

COMPUTER ARCHITECTURE AND SYSTEMS LAB

I3D VLSI Laboratory

NPUWattch: ML-based Power, Area, and Timing Modeling for Neural Accelerators

Sehyeon Kim[1], Minkwan Kim[1], Chanho Park[1],
Hanmok Park[2], Seonghoon Kim[2], Taigon Song[2], and William J. Song[1]

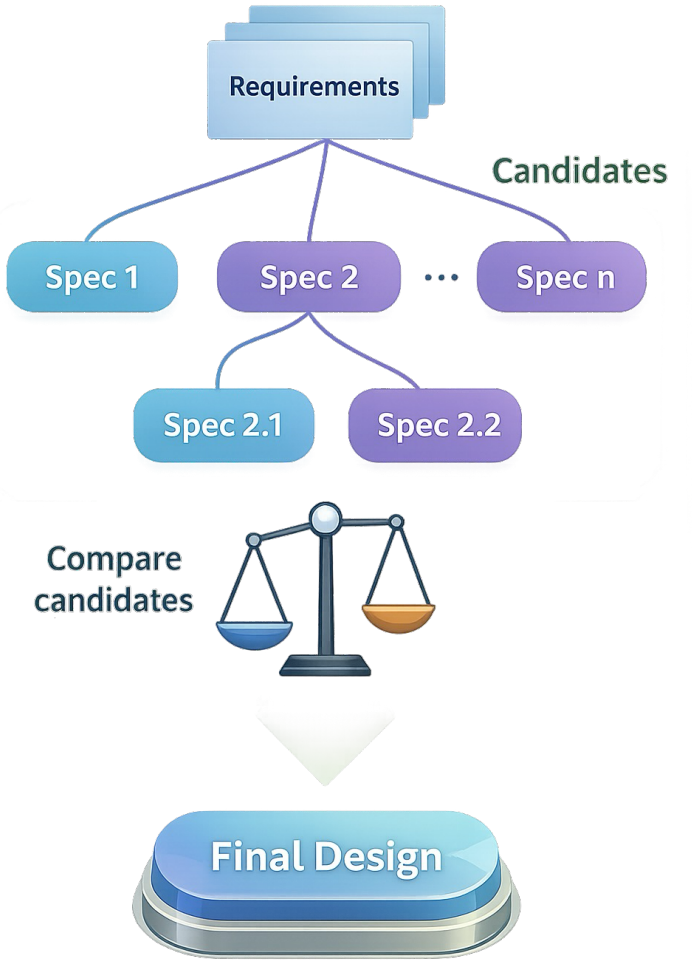
[1] School of Electrical and Electronic Engineering, Yonsei University

[2] School of Electronic and Electrical Engineering, Kyungpook National University

Content

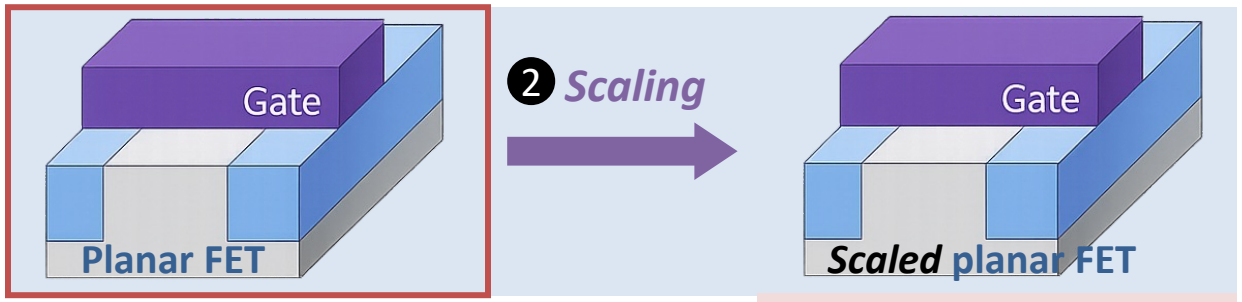
- **Introduction:** Modeling Neural Accelerators
- **Background:** Common Practice for Scaling Design Costs
- **Motivation:** Three Pitfalls in Accelerator Modeling
- **NPUWattch:** Framework Workflow, Overall Design Procedure
- **Evaluation:** Experiment Setup, Modeling Accuracy
- **Conclusion**

Introduction: Modeling Neural Accelerators



- **Neural accelerators** have become one of the major architectural research topics.
- Full **tape-out** design procedure
  - ➡ Too costly for **early design space exploration** (DSE)
- **Power, area, and timing (PAT) models**
  (Examples: CACTI, Aladdin, Accelergy)
  - ➡ **Efficiency** and **accuracy** tradeoff
- Traditional **rule-based models** often lack sufficient details.
  - ➡ **Inaccurate** pre-silicon design analysis

