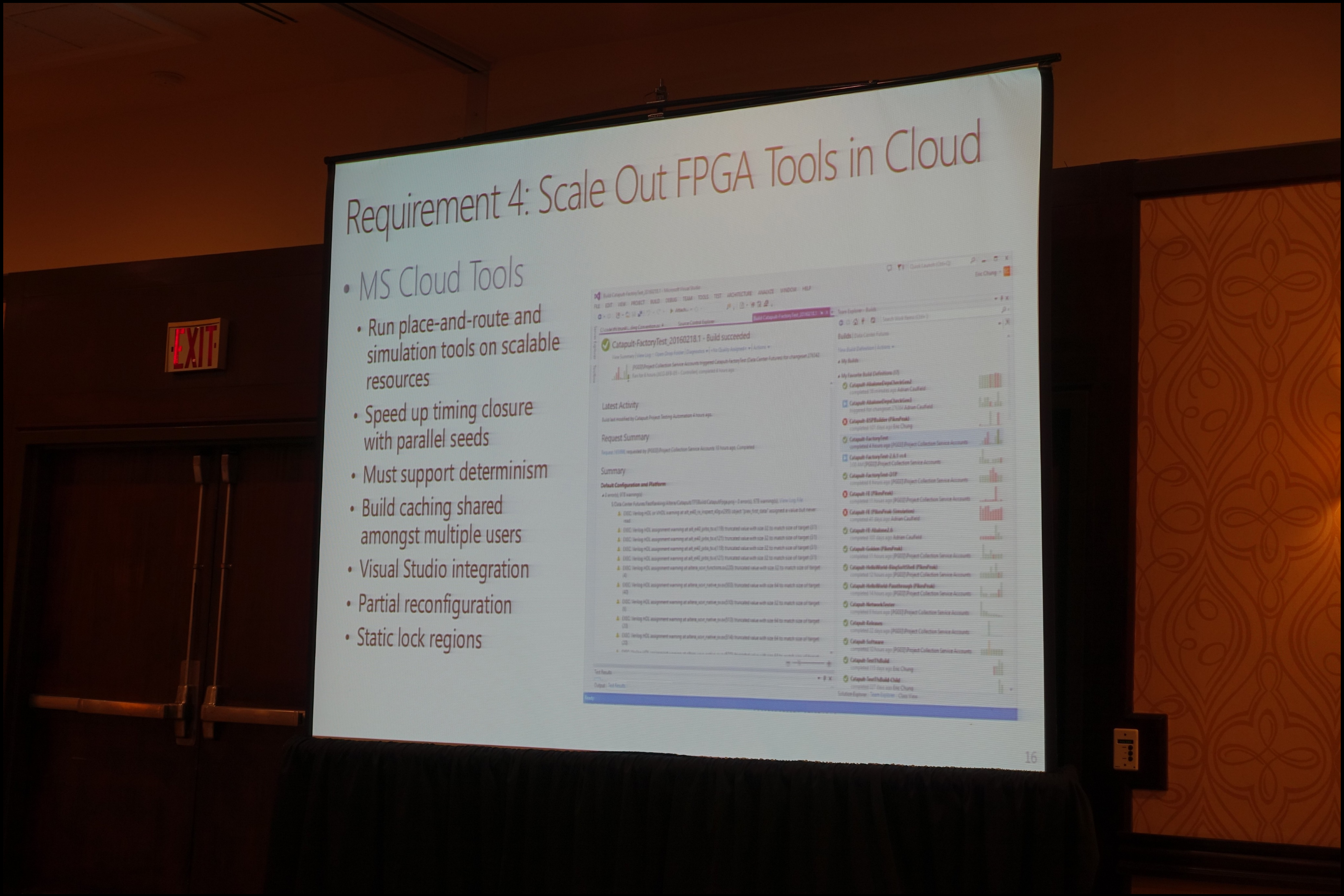
Case for Design-Specific Machine Learning in Timing Closure of FPGA Designs

Que Yanghua, Chinnakkannu Adaikkal Raj,
Harnhua Ng, Kirvy Teo, Nachiket Kapre
nachiket@ieee.org + nachiket@plunify.com

InTime

Problem

- **Problem**: Timing Closure of FPGA designs hard. FPGA CAD tools difficult to control.
- **Opportunity**: Timing influenced by FPGA CAD parameters.
- **Solution**: Machine Learning + Cloud Computing.
- Design-Specific customisation of FPGA CAD parameter selection >> Generic selection
— Not sufficient to simply use machine learning
— Tailor the learning procedure to each design,

