PVTSizing: A TuRBO-RL-Based Batch-Sampling Optimization Framework for PVT-Robust Analog Circuit Synthesis

Zichen Kong[1], Xiyuan Tang[1]*, Wei Shi[2], Yiheng Du[1], Yibo Lin[1], Yuan Wang[1]*

[1]Peking University
[2]The University of Texas at Austin

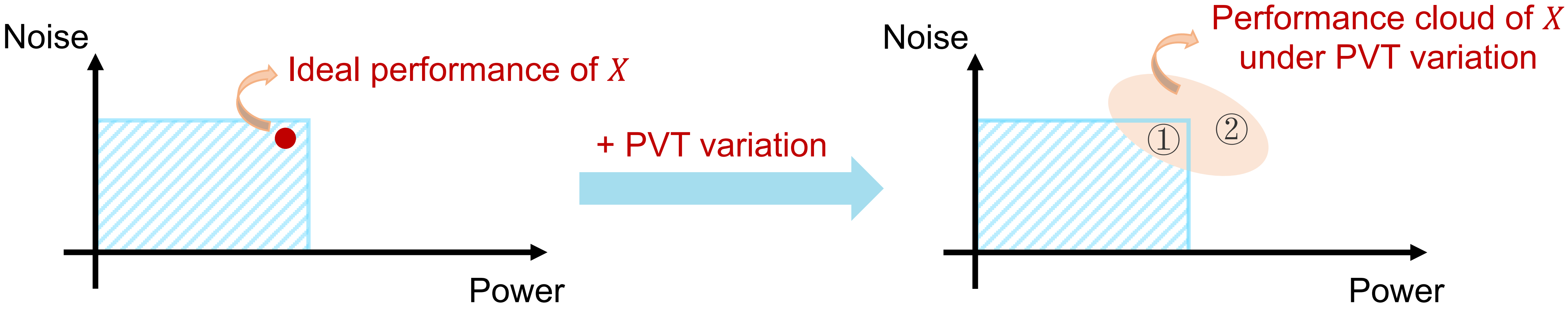
*Corresponding authors

Outline

- Background & Introduction
- Proposed PVTSizing
- Experimental Results
- Conclusion

PVT robustness is significant for analog synthesis.

$X = (W_1, L_1, \dots)$ is a certain set of design solution of analog synthesis.



▨ Performance space that meets given specifications

① X meets given specifications under some PVT corners.

② X fails to meet given specifications under some other PVT corners.

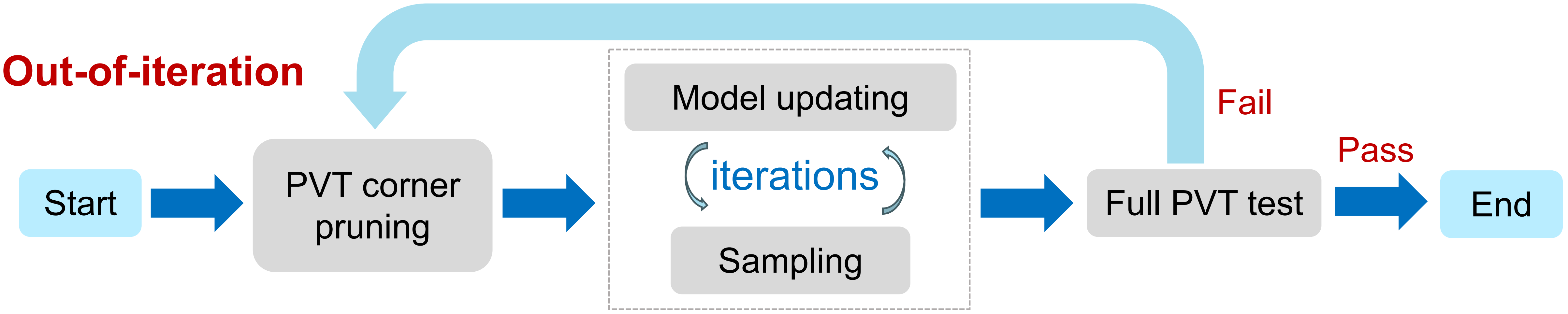
Challenge of PVT-robust analog synthesis

Challenge 1: Randomness of starting point limits success rate.

- The mapping between device sizes and circuit performance is highly nonlinear.
- With bad starting point, it’s easy to stuck at the local optimum for a long time.

Challenge of PVT-robust analog synthesis

Challenge 2: Out-of-iteration PVT corner pruning leads to corner-overfitting.



Out-of-iteration

Start

PVT corner pruning

Model updating

iterations

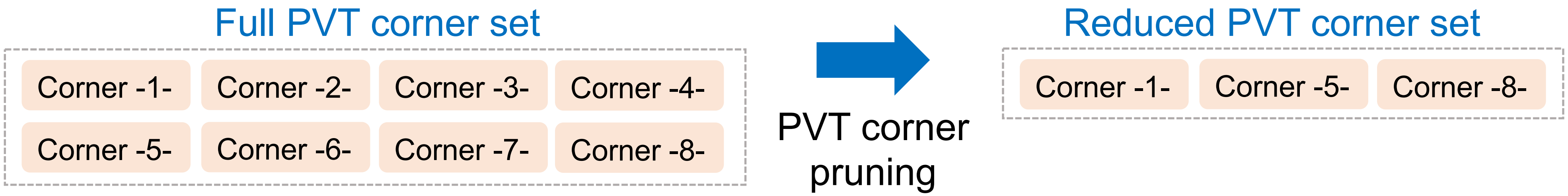
Sampling

Full PVT test

Fail

Pass

End



Full PVT corner set

Corner -1-
Corner -2-
Corner -3-
Corner -4-
Corner -5-
Corner -6-
Corner -7-
Corner -8-

