Fast IR Drop Estimation with Machine Learning

Zhiyao Xie*, Hai Li*, Xiaoqing Xu‡, Jiang Hu§, Yiran Chen*

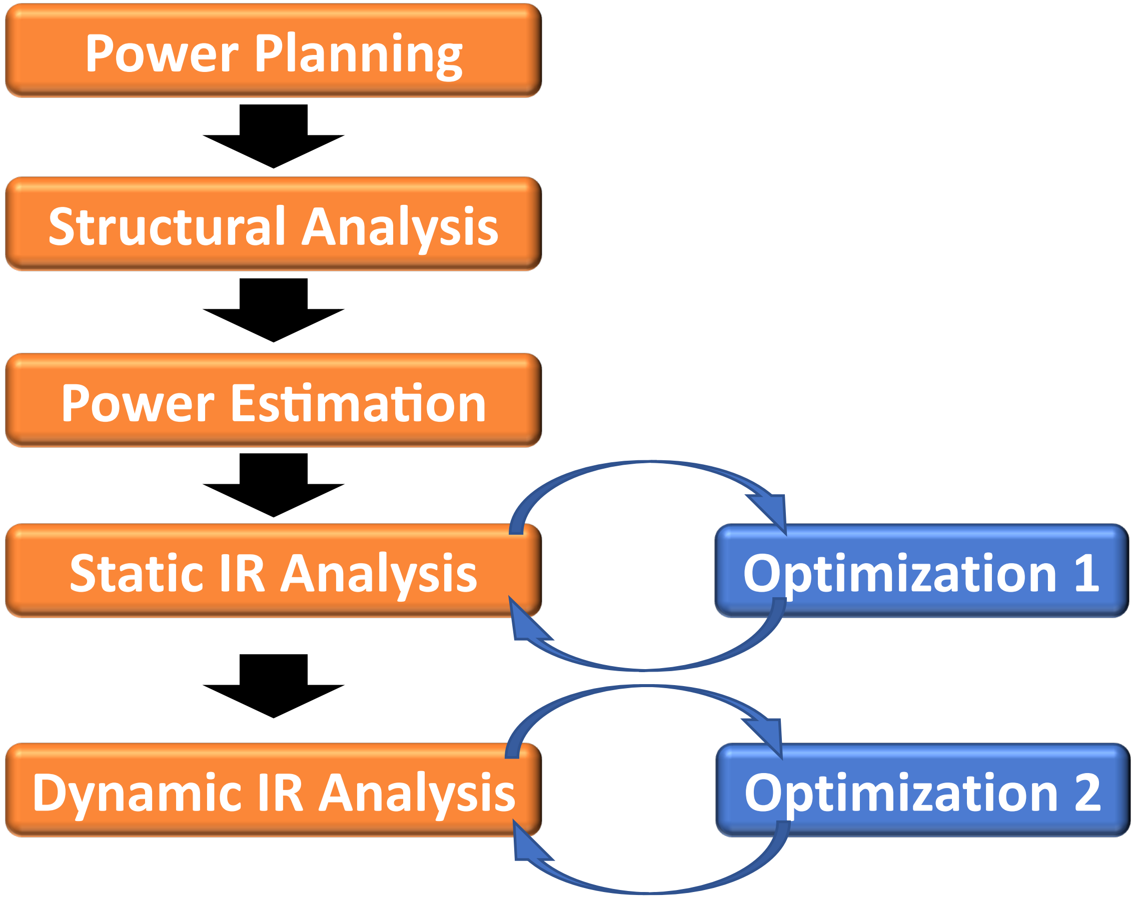
*Duke University, ‡ARM Inc, §Texas A&M University

Outline

- Background
- An overview of ML-based IR drop estimators
- Static IR drop estimators
- Dynamic IR drop estimators
- General challenges and possible solutions

Challenge: Faster IR Drop Estimator Desired

- To optimize PPA & meet constraint, IR drop mitigation may **takes many iterations**
- Accurate IR drop simulation by commercial tools is **very time consuming**
- Fast IR drop estimation with machine learning (**ML**) !