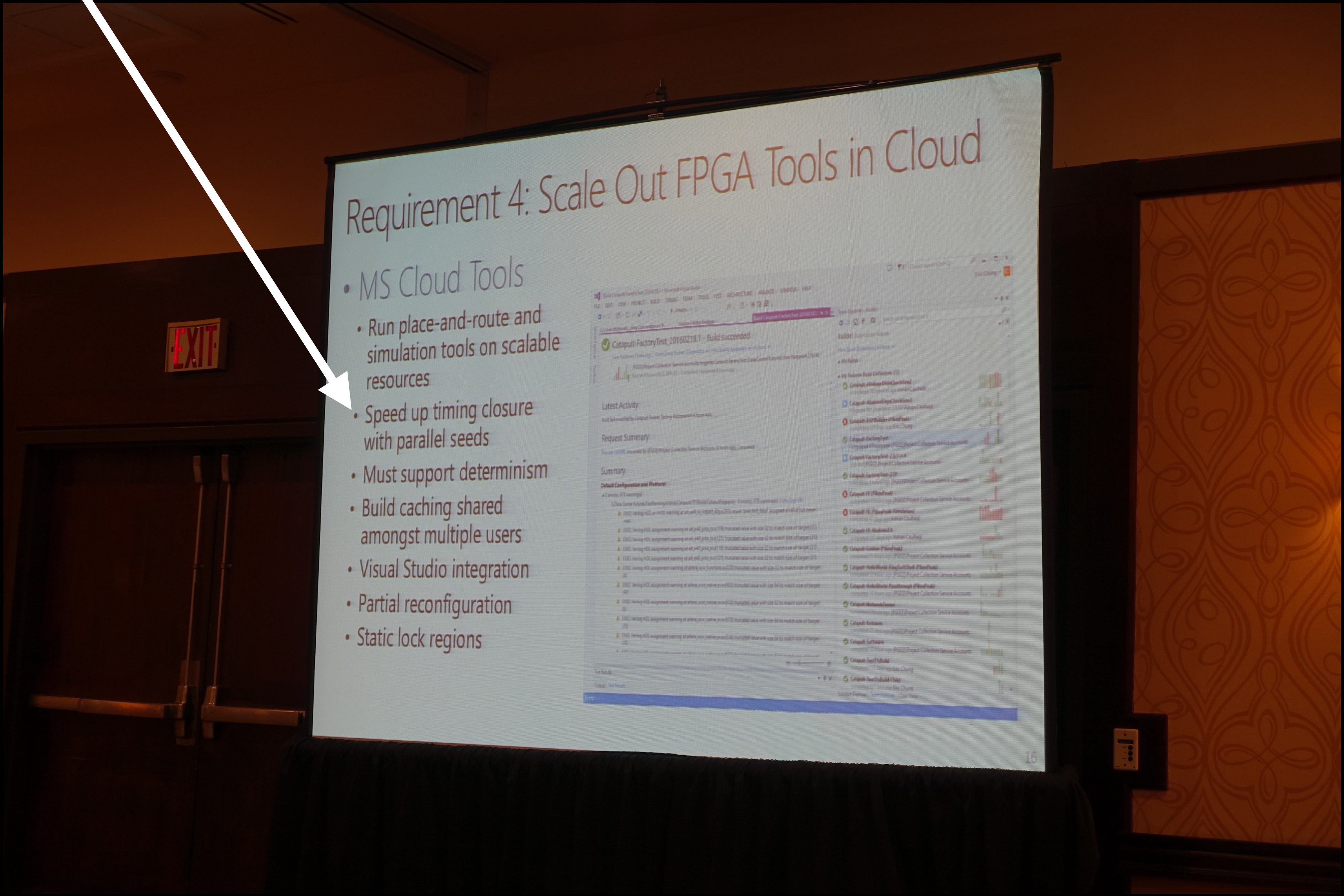


Requirement 4: Scale Out FPGA Tools in Cloud

- MS Cloud Tools
  - Run place-and-route and simulation tools on scalable resources
  - Speed up timing closure with parallel seeds
  - Must support determinism
  - Build caching shared amongst multiple users
  - Visual Studio integration
  - Partial reconfiguration
  - Static lock regions

Eric Chung (Microsoft) — "Agile Co-design for a Reconfigurable Datacenter"
FPGA 2016 Designer's Day (yesterday!)

“Speed up Timing Closure with parallel seeds”