PVT corner pruning

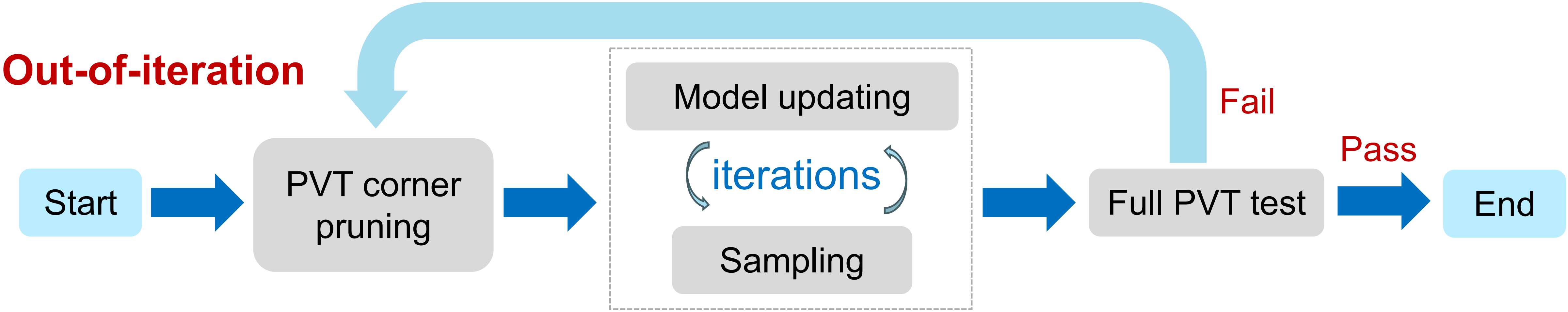
Reduced PVT corner set

Corner -1-
Corner -5-
Corner -8-

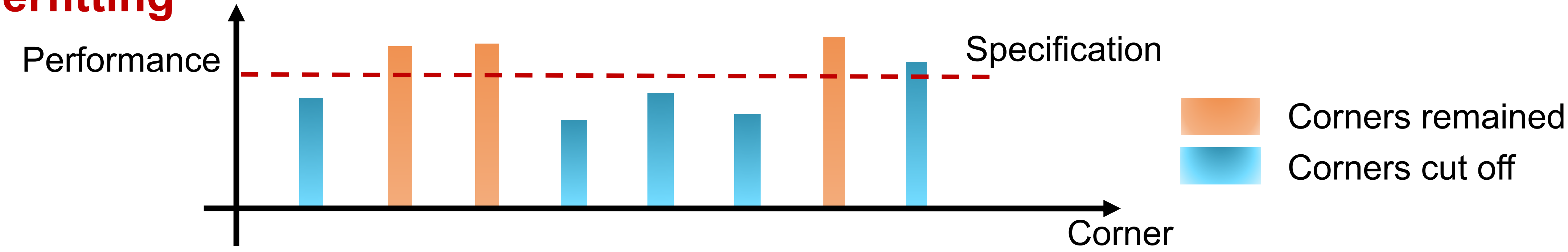
Challenge of PVT-robust analog synthesis

Challenge 2: Out-of-iteration PVT corner pruning leads to corner-overfitting.

Out-of-iteration

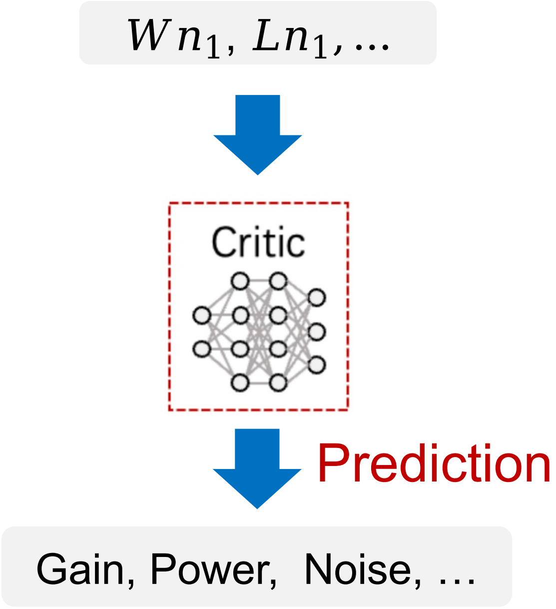


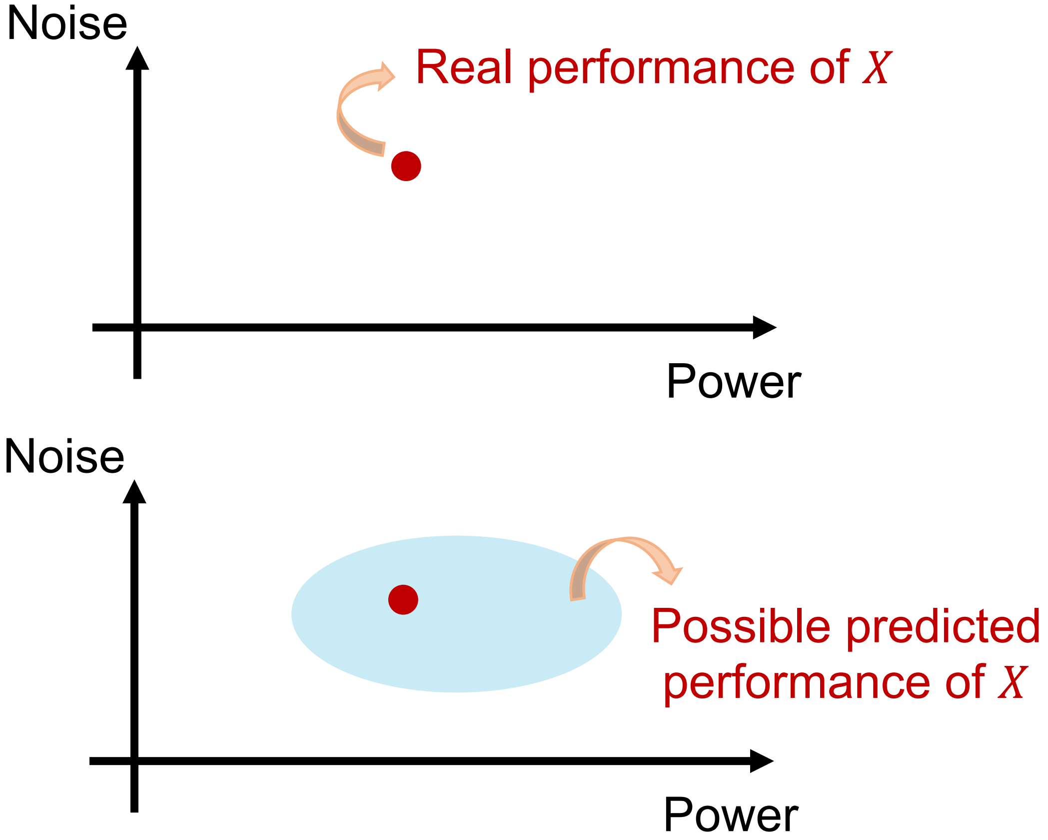
Corner-overfitting



Challenge of PVT-robust analog synthesis

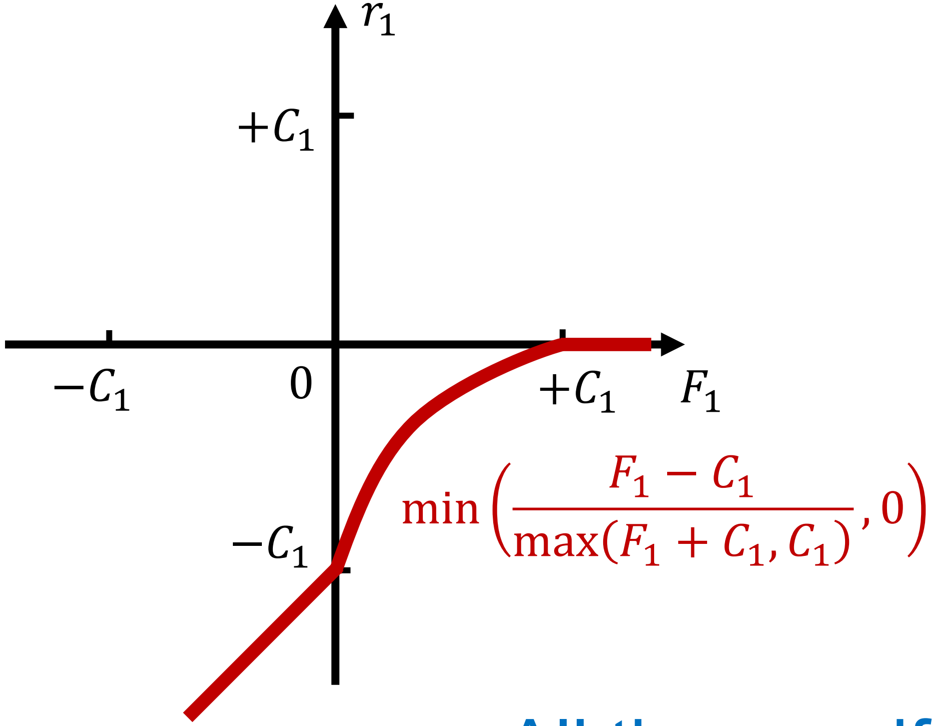
Challenge 3: Inaccurate prediction limits the effectiveness of sample pruning.