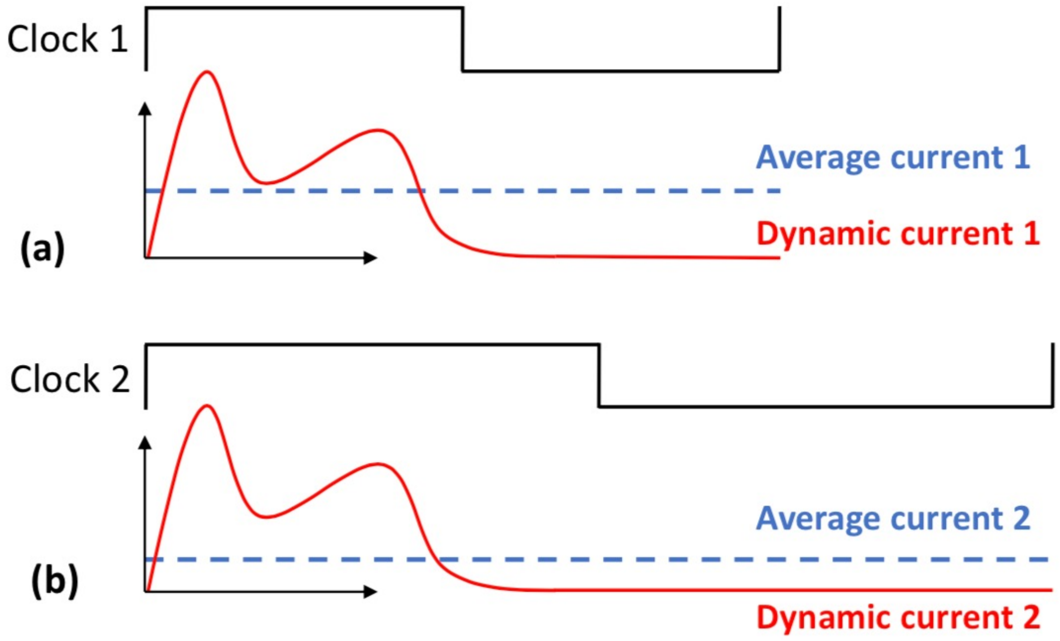


Early rail analysis by Voltus. [*]

[*] Cadence, Voltus User Guide.

Background: IR Drop Types

- Static IR drop
  - Usually measures the average current
  - Identify the weakness of PDN
- Dynamic IR drop
  - Captures the peak transient current
  - Whether toggling vectors required:
    - Yes: Vector-based IR drop
    - No: Vectorless IR drop
- Power Supply Noise (PSN)
  - Comprises both IR drop & $L * di/dt$



Static and dynamic analysis. [*]

[*] Nithin, et al., ISQED

Methods: IR Drop Estimators Overview

Methods	Type of IR Drop	Time	ML Model	Cross-Design	Features	Objective
IncPIRD	Static IR	2019	XGBoost	Yes	I_g, R_g, PDN, G	IR mitigation
XGBIR		2020	XGBoost	-	I_g, R_g, PDN	PDN design
Yamato *et al.*	Vector-based IR	2012	Linear Regression	No	P_c	IR-aware timing
Dhotre *et al.*		2017	Clustering	Yes	r_{tog}, c	IR prediction
Lin *et al.*		2018	ANN	No	P_c, R_c, t_c, r_{tog}, c	IR mitigation
Fang *et al.*		2018	CNN, XGBoost	No	P_c, I_c, R_c, t_c, r_{tog}, c	IR mitigation
PowerNet	Vectorless IR	2020	CNN	Yes	P_c, t_c, r_{tog}	IR mitigation
Mozaffari *et al.*	Sillicon PSN	2019	ANN, CNN, NLP	Yes	r_{tog}, G	PSN prediction

The existing ML estimators cover different types of IR drop:

- Static IR drop
- Dynamic IR drop
- Power Supply Noise (PSN)

Methods: IR Drop Estimators Overview

Methods	Type of IR Drop	Time	ML Model	Cross-Design	Features	Objective
IncPIRD	Static IR	2019	XGBoost	Yes	I_g, R_g, PDN, G	IR mitigation
XGBIR		2020	XGBoost	-	I_g, R_g, PDN	PDN design
Yamato *et al.*	Vector-based IR	2012	Linear Regression	No	P_c	IR-aware timing
Dhotre *et al.*		2017	Clustering	Yes	r_{tog}, c	IR prediction
Lin *et al.*		2018	ANN	No	P_c, R_c, t_c, r_{tog}, c	IR mitigation
Fang *et al.*		2018	CNN, XGBoost	No	P_c, I_c, R_c, t_c, r_{tog}, c	IR mitigation
PowerNet	Vectorless IR	2020	CNN	Yes	P_c, t_c, r_{tog}	IR mitigation
Mozaffari *et al.*	Sillicon PSN	2019	ANN, CNN, NLP	Yes	r_{tog}, G	PSN prediction