Requirement 4: Scale Out FPGA Tools in Cloud

- MS Cloud Tools
  - Run place-and-route and simulation tools on scalable resources
  - Speed up timing closure with parallel seeds
  - Must support determinism
  - Build caching shared amongst multiple users
  - Visual Studio integration
  - Partial reconfiguration
  - Static lock regions

Catapult-FactoryTest_20160218.1 - Build succeeded

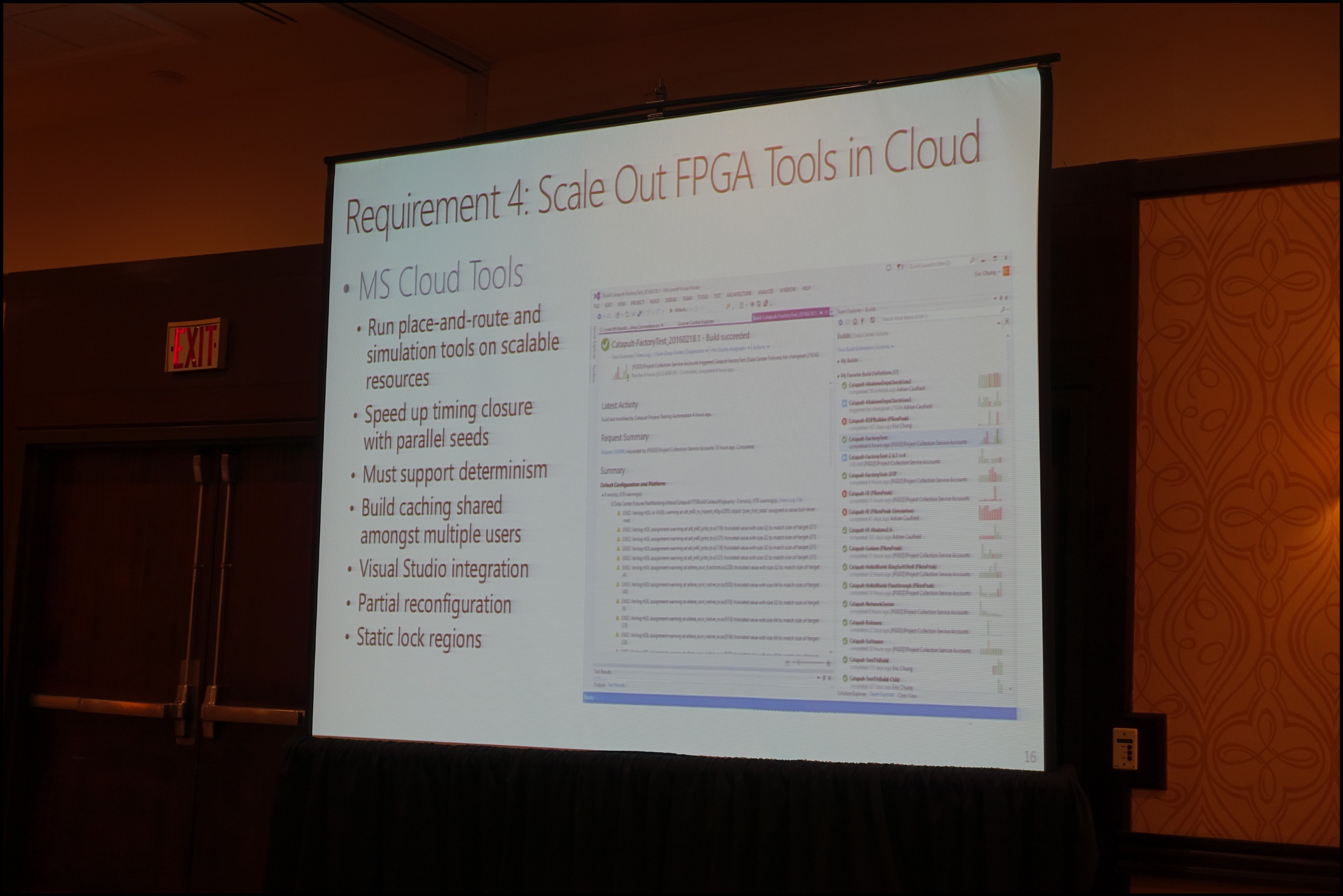
Latest Activity

Request Summary

Summary

Eric Chung (Microsoft) — “Agile Co-design for a Reconfigurable Datacenter”
FPGA 2016 Designer’s Day (yesterday!)