Formulation of PVT-robust analog synthesis

Define figure of merit (FoM) to convert multiple specifications into a single target.



$$FoM = \begin{cases} r, & r < 0 \\ 0.2, & r \geq 0 \end{cases}$$

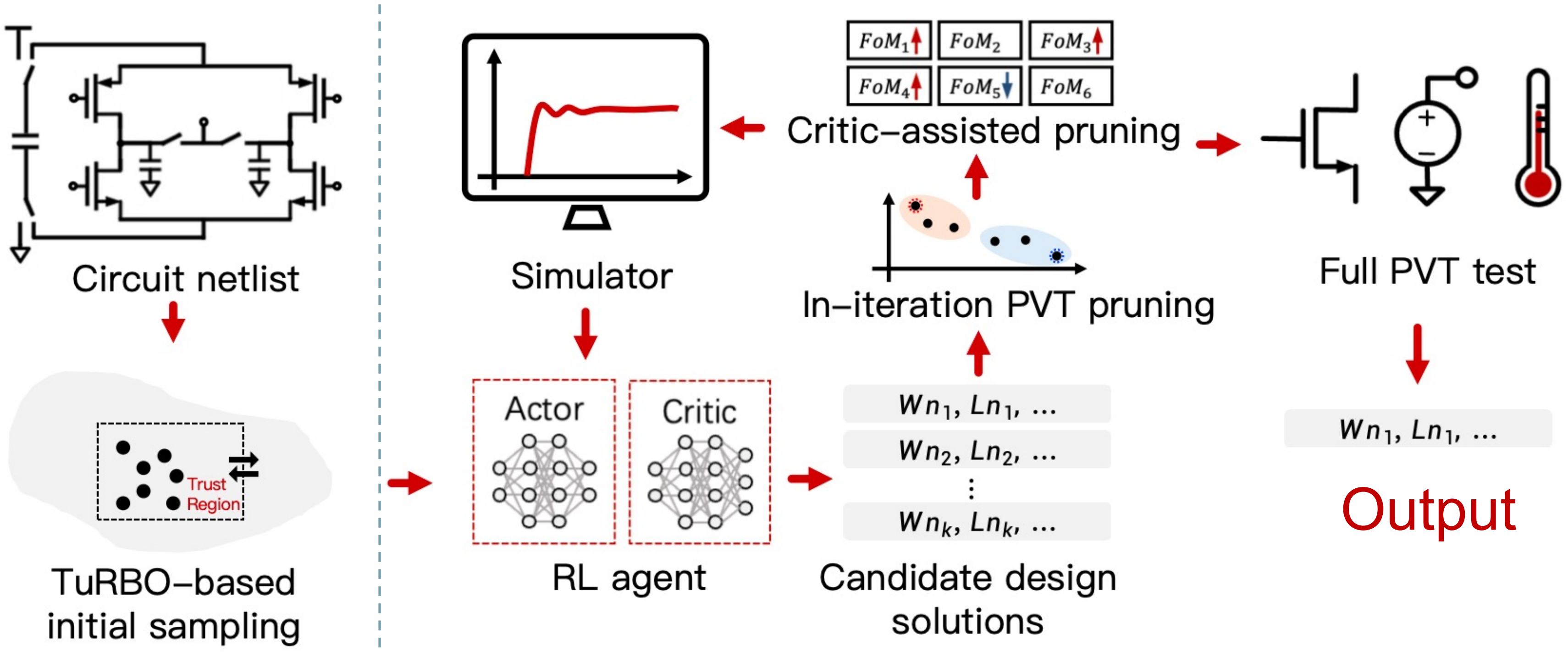
Performance

Specification

$$r = \begin{cases} \sum_{i=1}^{m} \min\left(\frac{\boxed{F_i} - \boxed{C_i}}{\max(F_i + C_i, C_i)}, 0\right), if\ we\ want\ F_i \geq C_i \\ \sum_{i=1}^{m} \min\left(\frac{C_i - F_i}{\max(F_i + C_i, C_i)}, 0\right), if\ we\ want\ F_i \leq C_i \end{cases}$$

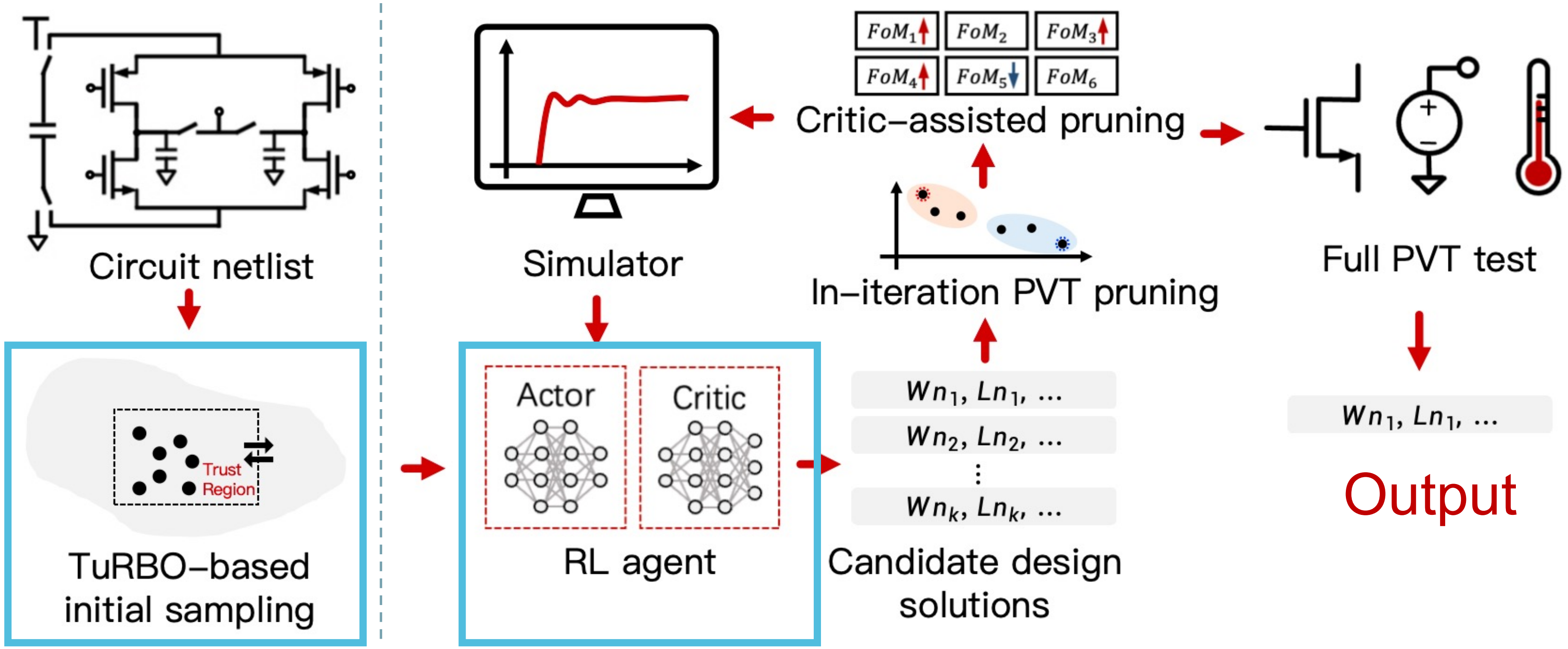
All the specifications are met when $FoM \geq 0$.