The existing ML estimators cover different ML models:

- One-dimensional input: Linear Regression, ANN, XGBoost, ….
- Two-dimensional input: CNN, NLP…

Methods: IR Drop Estimators Overview

Methods	Type of IR Drop	Time	ML Model	Cross-Design	Features	Objective
IncPIRD	Static IR	2019	XGBoost	Yes	I_g, R_g, PDN, G	IR mitigation
XGBIR		2020	XGBoost	-	I_g, R_g, PDN	PDN design
Yamato *et al.*	Vector-based IR	2012	Linear Regression	No	P_c	IR-aware timing
Dhotre *et al.*		2017	Clustering	Yes	r_{tog}, c	IR prediction
Lin *et al.*		2018	ANN	No	P_c, R_c, t_c, r_{tog}, c	IR mitigation
Fang *et al.*		2018	CNN, XGBoost	No	P_c, I_c, R_c, t_c, r_{tog}, c	IR mitigation
PowerNet	Vectorless IR	2020	CNN	Yes	P_c, t_c, r_{tog}	IR mitigation
Mozaffari *et al.*	Sillicon PSN	2019	ANN, CNN, NLP	Yes	r_{tog}, G	PSN prediction

Cross-design:

- Model applies to new designs that are not in the training set
- Test and training design differ at least at netlist level

Methods: IR Drop Estimators Overview

Methods	Type of IR Drop	Time	ML Model	Cross-Design	Features	Objective
IncPIRD	Static IR	2019	XGBoost	Yes	I_g, R_g, PDN, G	IR mitigation
XGBIR		2020	XGBoost	-	I_g, R_g, PDN	PDN design
Yamato *et al.*	Vector-based IR	2012	Linear Regression	No	P_c	IR-aware timing
Dhotre *et al.*		2017	Clustering	Yes	r_{tog}, c	IR prediction
Lin *et al.*		2018	ANN	No	P_c, R_c, t_c, r_{tog}, c	IR mitigation
Fang *et al.*		2018	CNN, XGBoost	No	P_c, I_c, R_c, t_c, r_{tog}, c	IR mitigation
PowerNet	Vectorless IR	2020	CNN	Yes	P_c, t_c, r_{tog}	IR mitigation
Mozaffari *et al.*	Sillicon PSN	2019	ANN, CNN, NLP	Yes	r_{tog}, G	PSN prediction

$\{I_c, I_g\}$ Current:

- I_c : Average or peak current measured on each **c**ell instance
- I_g : Current loads or the total current on power **g**rids

$\{R_c, R_g\}$ Resistance:

- R_c : resistance on the path from power pad to each **c**ell instance
- R_g : resistance measured on power **g**rids and power nodes

Methods: IR Drop Estimators Overview

Methods	Type of IR Drop	Time	ML Model	Cross-Design	Features	Objective
IncPIRD	Static IR	2019	XGBoost	Yes	I_g, R_g, PDN, G	IR mitigation
XGBIR		2020	XGBoost	-	I_g, R_g, PDN	PDN design
Yamato *et al.*	Vector-based IR	2012	Linear Regression	No	P_c	IR-aware timing
Dhotre *et al.*		2017	Clustering	Yes	r_{tog}, c	IR prediction
Lin *et al.*		2018	ANN	No	P_c, R_c, t_c, r_{tog}, c	IR mitigation
Fang *et al.*		2018	CNN, XGBoost	No	P_c, I_c, R_c, t_c, r_{tog}, c	IR mitigation
PowerNet	Vectorless IR	2020	CNN	Yes	P_c, t_c, r_{tog}	IR mitigation
Mozaffari *et al.*	Sillicon PSN	2019	ANN, CNN, NLP	Yes	r_{tog}, G	PSN prediction

{PDN}: The information about PDN

$\{P_c\}$ Power: Power dissipation of each cell instance (internal, switching, leakage)