We do parallel seeds + machine learning!



Requirement 4: Scale Out FPGA Tools in Cloud

- MS Cloud Tools
  - Run place-and-route and simulation tools on scalable resources
  - Speed up timing closure with parallel seeds
  - Must support determinism
  - Build caching shared amongst multiple users
  - Visual Studio integration
  - Partial reconfiguration
  - Static lock regions

Catapult-FactoryTest_20160218.1 - Build succeeded

Latest Activity

Request Summary

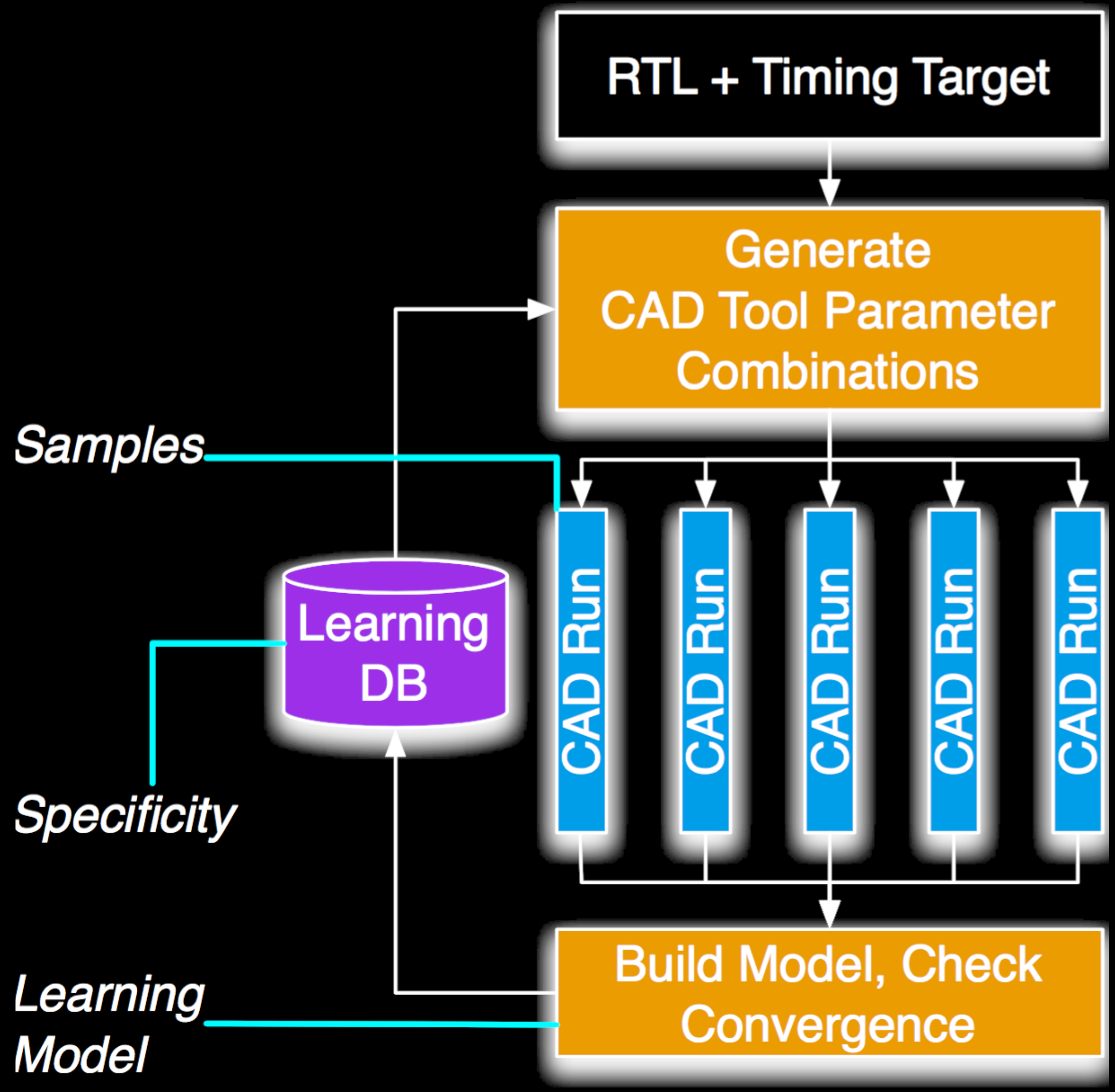
Summary

16

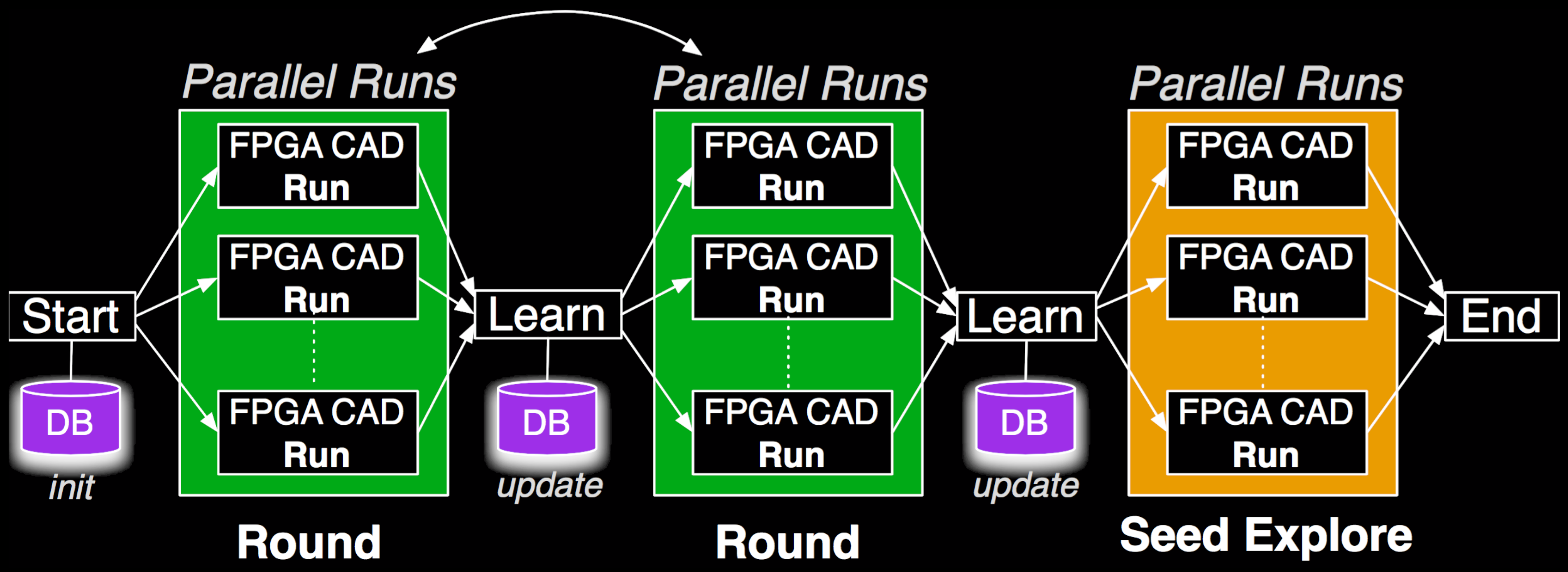
Eric Chung (Microsoft) — “Agile Co-design for a Reconfigurable Datacenter”
FPGA 2016 Designer’s Day (yesterday!)

High-Level View

- Idea:
  — Run multiple CAD tool instances in parallel
  — Build models of timing, devise strategies
- Machine Learning
  — Each CAD run == data point
  — Gather data
- Run machine-learning routines to revise models