This work: PVTSizing



Overview of PVTSizing

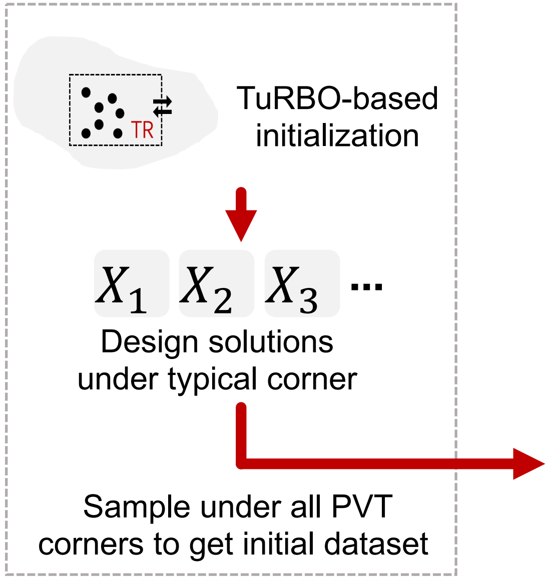
This work: PVTSizing



Contribution 1/3: Divide sizing problem into 1) single-corner initial sampling phase and 2) multi-corner sizing phase to improve success rate.

Contribution 1: TuRBO-RL-based optimization framework

Initialization

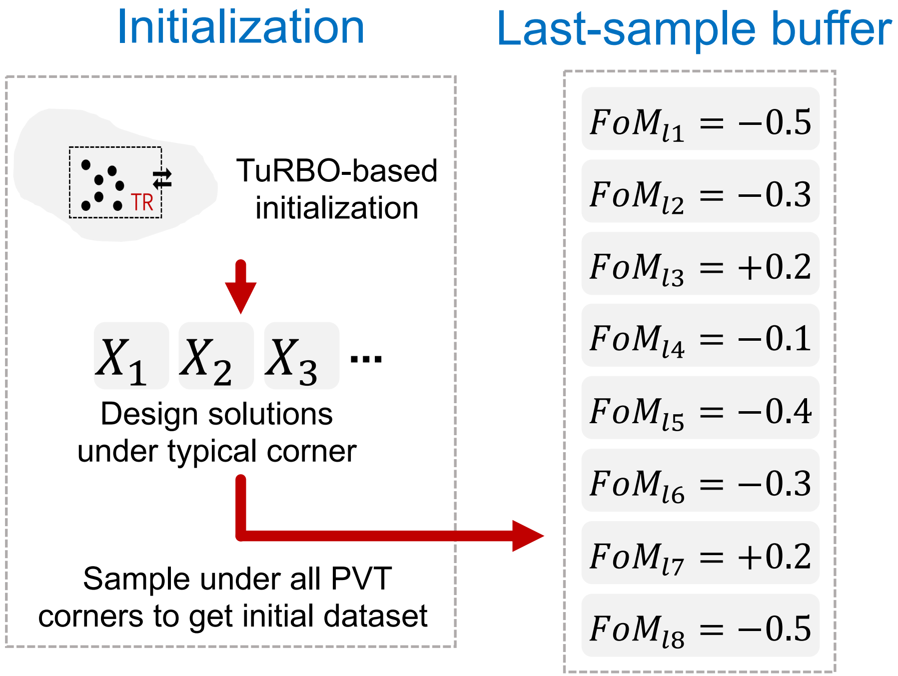


Design solutions under typical corner

Sample under all PVT corners to get initial dataset

Do TuRBO-based initial sampling.

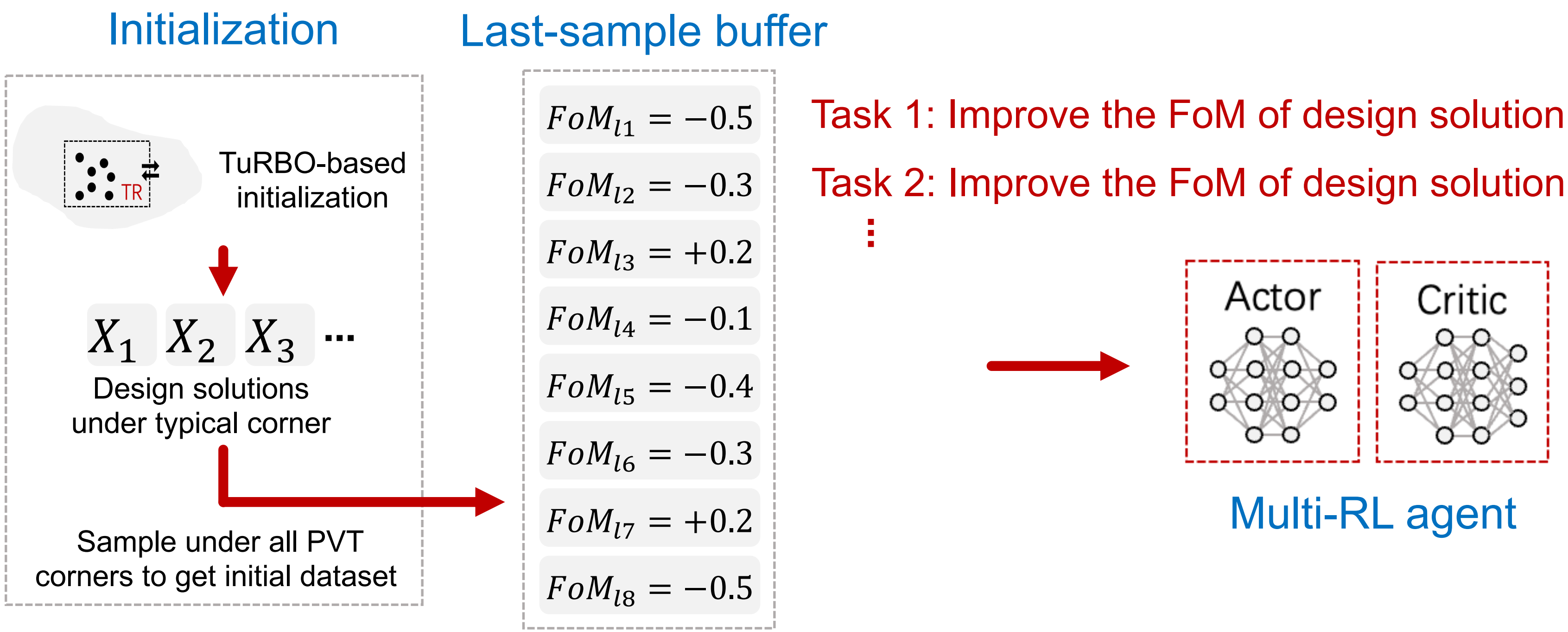
Contribution 1: TuRBO-RL-based optimization framework



$FoM_{l1} = -0.5$
$FoM_{l2} = -0.3$
$FoM_{l3} = +0.2$
$FoM_{l4} = -0.1$
$FoM_{l5} = -0.4$
$FoM_{l6} = -0.3$
$FoM_{l7} = +0.2$
$FoM_{l8} = -0.5$

Build last-sample buffer to record the results of latest sampling of each PVT corner.