$\{G\}$ Global information: process, voltage, temperature, frequency, layout size

$\{c\}$ Cell information: cell area, cell load, cell type

Methods: IR Drop Estimators Overview

Methods	Type of IR Drop	Time	ML Model	Cross-Design	Features	Objective
IncPIRD	Static IR	2019	XGBoost	Yes	I_g, R_g, PDN, G	IR mitigation
XGBIR		2020	XGBoost	-	I_g, R_g, PDN	PDN design
Yamato *et al.*	Vector-based IR	2012	Linear Regression	No	P_c	IR-aware timing
Dhotre *et al.*		2017	Clustering	Yes	r_{tog}, c	IR prediction
Lin *et al.*		2018	ANN	No	P_c, R_c, t_c, r_{tog}, c	IR mitigation
Fang *et al.*		2018	CNN, XGBoost	No	P_c, I_c, R_c, t_c, r_{tog}, c	IR mitigation
PowerNet	Vectorless IR	2020	CNN	Yes	P_c, t_c, r_{tog}	IR mitigation
Mozaffari *et al.*	Sillicon PSN	2019	ANN, CNN, NLP	Yes	r_{tog}, G	PSN prediction

$\{r_{tog}\}$ Toggling activity

- Switching activity of each cell. Usually measured by toggle rate.

$\{t_c\}$ Timing window

- The timing interval of switching for each cell. Min/max signal arrival time.

Methods for Static IR Drop

- Similarities between **IncPIRD** & **XGBIR**:
  - ML model: both use XGBoost

Features in **IncPIRD**

Category	Description of Feature on node n
Chip / PDN	Pitch of all metal layers Width / height of the chip
Electrical	*Pullup*: The effective resistance at n *Pulldown*: The symbolic IR drop at n Pullup and Pulldown of n's neighbors

Features in **XGBIR**

Category	Description of Feature on node n
Chip / PDN	The number of power tracks Distance between n and boundary Power track segmental resistance
Electrical	*Pullup*: Voltage sources' impact on n *Pulldown*: Current loads' impact on n *V2I*: Resistance between voltage sources and current loads

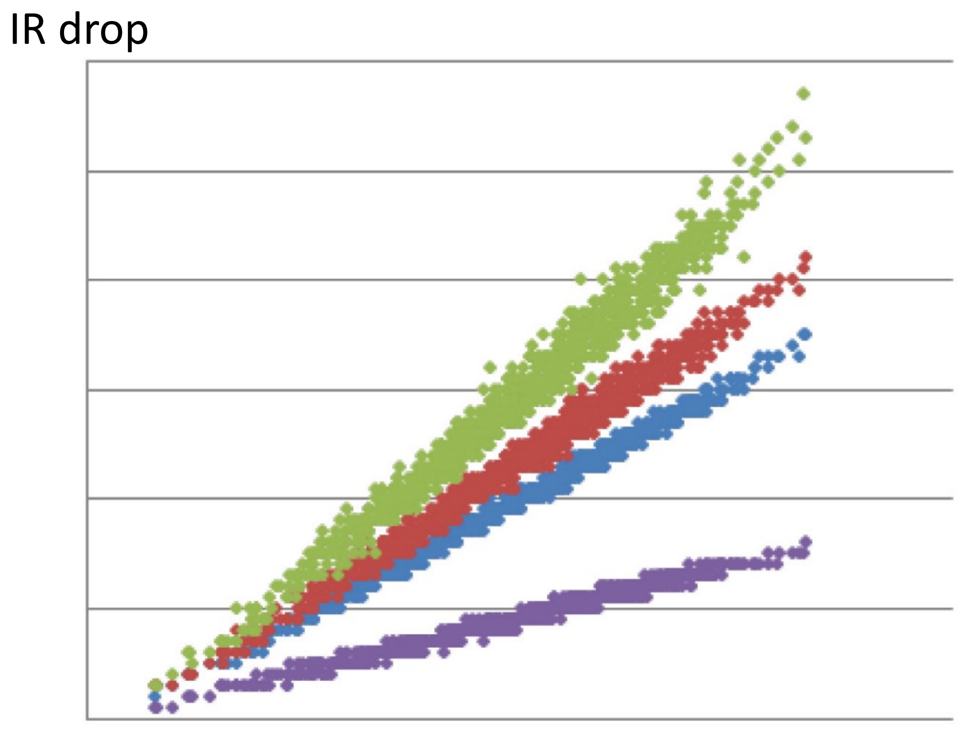
- Special property of **IncPIRD** :
  - Model used for iterative PDN design
  - Use 'update condition' to decide whether model needs to be retrained

IncPIRD: Ho, *et al*., ICCAD 19
XGBIR: Pao, *et al*., DATE 20

Dynamic IR Drop with Power Only [*]

- A linear model for each cell instance
  - IR drop = k * power + b
  - Captures correlation between IR drop and power

- Discussion:
  - Impact from neighboring cells not captured.