Background: Common Practice for Scaling Design Costs



❶ Design cost *measured* at an old process node *(e.g., 45nm)*

❸ Design cost *estimated* at a new process node *(e.g., 10nm)*

!= Mismatch

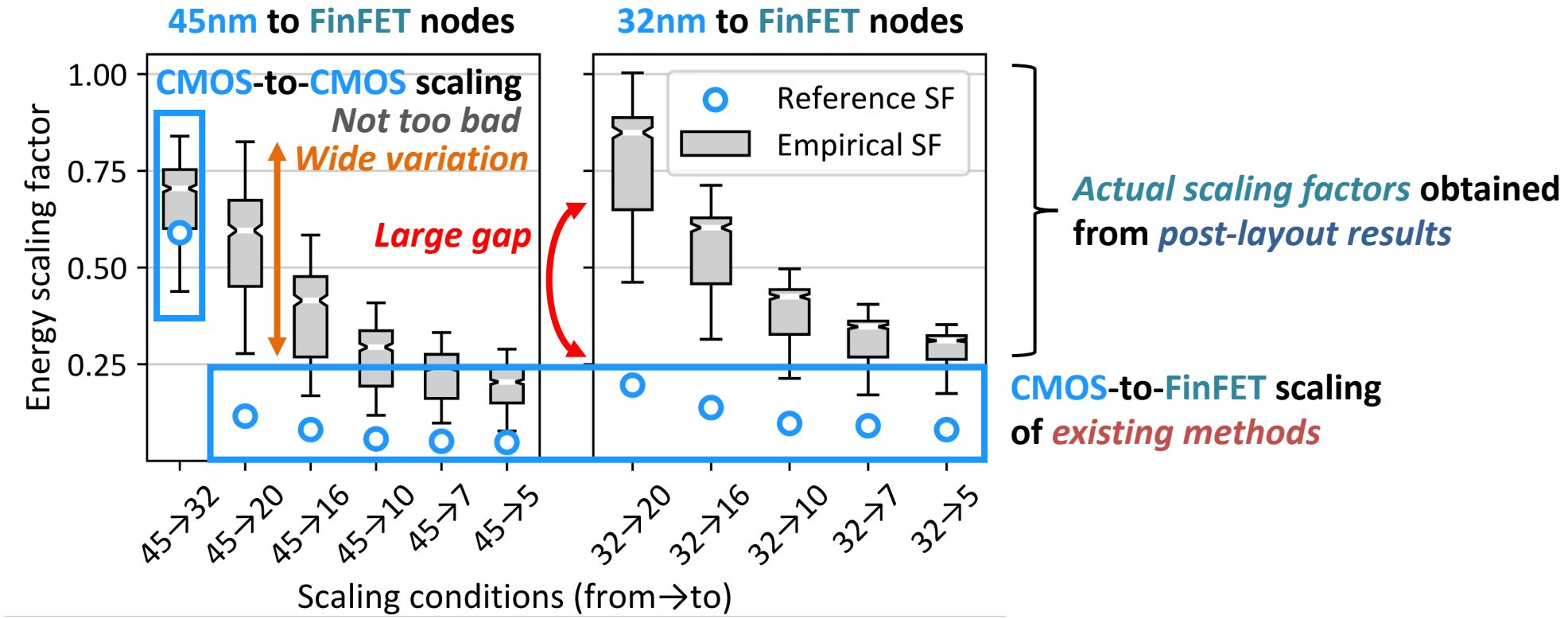
❹ *Actual* design cost at the new process node

Simply scaling *CMOS design costs* to *emerging FinFET/GAAFET nodes* does not work!

Motivation: Three Pitfalls in Accelerator Modeling

Pitfall A. ***Technology Scaling Factors: One Size Never Fits All***

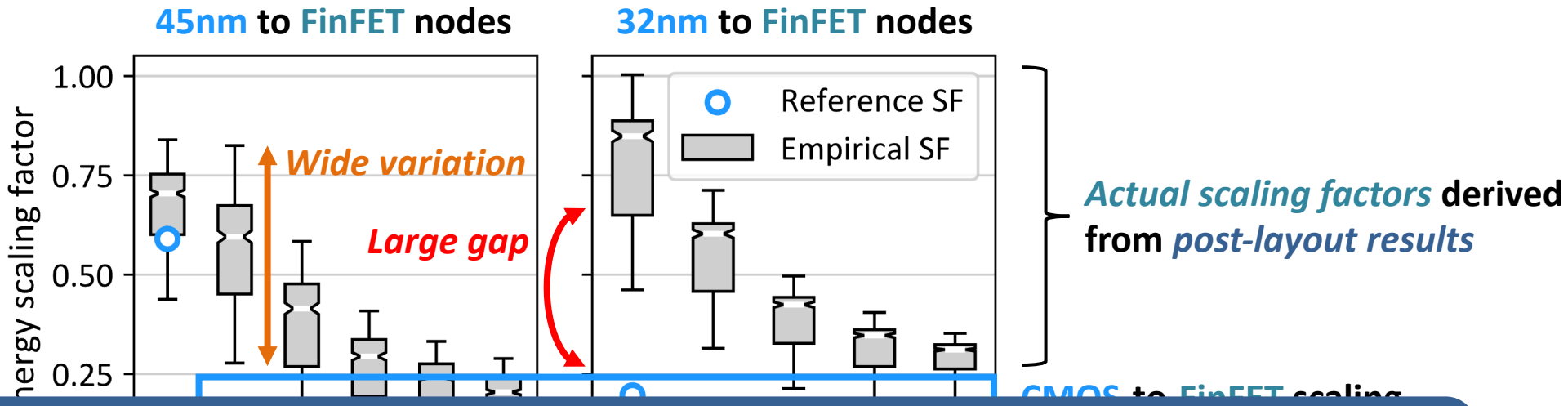
- A *single scaling factor* derived from a *small reference circuit* does not provide accurate estimations for *complex designs* at a *new technology node*.



Motivation: Three Pitfalls in Accelerator Modeling

Pitfall A. ***Technology Scaling Factors: One Size Never Fits All***

- A *single scaling factor* derived from a *small reference circuit* does not provide accurate estimations for *complex designs* at a *new technology node*.