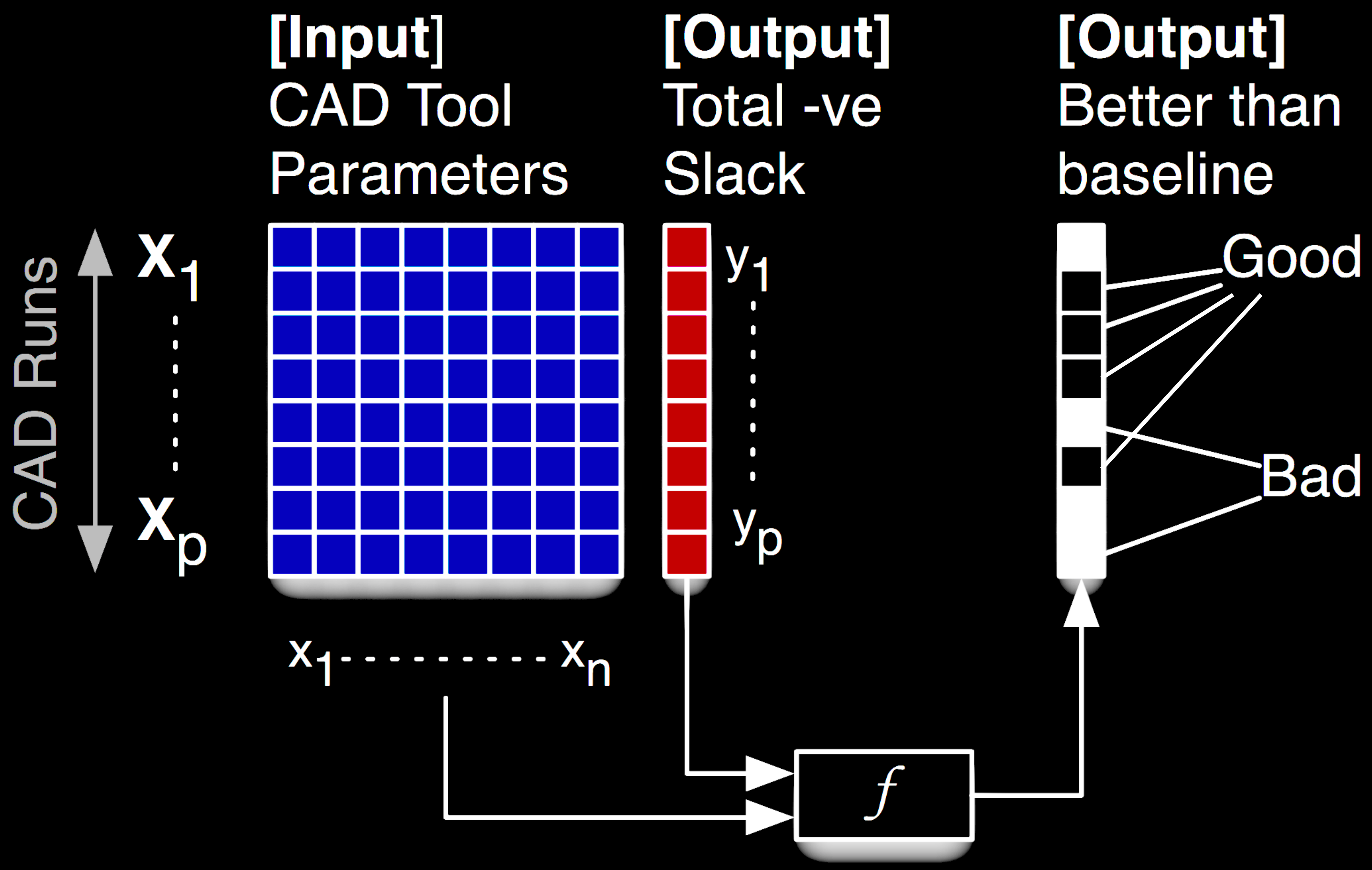
Flow



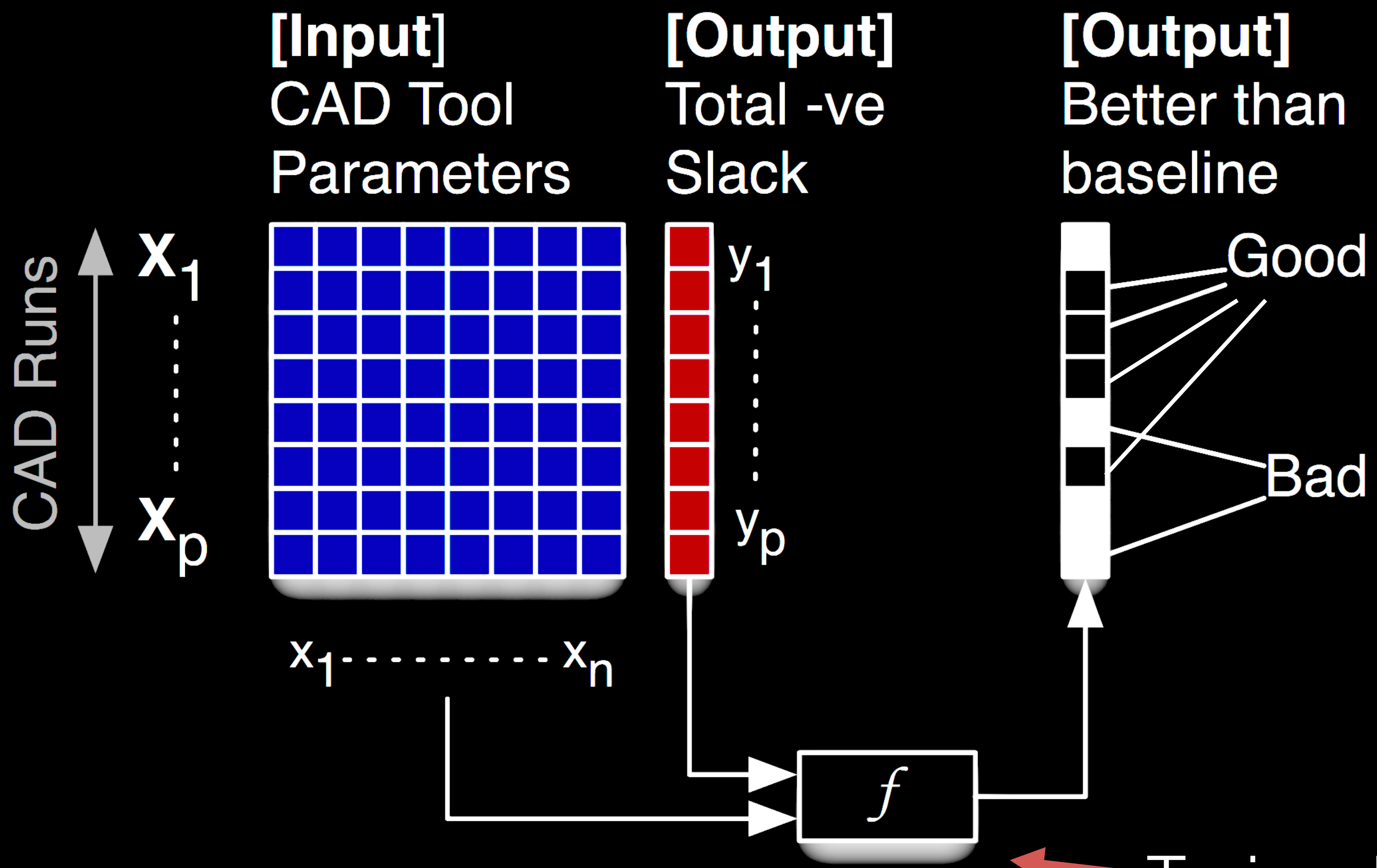
- Organized as a series of “rounds”
  — Each “round” consists of multiple CAD “runs”

- Model used to generate candidate sets of CAD parameters
  — Refined/corrected after each “round”
  — Final round is a trivial seed exploration