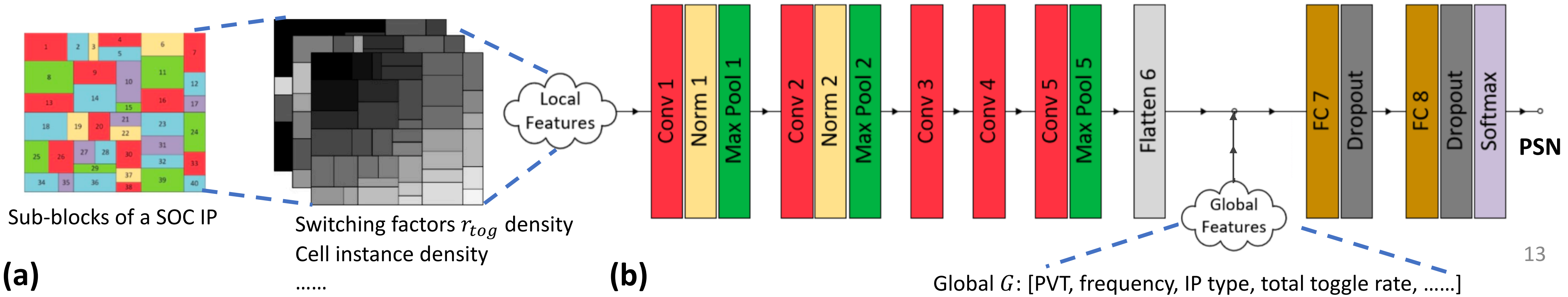
Power vs. IR-drop on cells with high correlation. [*]

[*] Yamato, *et al.*, ITC 12

Dynamic IR Drop with Spatial Info [*]

- Predict the overall PSN of the whole SOC
- Method:
  - Input (local) features: Togging rate density map + cell density map
  - Use CNN to capture **spatial information** at local region
  - Global information about the whole chip provided in FC layers
- Discussion:
  - **Timing information** not incorporated

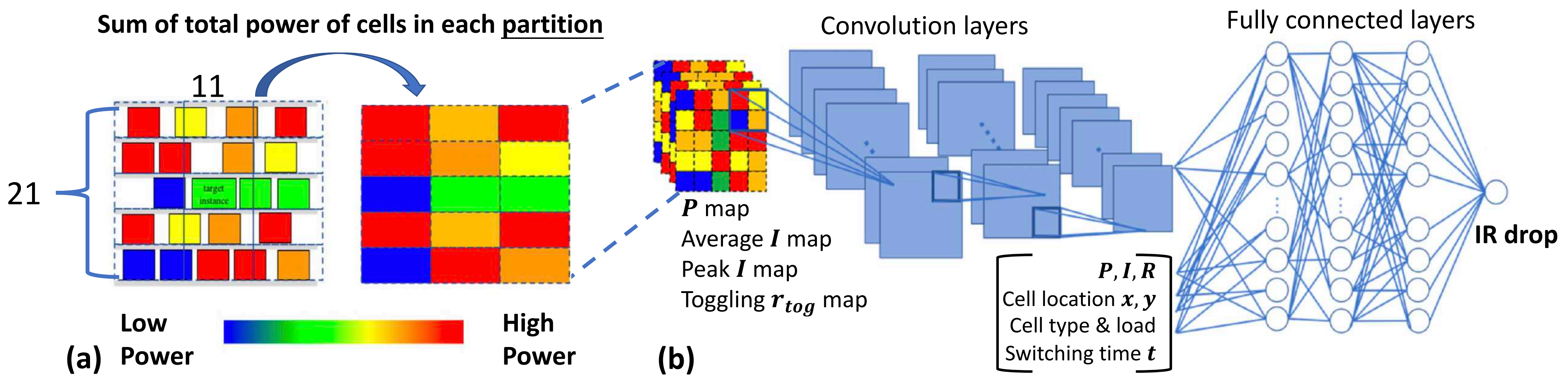
[*] Mozaffari, *et al.*, ITC 19



Dynamic IR Drop with Spatial & Timing Info [*]