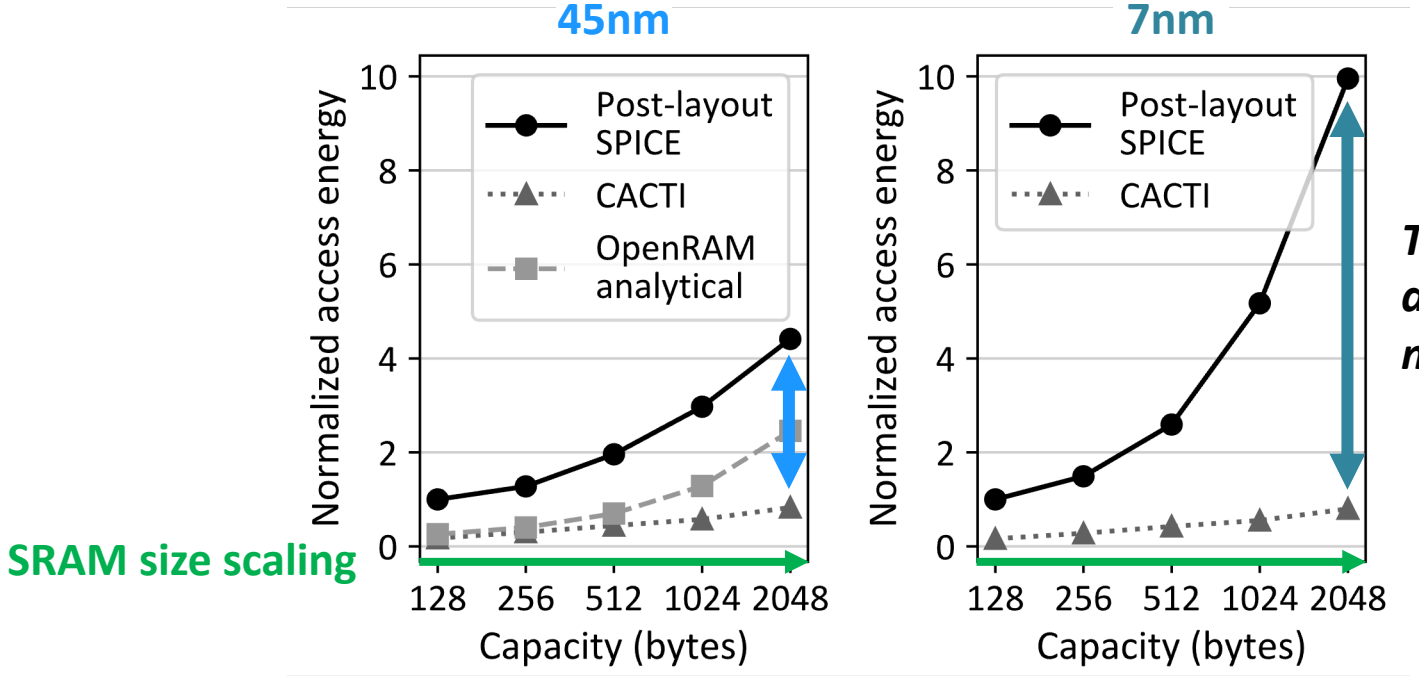
A *single scaling factor* cannot correctly represent *design variations* and technology scaling to *newly emerging process nodes*.

Motivation: Three Pitfalls in Accelerator Modeling

Pitfall B. ***Nonlinearity in Design Scaling***

- Traditional *analytical* or *rule-based models* fail to capture *nonlinearity* in design scaling.

  ➡ Even identical designs *do not scale the same way* at different technology nodes.



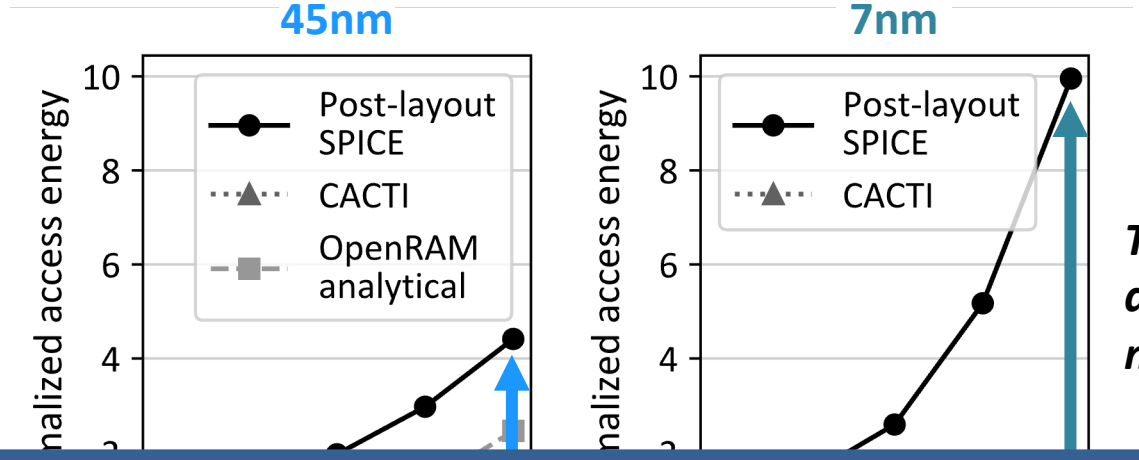
The difference between post-layout results and rule-based models (e.g., CACTI) is more pronounced in the advanced nodes.

Motivation: Three Pitfalls in Accelerator Modeling

Pitfall B. ***Nonlinearity in Design Scaling***

- Traditional *analytical* or *rule-based models* fail to capture *nonlinearity* in design scaling.

  ➡ Even identical designs *do not scale the same way* at different technology nodes.



The difference between post-layout results and rule-based models (e.g., CACTI) is more pronounced in the advanced nodes.

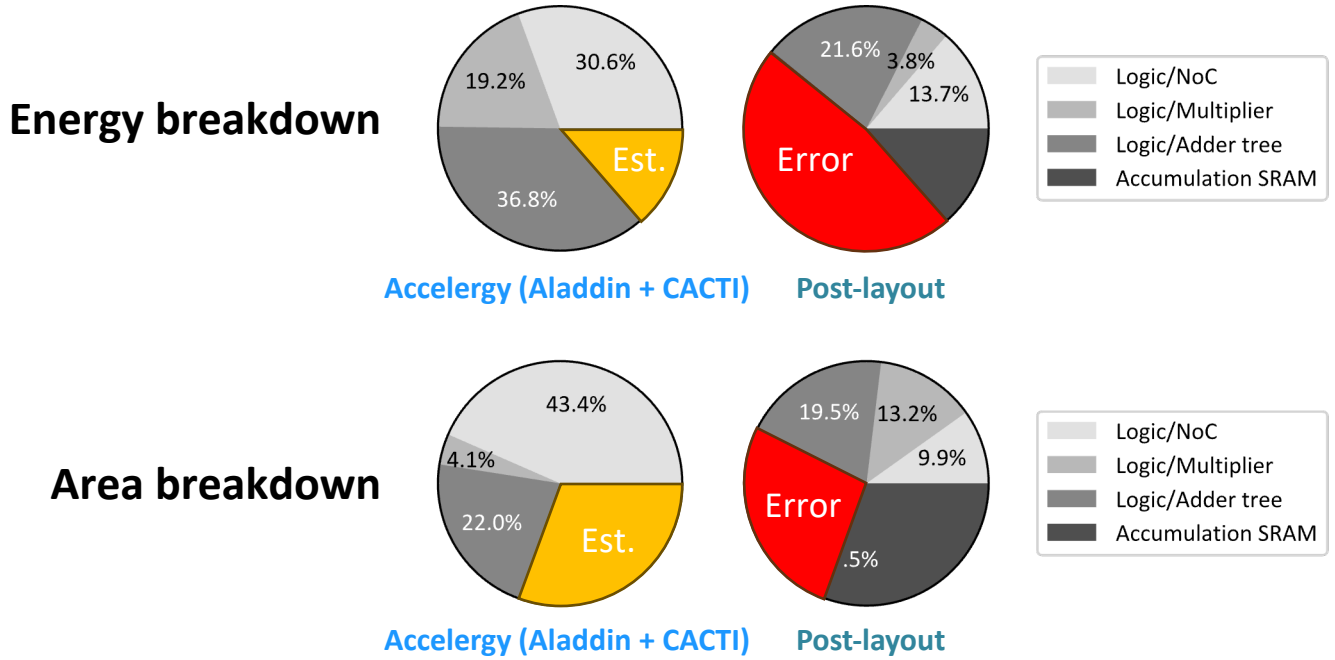
Traditional rule-based models **cannot correctly capture** ***nonlinearity*** **in design and technology scaling.**