Learning Approach



Learning Approach



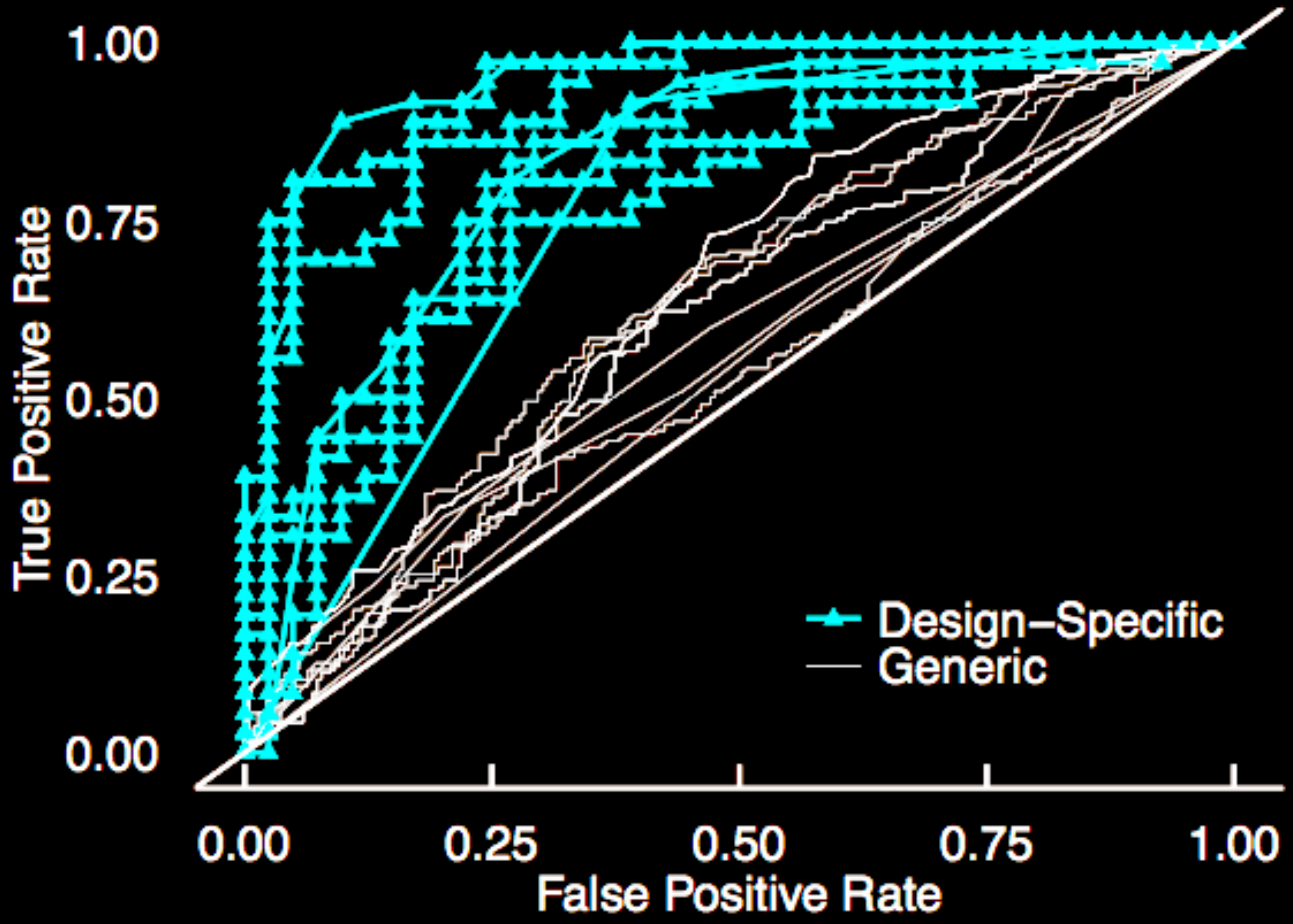
[Input]
CAD Tool
Parameters

[Output]
Total -ve
Slack

[Output]
Better than
baseline

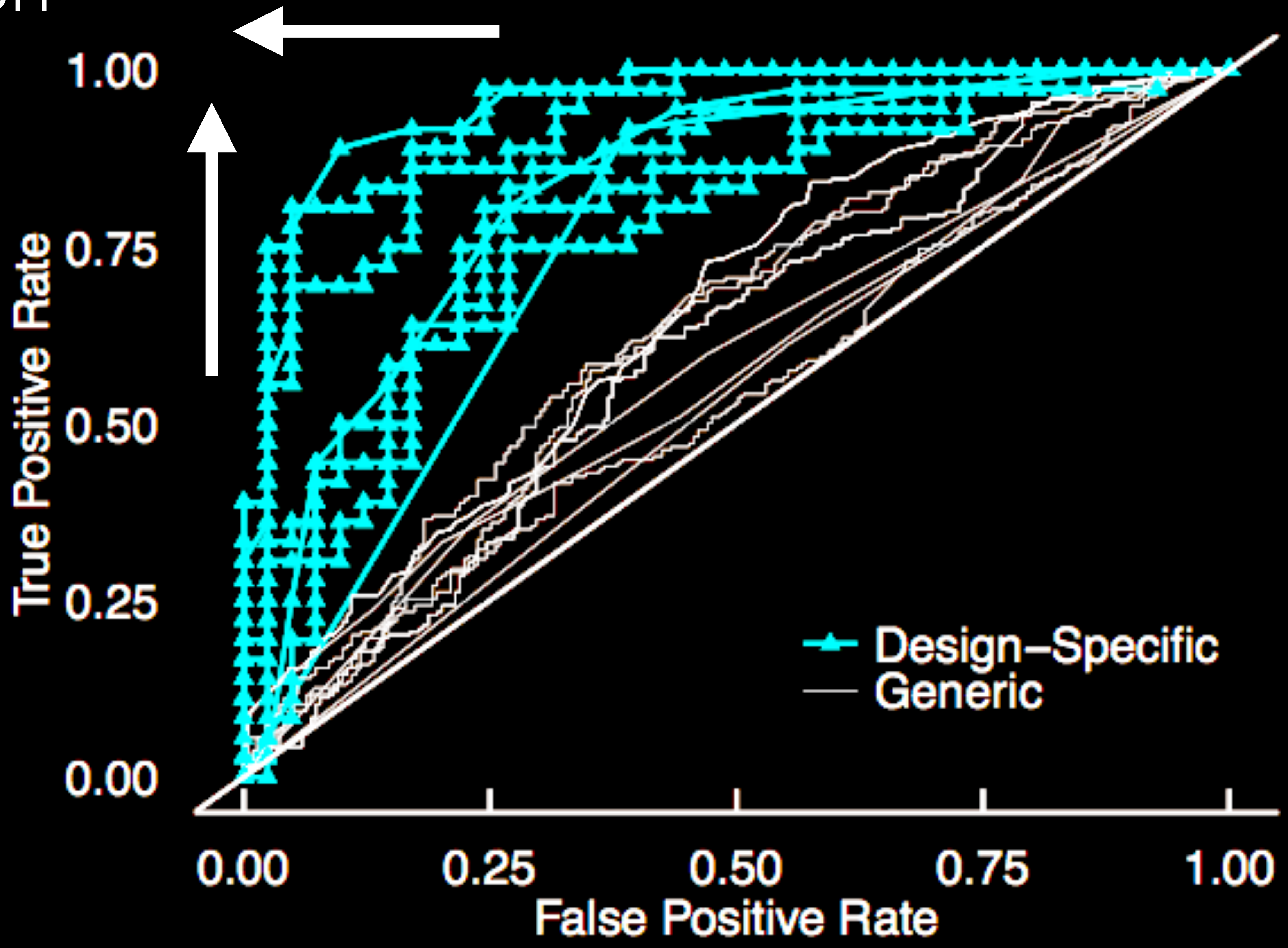
Trained per design
or benchmark?

