DESIGN AUTOMATION CONFERENCE
JUNE 23-27, 2024
MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

COMPUTER ARCHITECTURE AND SYSTEMS LAB

Nona: Accurate Power Prediction Model Using Neural Networks

HoSun Choi, Chanho Park, Euijun Kim, and William J. Song

{**hosunhc**, ch.park, cwchris, wjhsong}@yonsei.ac.kr

School of Electrical and Electronic Engineering

Yonsei University



Power Modeling and Its Uses

- Diverse application behaviors and aggressive on-device power management make accurate power prediction more challenging.
- A power prediction model can be used for power management, benchmarking, code optimization, and more.
- Nona utilizes **performance monitoring counters (PMCs)** to predict the power usage of **real applications** on a **commercial device** (i.e., **Google Pixel 6**).

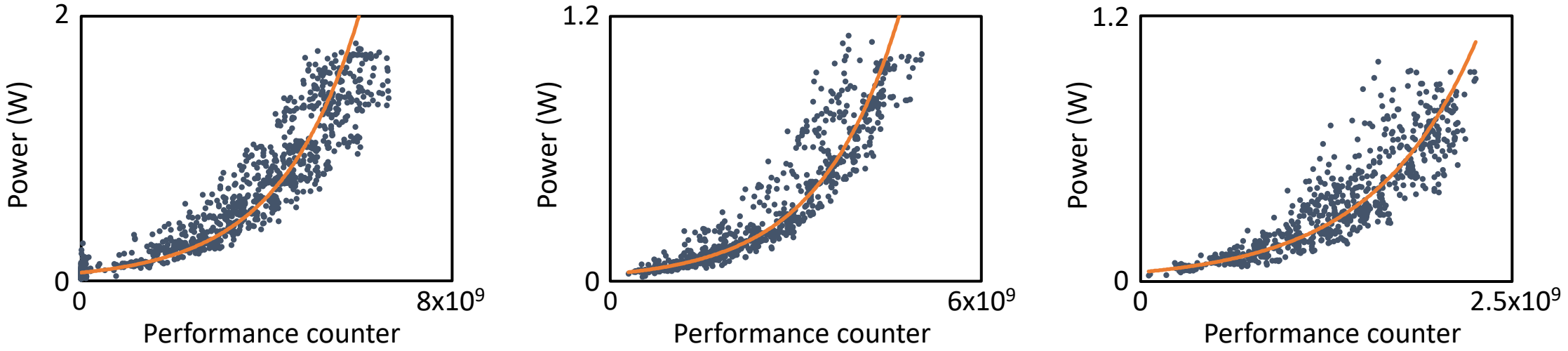


NextPit, “Free apps for Android & iOS: These Pro versions are currently free” [Online] https://www.nextpit.com/free-apps-of-the-week-29-2022-a

Relationship of PMCs and Power

- Previous power models were established based on the assumption that processor **power and PMCs are linearly correlated** (or at least can be modeled linearly).

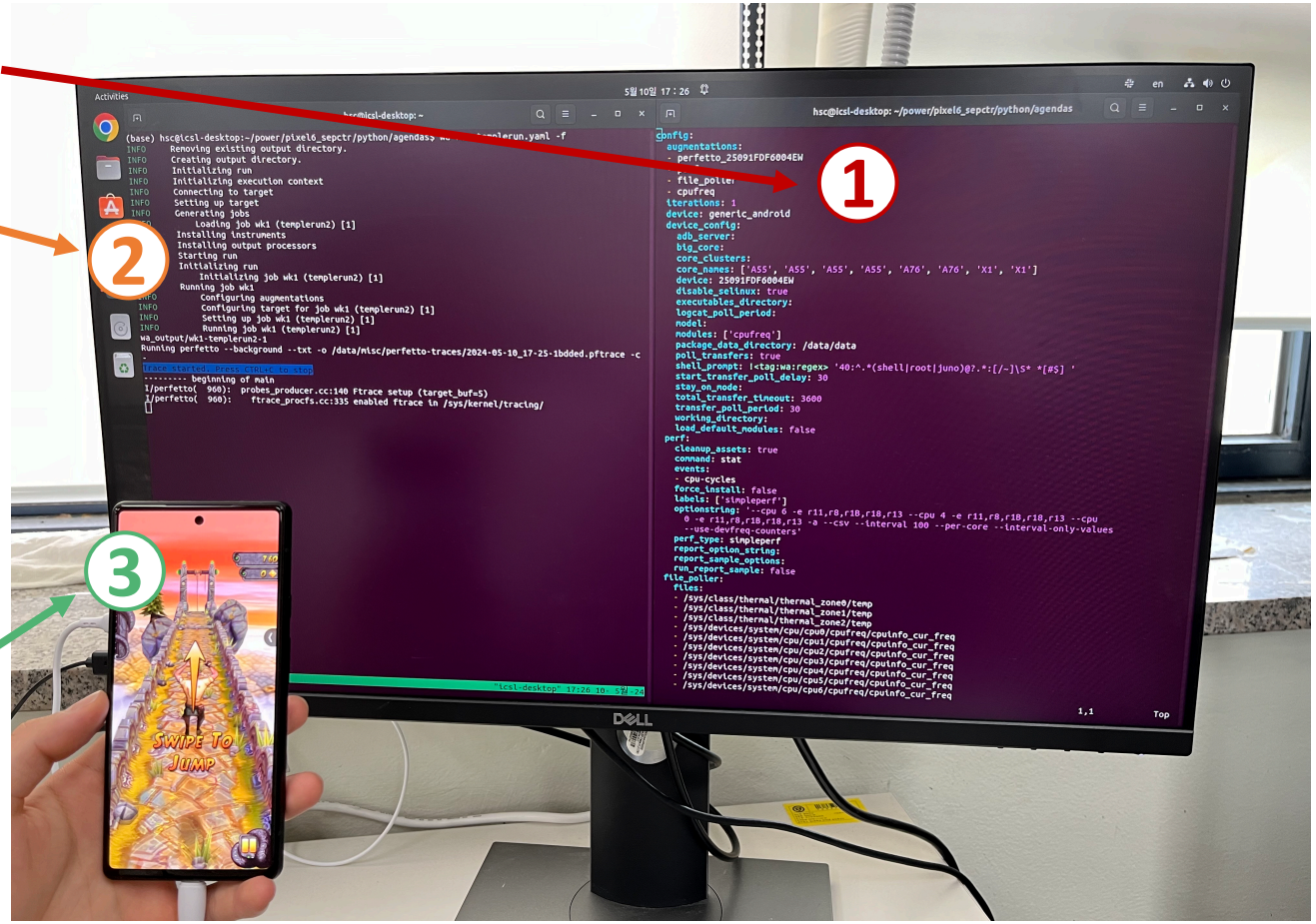
Measured power against PMC values (# retired instructions) for big, middle, and little clusters