Motivation: Three Pitfalls in Accelerator Modeling

Pitfall C. ***Inconsistency in Heterogeneous Modeling***

- Monolithic-die modeling requires a *common calibration point* (i.e., transistor model) between **logic** and **SRAM** for *consistency*.

  ➡ The mixed use of heterogeneous models (e.g., Aladdin + CACTI) introduces *bias*.



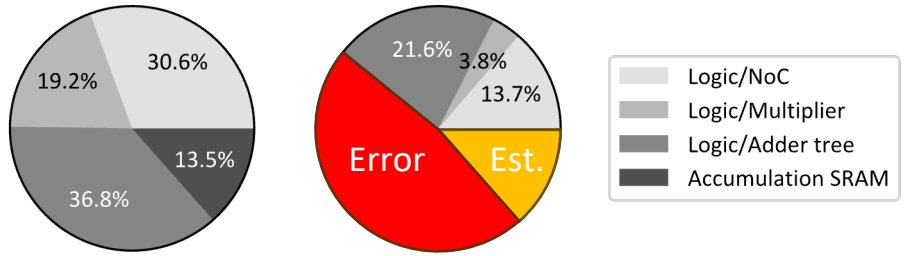
Motivation: Three Pitfalls in Accelerator Modeling

Pitfall C. ***Inconsistency in Heterogeneous Modeling***

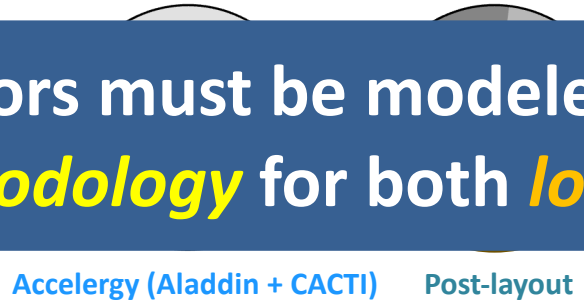
- Monolithic-die modeling requires a *common calibration point* (i.e., transistor model) between **logic** and **SRAM** for *consistency*.

  ➡ The mixed use of heterogeneous models (e.g., Aladdin + CACTI) introduces *bias*.

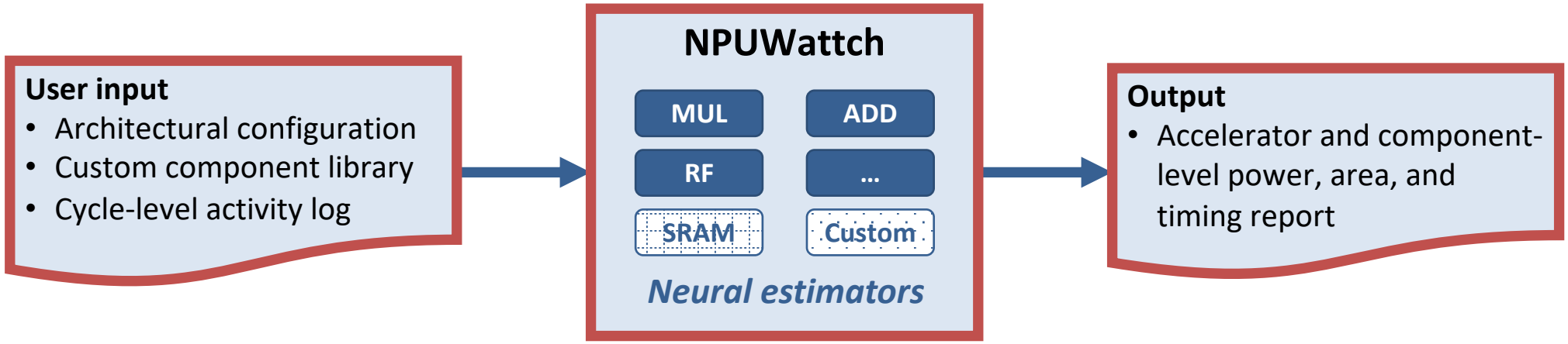
Energy breakdown



Neural accelerators must be modeled using a *coherent design methodology* for both *logic* and *SRAM*.