Contribution 1: TuRBO-RL-based optimization framework



Initialization

TuRBO-based initialization

X_1 X_2 X_3 ···

Design solutions under typical corner

Sample under all PVT corners to get initial dataset

Last-sample buffer

$FoM_{l1} = -0.5$

$FoM_{l2} = -0.3$

$FoM_{l3} = +0.2$

$FoM_{l4} = -0.1$

$FoM_{l5} = -0.4$

$FoM_{l6} = -0.3$

$FoM_{l7} = +0.2$

$FoM_{l8} = -0.5$

Task 1: Improve the FoM of design solution under corner -1-.

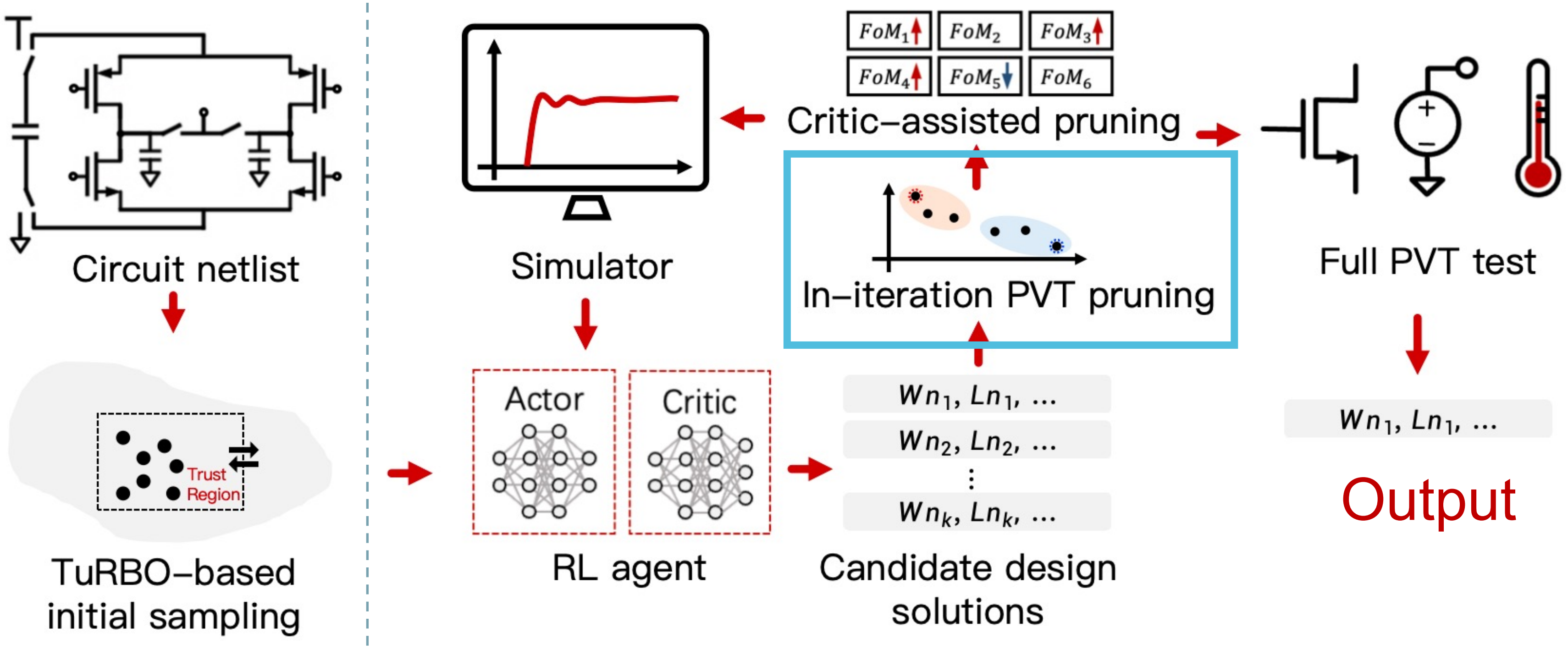
Task 2: Improve the FoM of design solution under corner -2-.

⋮

Multi-RL agent

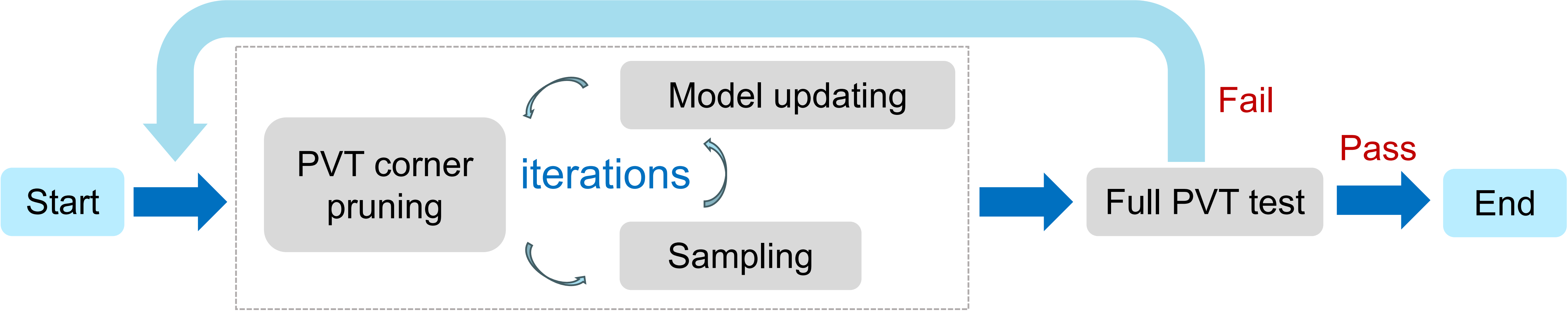
Multi-task reinforcement learning for multi-corner sizing.