Results (aes)

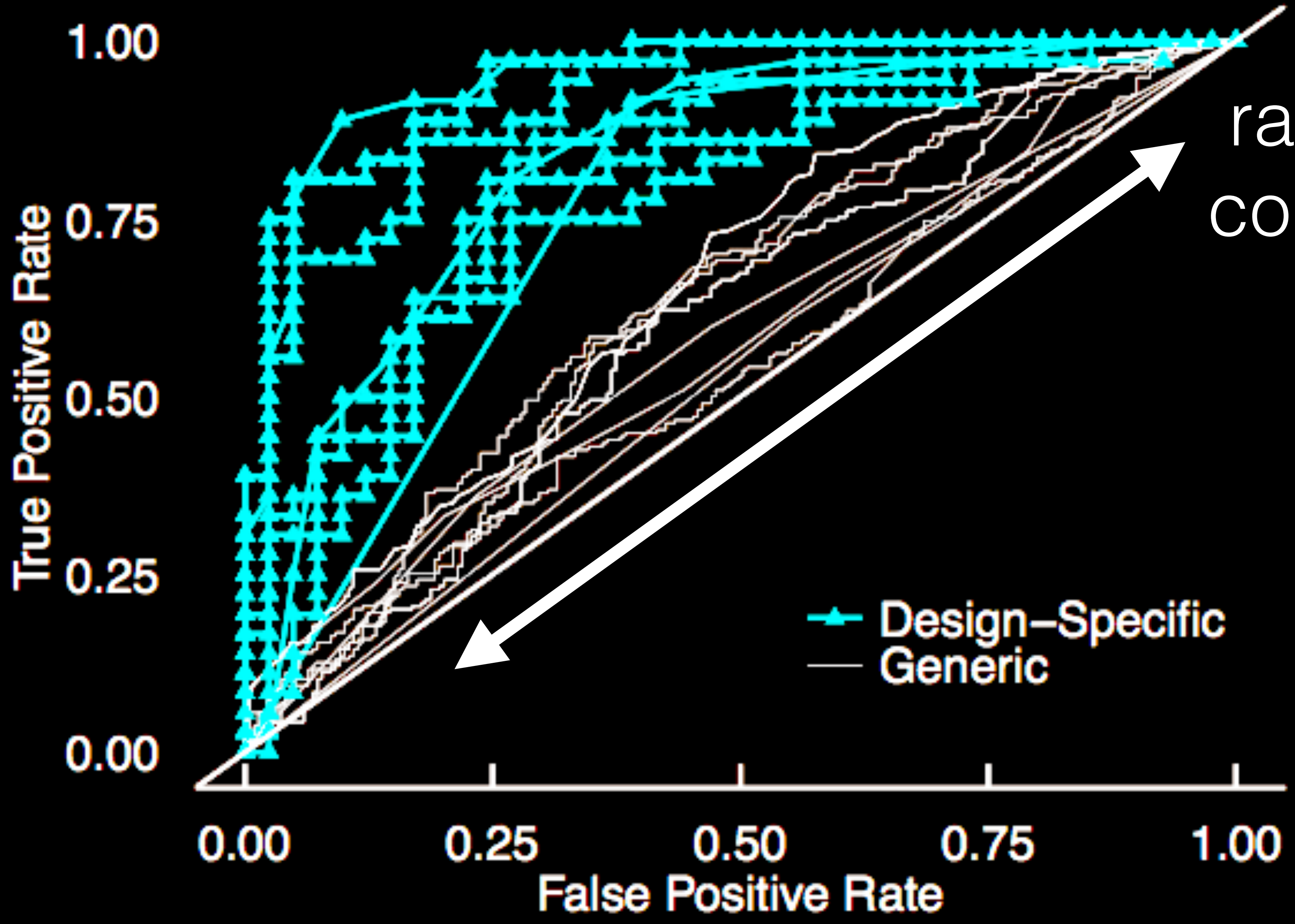

1.00
0.75
0.50
0.25
0.00
True Positive Rate
0.00
0.25
0.50
0.75
1.00
False Positive Rate
Design–Specific
Generic

Results (aes)

best classification

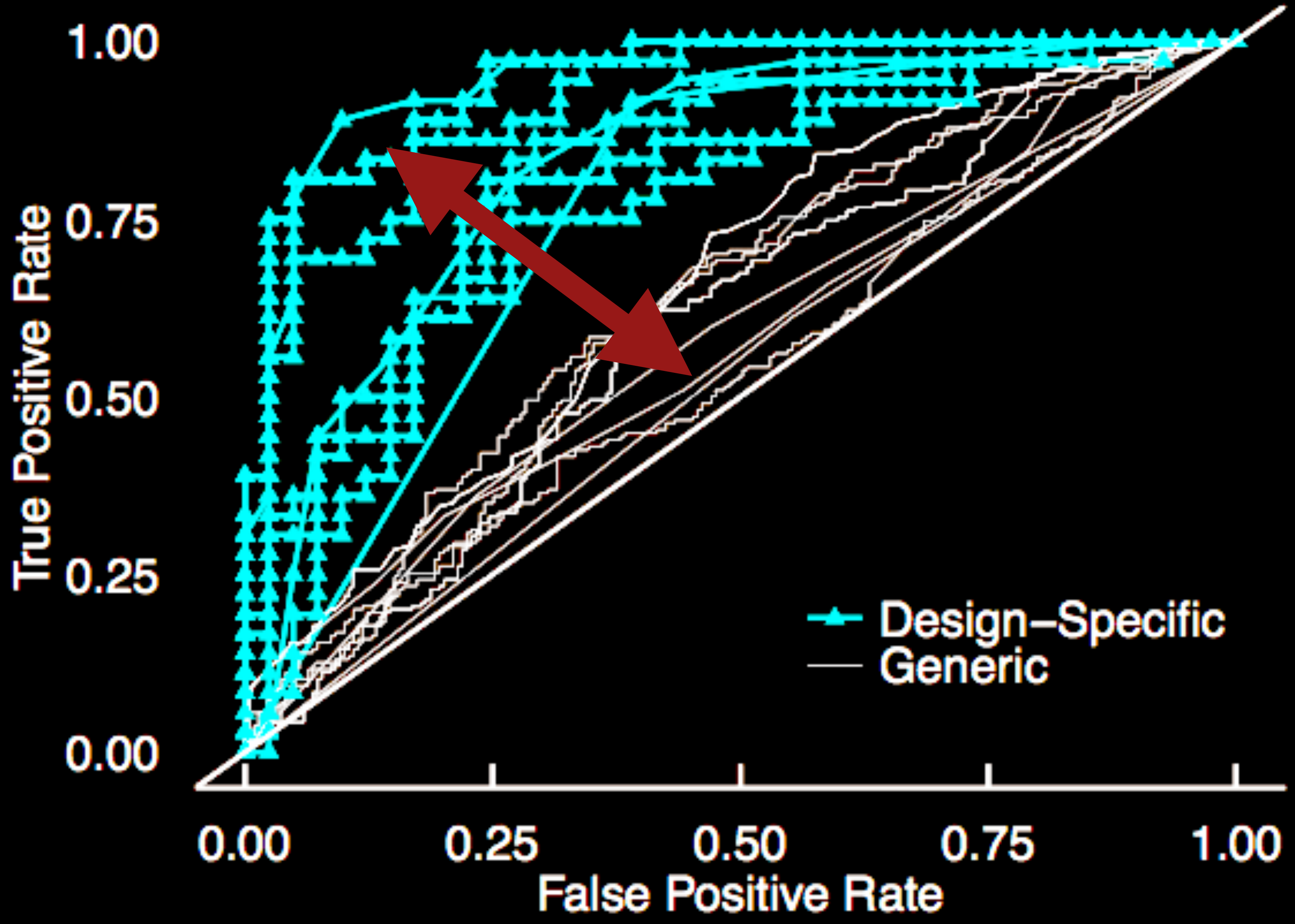


Results (aes)