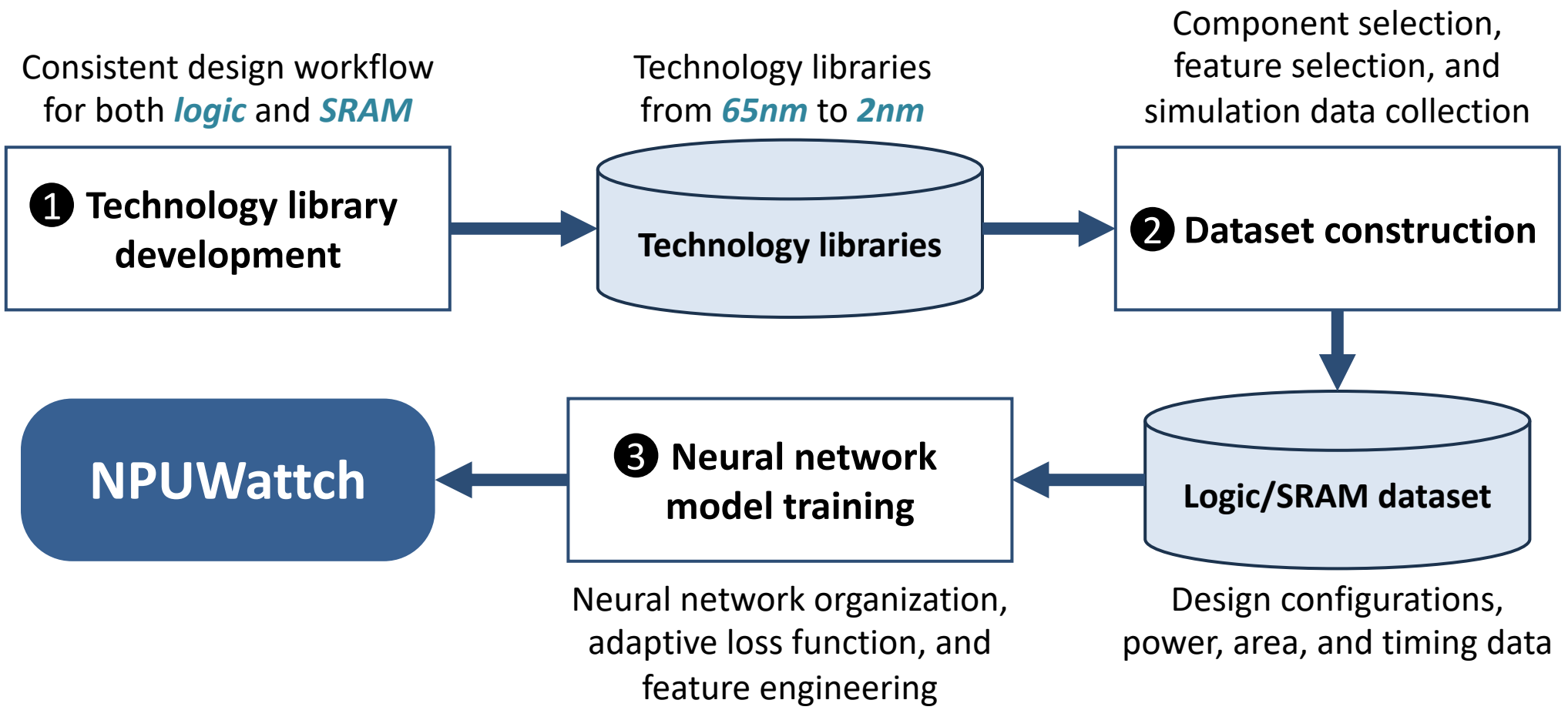


NPUWattch: Framework Workflow



NPUWattch provides accurate design cost estimates through *neural estimators* trained on carefully curated *post-layout data*.

NPUWattch: Overall Design Procedure

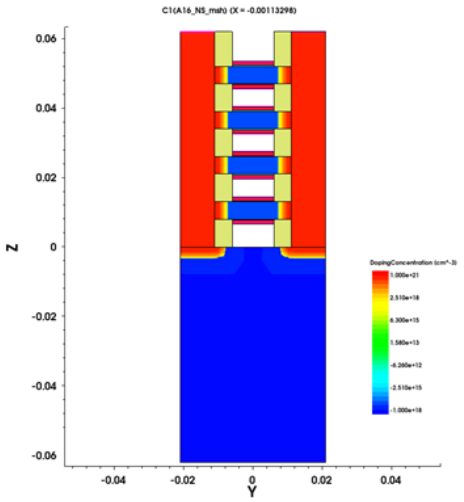


❶ Technology Library Development

Transistor Modeling:

- We *modeled 3D transistors* that conform to *scaling trends*, while considering doping concentration, thermal diffusion, and related physical effects.

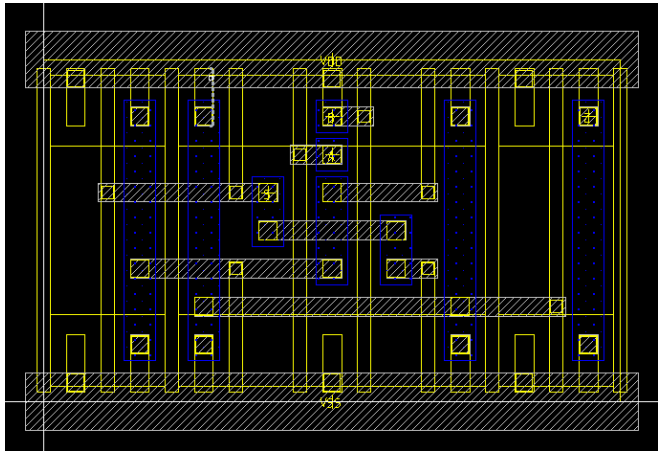
Doping Concentration Map



Technology Library Building:

- We developed *multi-corner*-characterized *technology libraries*, including design rules, logic cells, and SRAM periphery circuits.

Standard Cell Layout

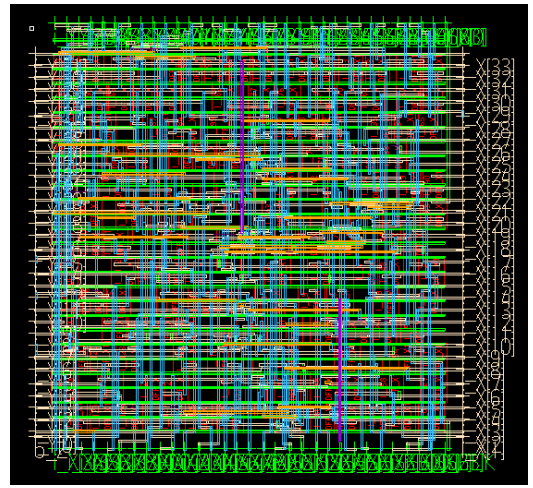


❷ Dataset Construction

Digital Logic:

- We performed *post-layout simulations* on functional RTL designs to extract power, area, and timing information.