This work: PVTSizing

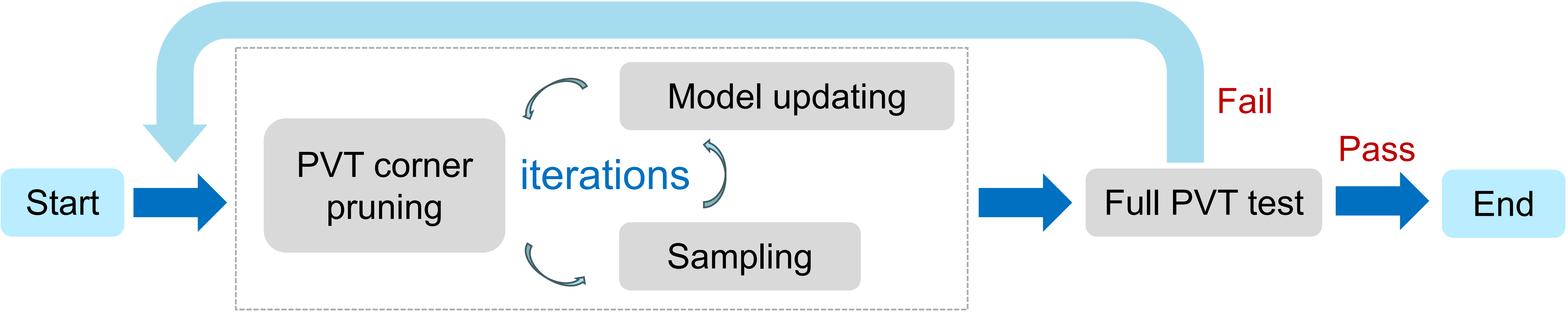


Contribution 2/3: In-iteration PVT pruning based on last-sample buffer

Contribution 2: In-iteration PVT pruning



Contribution 2: In-iteration PVT pruning



Start

PVT corner pruning

iterations

Model updating

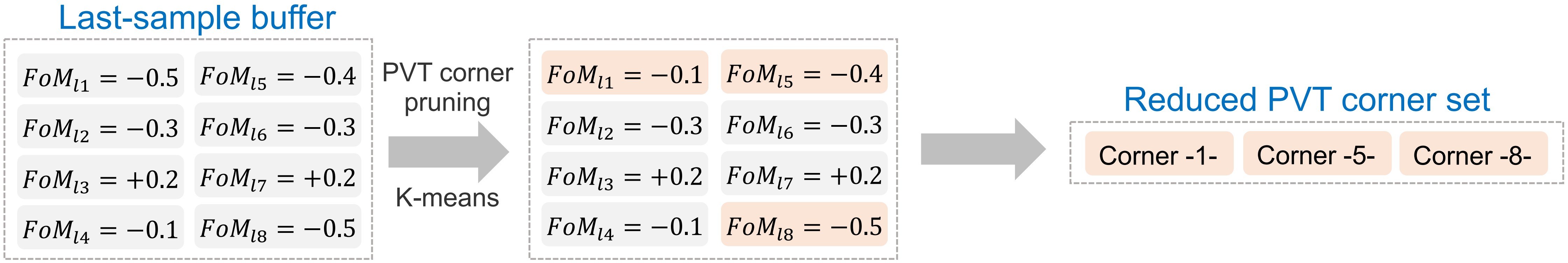
Sampling

Full PVT test

Fail

Pass

End



Last-sample buffer

$FoM_{l1} = -0.5$	$FoM_{l5} = -0.4$
$FoM_{l2} = -0.3$	$FoM_{l6} = -0.3$
$FoM_{l3} = +0.2$	$FoM_{l7} = +0.2$
$FoM_{l4} = -0.1$	$FoM_{l8} = -0.5$

PVT corner pruning

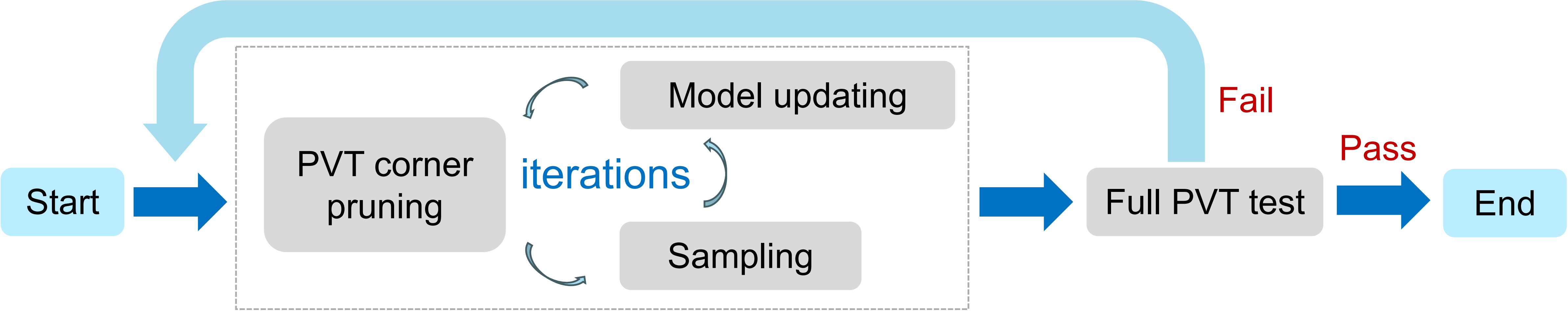
K-means

$FoM_{l1} = -0.1$	$FoM_{l5} = -0.4$
$FoM_{l2} = -0.3$	$FoM_{l6} = -0.3$
$FoM_{l3} = +0.2$	$FoM_{l7} = +0.2$
$FoM_{l4} = -0.1$	$FoM_{l8} = -0.5$

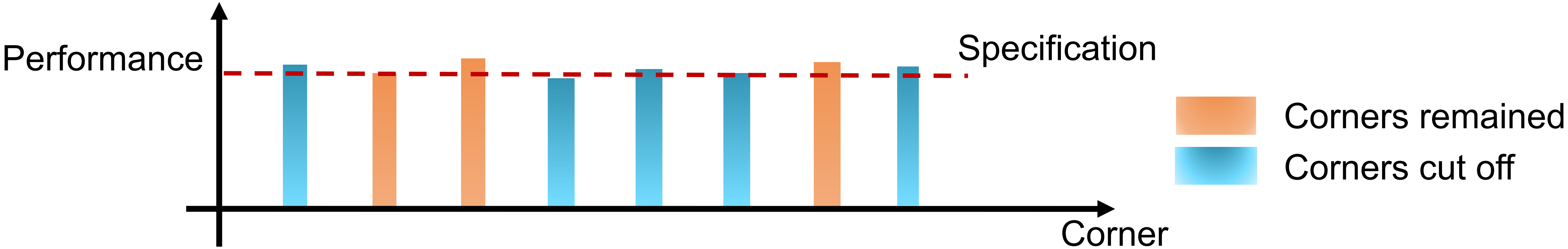
Reduced PVT corner set

Corner -1- Corner -5- Corner -8-

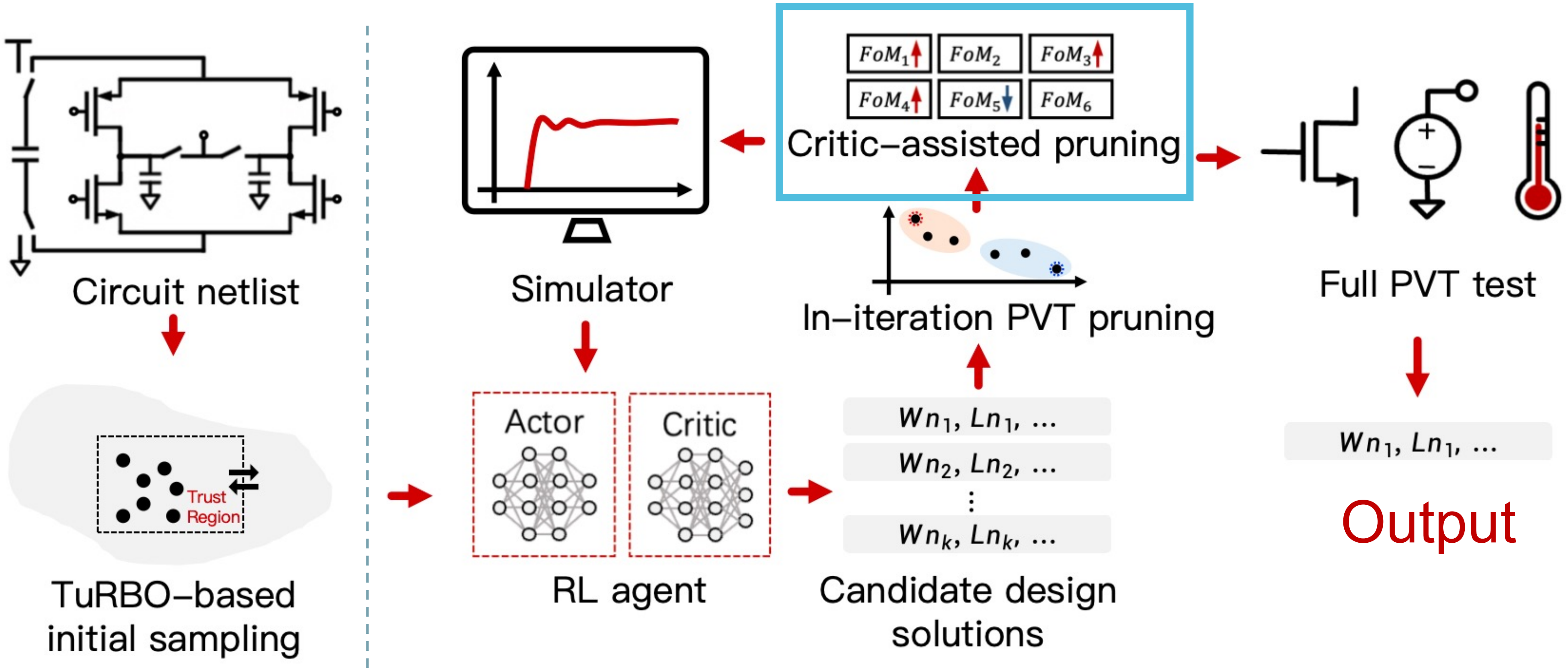
Contribution 2: In-iteration PVT pruning



w/o corner-overfitting



This work: PVTSizing



Contribution 3/3: Critic-assisted pruning for selecting candidate design solutions