random coin toss

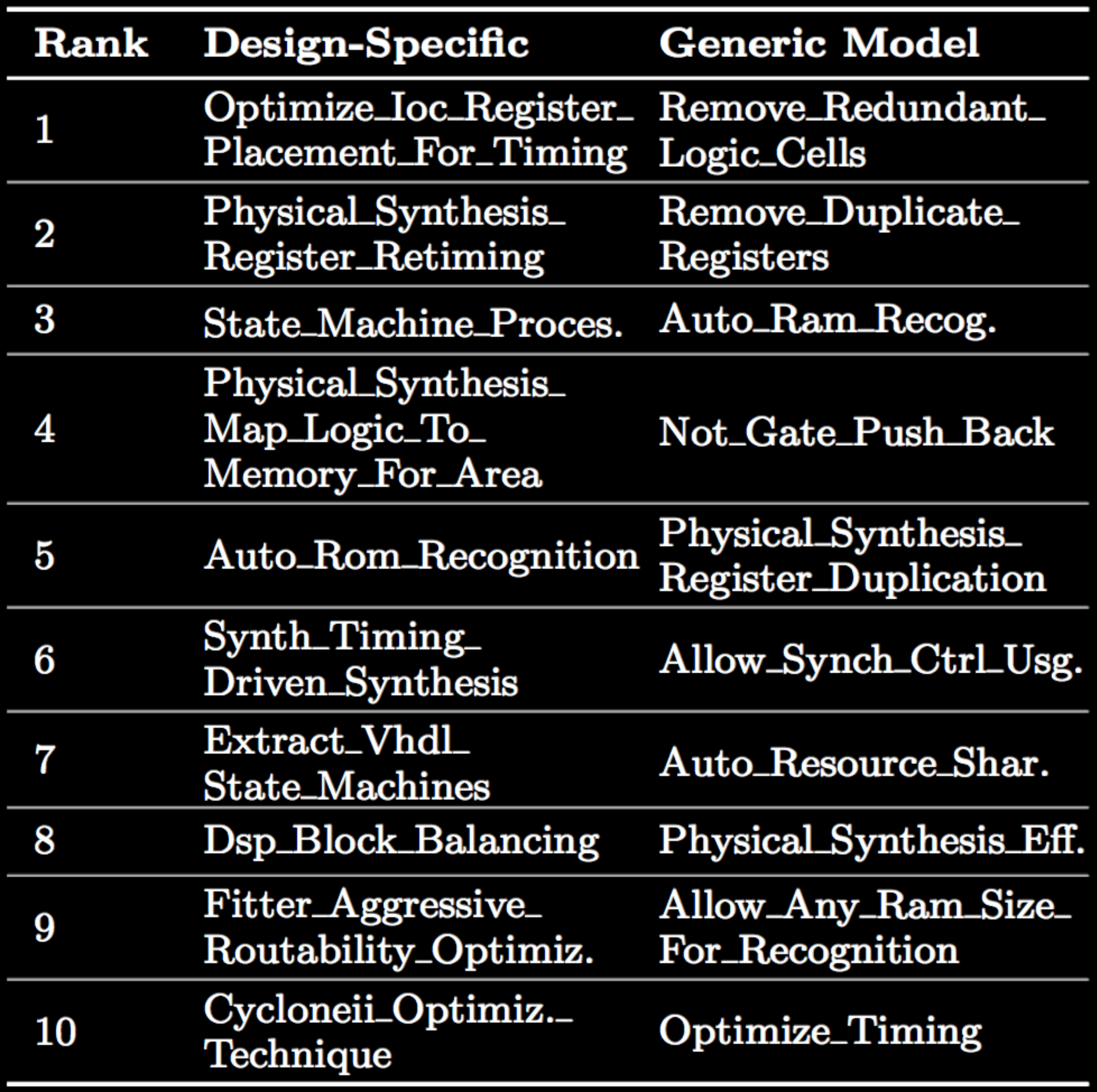
Results (aes)


1.00
0.75
0.50
0.25
0.00
True Positive Rate
0.00
0.25
0.50
0.75
1.00
False Positive Rate
Design–Specific
Generic

Conclusions

- Timing closure for FPGA designs is a challenge!
- FPGA CAD parameter selection offers a mechanism to deliver convergence
  — made possible by cheap cloud computing
- Design specific customisation AUC 0.7-0.8
  One-size fits all AUC 0.5-0.6 (random coin toss)
- Microsoft — We can give you a great discount!

Ranked Features (aes)



Rank	Design-Specific	Generic Model
1	Optimize_Ioc_Register_ Placement_For_Timing	Remove_Redundant_ Logic_Cells
2	Physical_Synthesis_ Register_Retiming	Remove_Duplicate_ Registers
3	State_Machine_Proces.	Auto_Ram_Recog.
4	Physical_Synthesis_ Map_Logic_To_ Memory_For_Area	Not_Gate_Push_Back
5	Auto_Rom_Recognition	Physical_Synthesis_ Register_Duplication
6	Synth_Timing_ Driven_Synthesis	Allow_Synch_Ctrl_Usg.
7	Extract_Vhdl_ State_Machines	Auto_Resource_Shar.
8	Dsp_Block_Balancing	Physical_Synthesis_Eff.
9	Fitter_Aggressive_ Routability_Optimiz.	Allow_Any_Ram_Size_ For_Recognition
10	Cycloneii_Optimiz._ Technique	Optimize_Timing