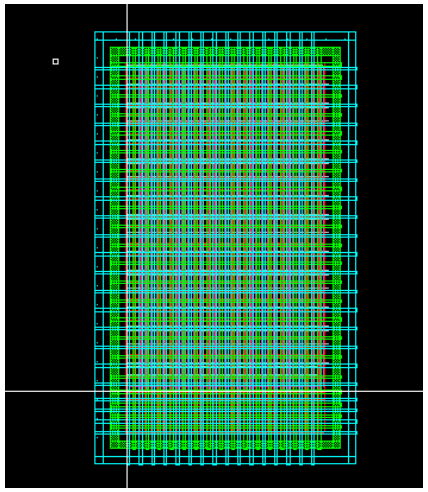
Digital Circuit Layout



SRAM:

- We compiled SRAM arrays of various sizes and performed *SPICE simulations* to evaluate their characteristics.

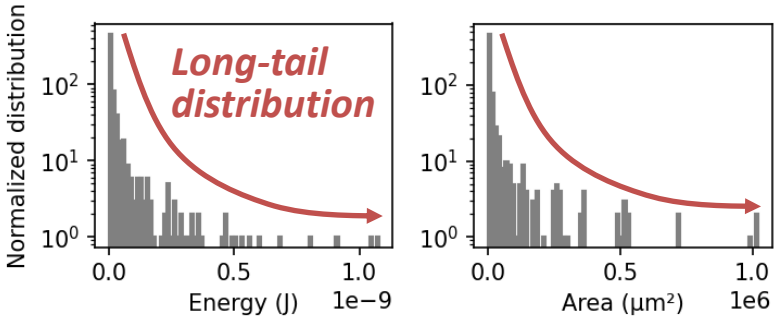
SRAM Array Layout



❸ Neural Network Model Training

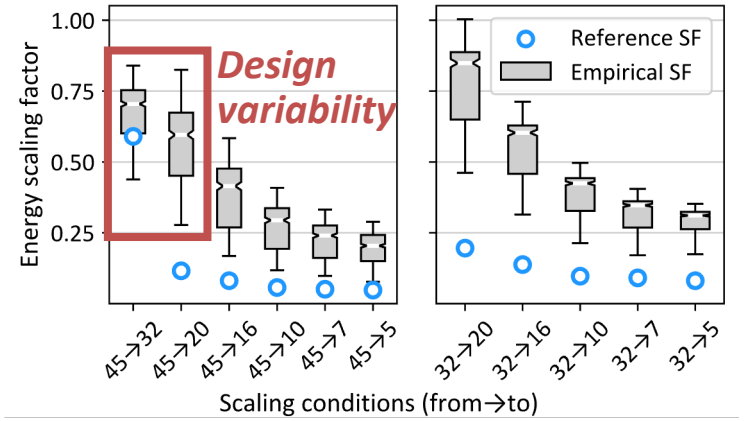
Neural Network Model:

- *Lightweight MLPs* for fast runtime inference (~0.4ms)
- A *dedicated neural network cluster* for each component



Neural Network Training:

- An *adaptive loss function*, which reweights uneven sample distributions to address the *long-tail distribution problem*
- New *design-aware input features** to address *variability* in technology scaling.

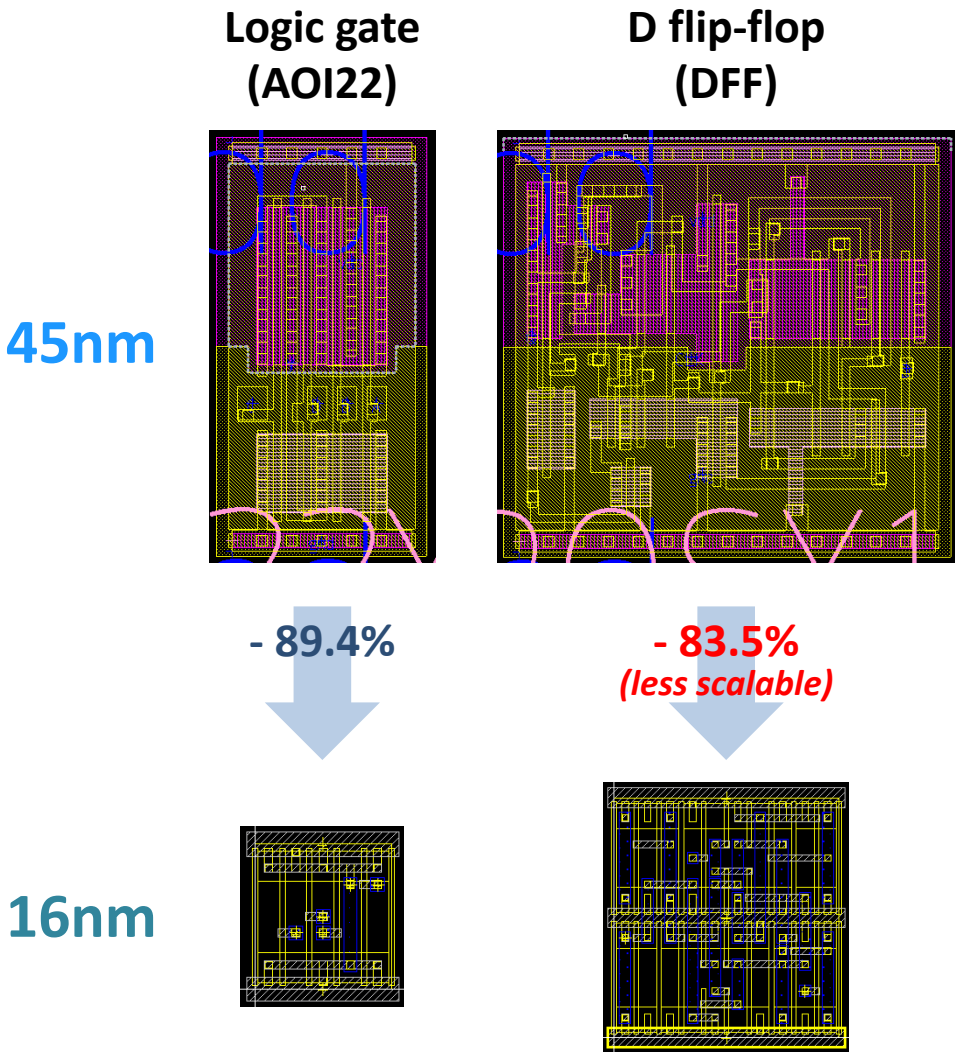


Neural Network Modeling and Training

Design-Aware Input Features:

- NPUWattch introduces two new *design-aware input features*, ***S**equential cell **C**ount **R**atio (**SCR**)* and ***S**equential cell **A**rea **R**atio (**SAR**)*.

- This approach is motivated by the observation that ***flip-flops*** exhibit different power and area scaling than ***logic gates*** due to *clock trees*, *latch analog effects*, and *wiring complexity*.



Experimental Setup

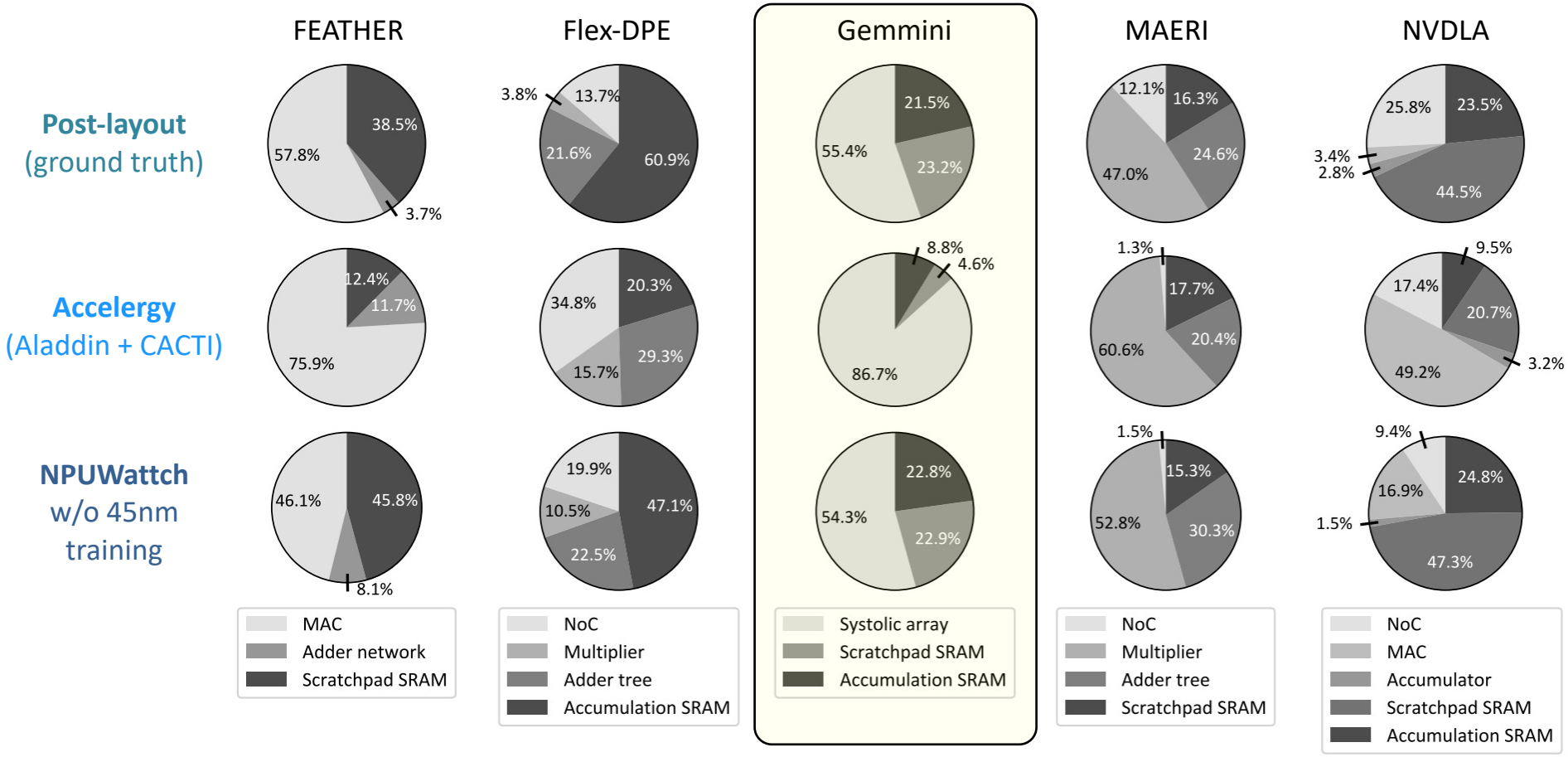
- **Modeling Frameworks:**
  Accelergy (Aladdin table + CACTI) at 45nm vs. **NPUWattch** without 45nm training
- **Post-layout Simulation Setup:** Nangate 45nm cell library + OpenRAM with FreePDK45
- **Accelerator Designs:** five *open-source RTL* accelerators

Accelerator	Data format	PEs	Network topology	Global buffer
FEATHER	int8	16 x 16	Spatial/adder tree	256KB
Flex-DPE	bfloat16	1 x 128	Adder tree	128KB
Gemmini	int8	16 x 16	Systolic	320KB
MAERI	int16	1 x 128	Adder tree	80KB
NVDLA	int8	8 x 8	Adder tree	65KB

Evaluation Results: Comparative Analysis

Accelerator-Level Energy Breakdown at 45nm:

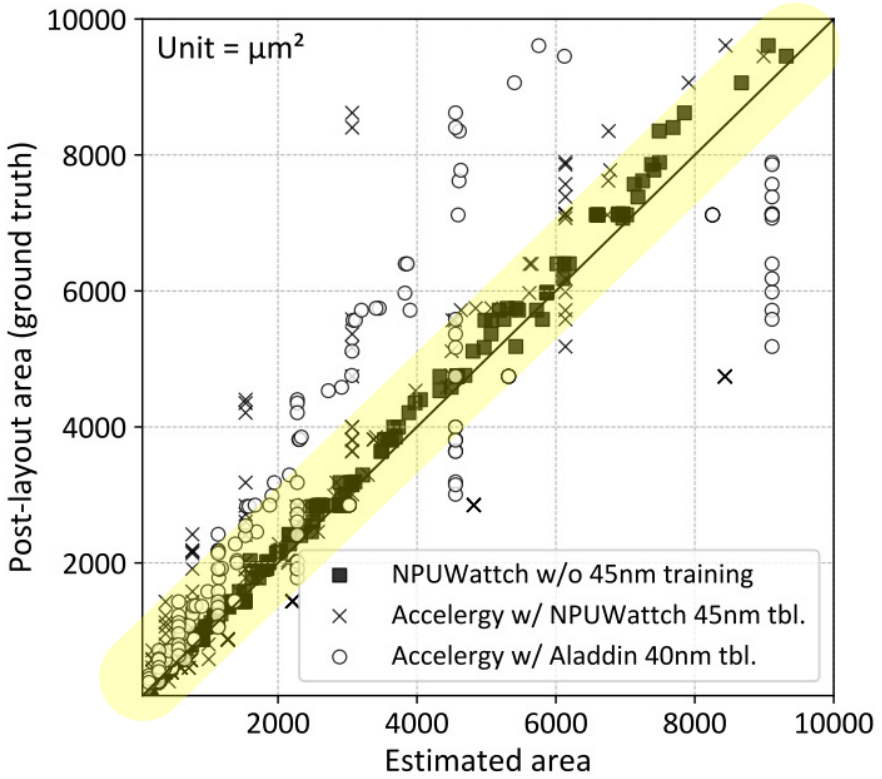
- *2.5x improvement* in mean absolute percentage error (MAPE) over Accelergy



Evaluation Results: Modeling Accuracy

Estimated vs. Ground Truth Area for Primitive Components:

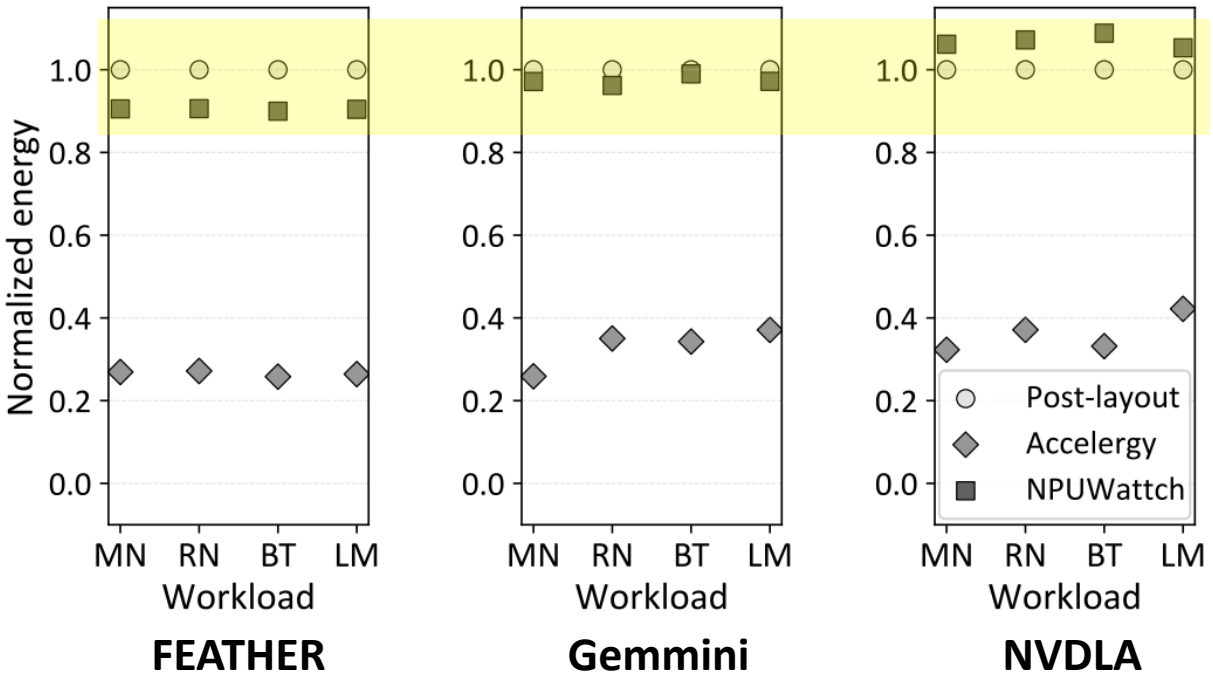
- *2.7% estimation error* on average
- NPUWattch (■) closely matches the post-layout data (diagonal line).



Evaluation Results: Workload-Level Analysis

Workload-Level Energy Estimation:

- *6.3% error* (NPUWattch) vs. *68.1% error* (Accelergy)
- NPUWattch (■) closely follows the post-layout results (●).



Networks	Abbr.	Features
MobileNetV3	MN	Depthwise convolution
ResNet-50	RN	Dense convolution
BERT	BT	Encoder-only transformer
Llama-3	LM	Generative AI

Conclusion

- Traditional *rule-based models* have *fundamental limitations* in capturing the *nonlinear design variability* and *scaling to sub-10nm technology nodes*.

- **NPUWattch**'s data-driven approach, trained on *post-layout data*, accurately models *complex structures* and scales to *emerging technology nodes* (e.g., 2nm).

🔗 **Repository:** github.com/yonseicasl/NPUWattch

🔗 **Tech Library:** i3dvlsi.wordpress.com/i3d-predictive-pdks/

 **Contact:** ikamusume@yonsei.ac.kr