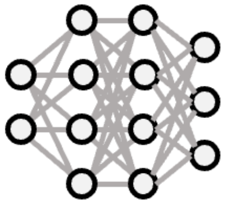
Contribution 3: Critic-assisted pruning

Reduced PVT corner set

Corner -1- Corner -5- Corner -8-

$X_{candidate}$

Critic



Predicted change in last-sample buffer

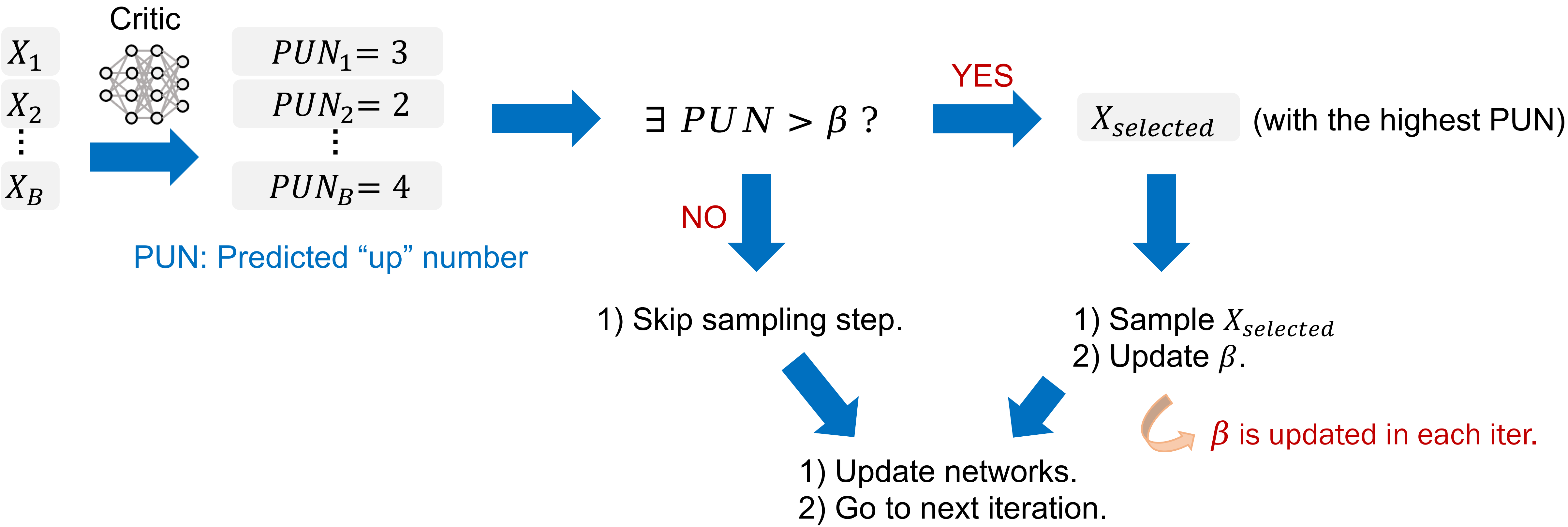
$FoM_{l1} = -0.5$	$FoM_{l1} = -0.1 > -0.5$	UP
$FoM_{l2} = -0.3$	$FoM_{l2} = -0.3$	
$FoM_{l3} = +0.2$	$FoM_{l3} = +0.2$	
$FoM_{l4} = -0.1$	$FoM_{l4} = -0.1$	
$FoM_{l5} = -0.4$	$FoM_{l5} = -0.5 < -0.4$	DOWN
$FoM_{l6} = -0.3$	$FoM_{l6} = -0.3$	
$FoM_{l7} = +0.2$	$FoM_{l7} = +0.2$	
$FoM_{l8} = -0.5$	$FoM_{l8} = -0.3 > -0.5$	UP

Predicted “up” number = 2

Predicted “down” number = 1

Use Critic network to predict the relative FoM of latest sample and $X_{candidate}$.

Contribution 3: Critic-assisted pruning



Critic

X_1

X_2

⋮

X_B

$PUN_1 = 3$

$PUN_2 = 2$

⋮

$PUN_B = 4$

PUN: Predicted "up" number

$\exists\ PUN > \beta$?

YES

$X_{selected}$ (with the highest PUN)

NO

1) Skip sampling step.

1) Sample $X_{selected}$

2) Update β.

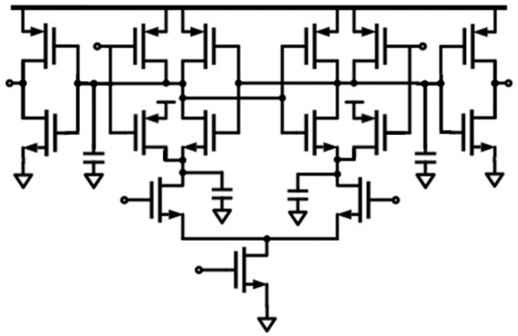
β is updated in each iter.

1) Update networks.

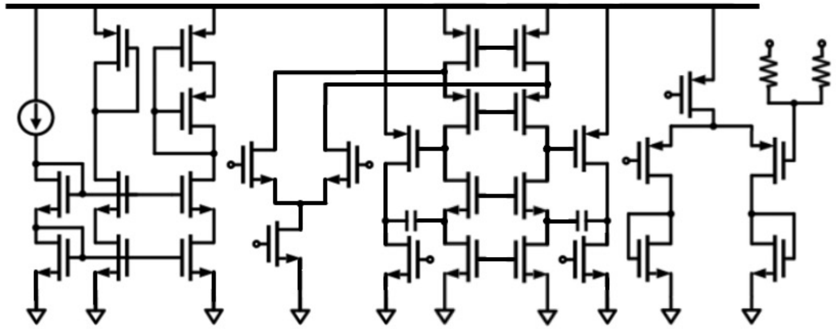
2) Go to next iteration.

Set sample threshold β for further pruning.