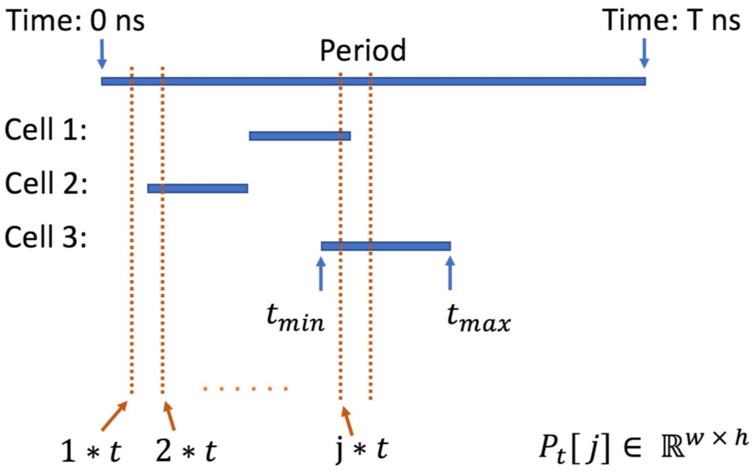
- Predict the IR drop of each cell, considering timing info
- Method:
  - Local current & power maps around the target cell as 2D input
  - Timing window & location of each cell directly provided in 1D input
  - Compared with Mozaffari, *et al.*, similar 2D & 1D inputs through CNN

[*] Fang, *et al.*, ICCAD 18

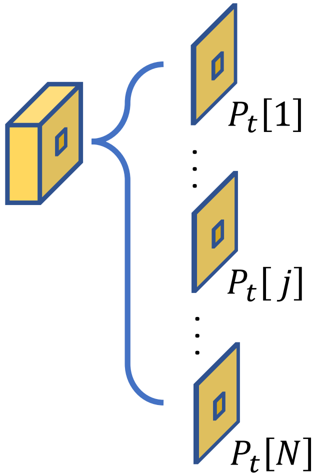


Dynamic IR Drop with Spatial & Timing Info [*]

- Predict IR drop of each grid. Capture the time instant with peak IR.
- Method (features):
  - Power maps as features $P_{internal}$, $P_{switching}$, P_{total}, ……
  - Divide clock cycle into N time instants.
  - For each grid, measure time-decomposed power maps $P_t[1]$–$P_t[N]$ around it
  - Only the cells that can switch at instant j contribute to $P_t[j]$