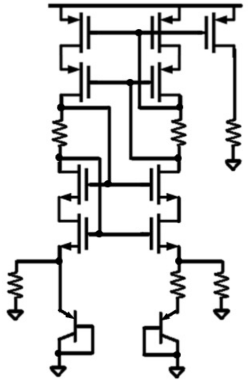
Testcases



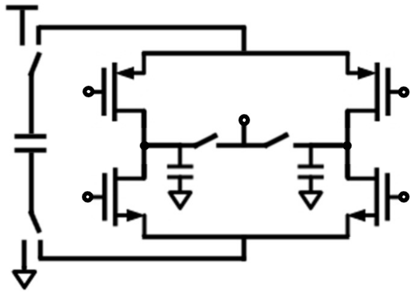
StrongArm Latch (SA)



Folded-Cascode OTA (FC)



Bandgap Reference (BGR)

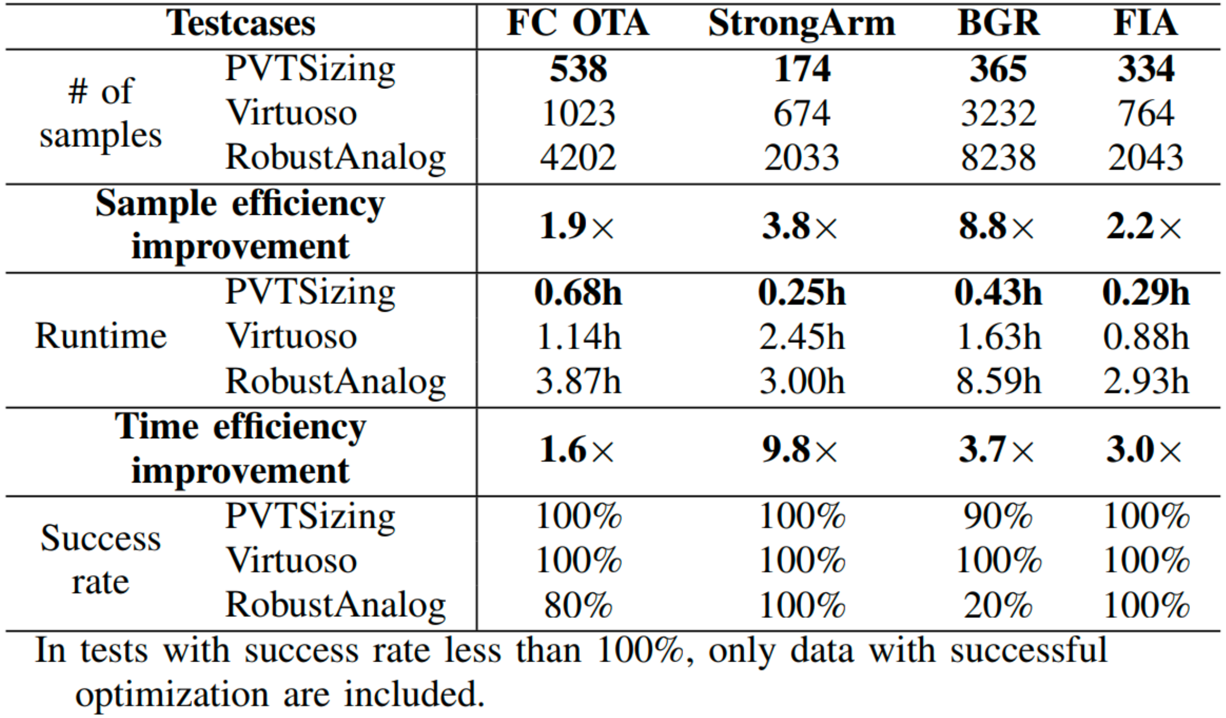


Floating Inverter Pre-Amplifier (FIA)

Topology	# of variables	# of constraints	Scale of design space
SA	14	4	10^{24}
FC	20	7	10^{55}
BGR	8	3	10^{23}
FIA	6	2	10^{17}

Synthesis results on four real-world circuits

- 1.9~8.8x sample efficiency and 1.6~9.8x time efficiency compared to tools from industry and academia.
- Remain high success rate.

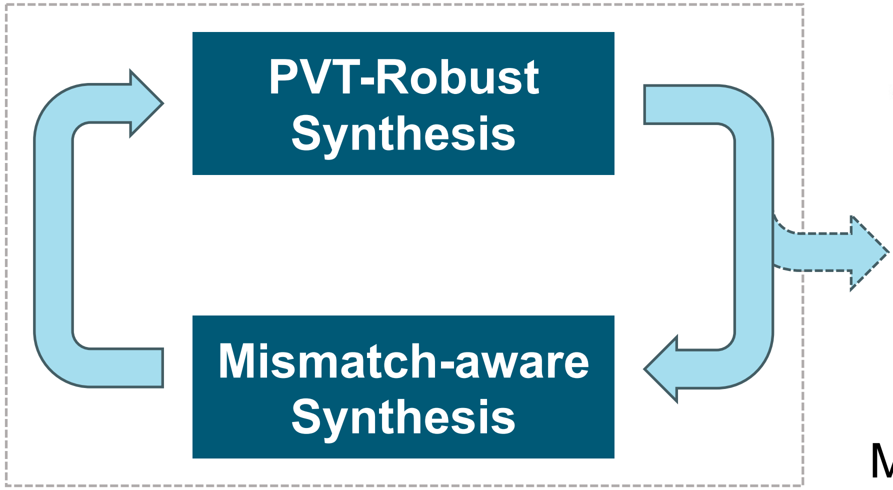


Testcases		FC OTA	StrongArm	BGR	FIA
# of samples	PVTSizing	**538**	**174**	**365**	**334**
	Virtuoso	1023	674	3232	764
	RobustAnalog	4202	2033	8238	2043
Sample efficiency improvement		**1.9×**	**3.8×**	**8.8×**	**2.2×**
Runtime	PVTSizing	**0.68h**	**0.25h**	**0.43h**	**0.29h**
	Virtuoso	1.14h	2.45h	1.63h	0.88h
	RobustAnalog	3.87h	3.00h	8.59h	2.93h
Time efficiency improvement		**1.6×**	**9.8×**	**3.7×**	**3.0×**
Success rate	PVTSizing	100%	100%	90%	100%
	Virtuoso	100%	100%	100%	100%
	RobustAnalog	80%	100%	20%	100%

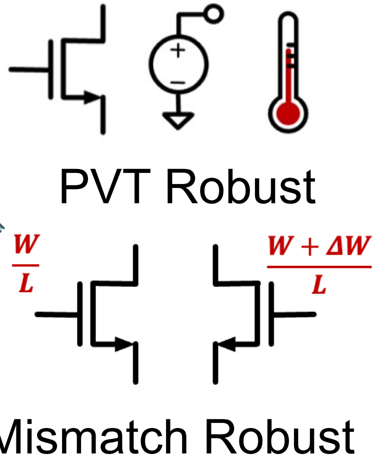
In tests with success rate less than 100%, only data with successful optimization are included.

Support for mismatch-aware synthesis & Results

- In Monte Carlo simulation, each simulation point corresponds to a set of mismatch parameters.
- Each set of mismatch parameters can be regarded as a mismatch corner.
- By treating mismatch corner as PVT corner, PVTSizing can handle mismatch-aware synthesis.


PVT-Robust Synthesis

Mismatch-aware Synthesis



Results of mismatch-aware synthesis on FC OTA

Framework	# of samples	Runtime	Successed?
PVTSizing	**823**	**3.73h**	Yes
Virtuoso	5267	5.27h	Yes
Improvement	6.4×	1.4×	/

Conclusion

- PVTSizing
  - A TuRBO-RL framework is proposed for high-success-rate PVT robust analog synthesis.
  - In-iteration PVT corner pruning is proposed to avoid corner-overfitting.
  - Critic-assisted pruning is proposed to improve effectiveness of performance prediction.
  - Mismatch-aware synthesis is supported.
- Results
  - 1.9~8.8x sample efficiency and 1.6~9.8x time efficiency compared to tools from industry and academia.

Thanks for your attention.