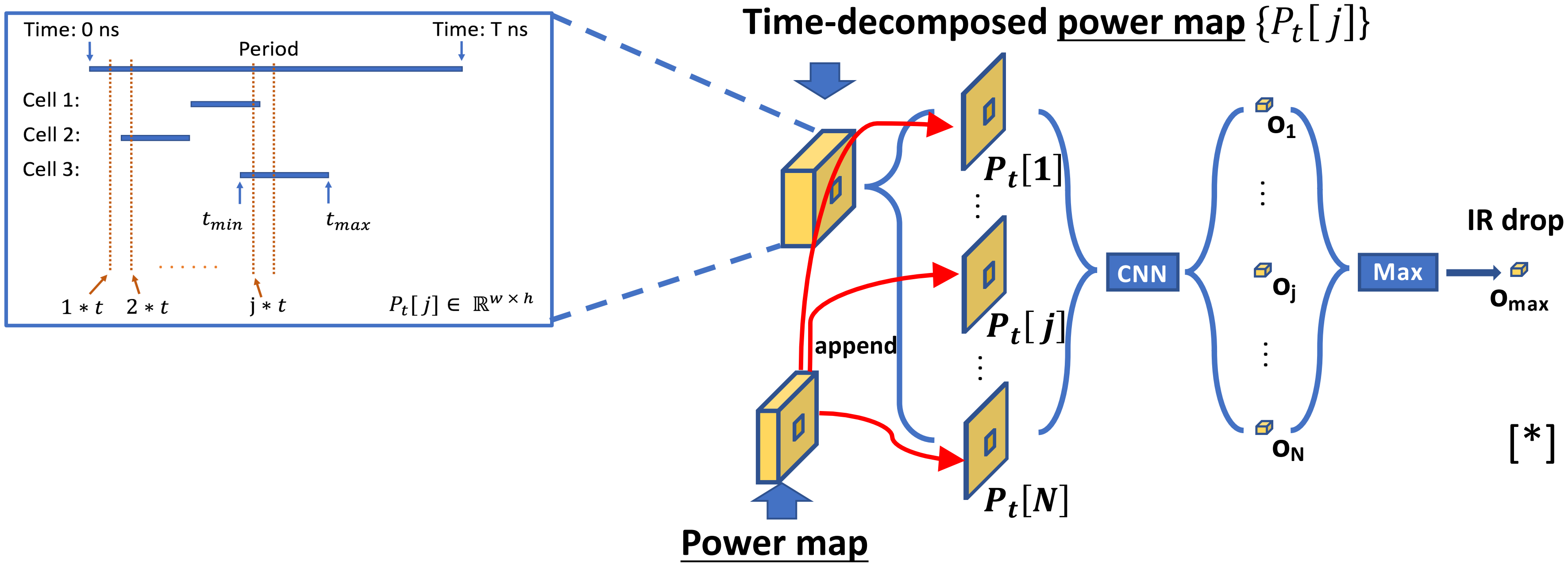
Time-decomposed **<u>power map</u>** $\{P_t[j]\}$



[*] Xie, *et al.*, ASPDAC 20

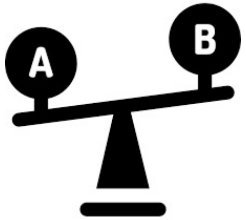
Dynamic IR Drop with Spatial & Timing Info [*]

- Predict IR drop of each grid. Capture the time instant with peak IR.
- Method (model):
  - Use the same CNN model to process all N input features $P_t[1]$–$P_t[N]$ in parallel
  - Generate N outputs $o_1 - o_N$, corresponding to N transient IR drops
  - Take the $\mathbf{max}([o_1 : o_N])$ as estimated IR. It measures the highest transient IR



[*] Xie, *et al.*, ASPDAC 20

Challenges & Future Works

- Challenge 1: **Evaluation & comparison** 

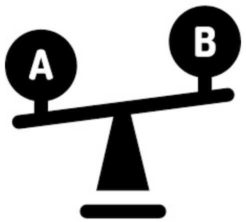
- Challenge 2: **Human effort** 

- Challenge 3: **Robustness & generalization** 

Challenges & Possible Solution

- Challenge 1: **difficulty in <u>evaluation & comparison</u> among models**
  - For static IR drop: IncPIRD vs XGBIR?
  - For dynamic IR drop: Fang *et al.* vs PowerNet vs …?
  - Similar for other tasks: routability, parameter tuning, …….

- Possible solution: open-sourced benchmark for ML applications
  - Designs & flows dedicated to ML applications on multiple design objectives
  - Benefit:
    - Enable rapid and clear comparisons
    - Ensure high-quality training & validation data
    - Relieve researchers from data generation



Challenges & Possible Solution

- Challenge 2: **model development & maintenance take <u>human effort</u>**
  - Estimators tuned for both feature selection and model architecture
  - Estimators may vary for different dataset & application scenarios
  - Tuning heavily rely on human expertise.

- Possible solution: search ML algorithms automatically
  - Search both appropriate features combination and ML model structure
  - Automated machine learning (AutoML)
  - NAS (Neural Architecture Searching)

Challenges & Possible Solution

- Challenge 3: **model's <u>robustness & generalization</u> not verified**
  - How model will perform on previously unseen data
  - Not likely to perform well on every new cases (designs/technology nodes)
  - Risky every time when inferring a new test case.

- Possible solution: measure model's robustness before inference
  - Quantify similarity between training & testing cases
  - Example: the 'update condition' in IncPIRD, deciding whether ML model needs to be retrained [1]

[1] Ho, *et al*., ICCAD 19

Conclusion

- We summarize the latest progress in ML models for fast IR drop estimation
- We introduce the innovations and technical details of representative methods
- We discuss some general challenges in ML estimators

Thank You!
Questions?