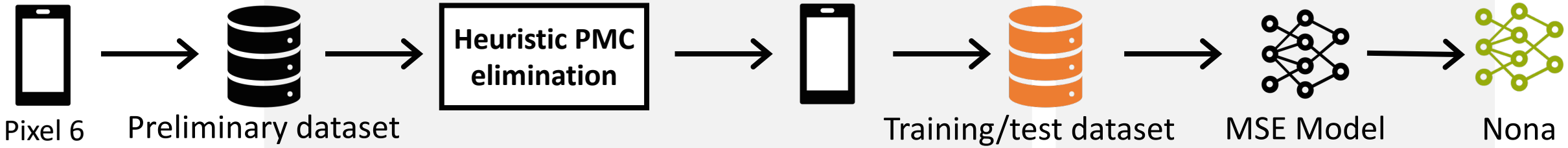
Neural networks can effectively capture nonlinear relations.

Experiment Setup

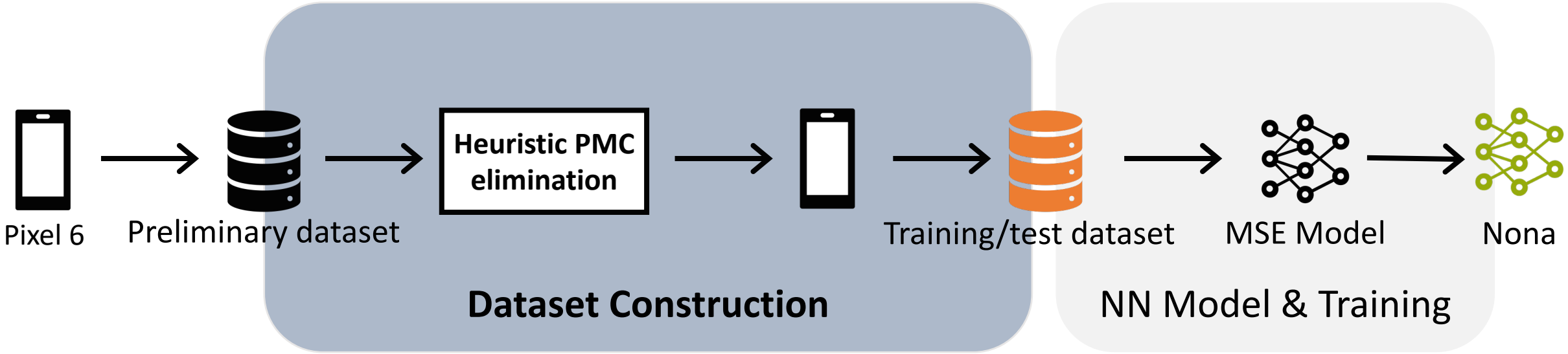
1. Creating an automation config file
2. ARM Workload Automation (WA) running the config file:
   - **Perfetto**: power measurement
   - **Simpleperf**: PMC monitoring
   - **File poller**: frequency, voltage, temperature measurement
3. Real mobile applications on Google Pixel 6



Overall Procedure



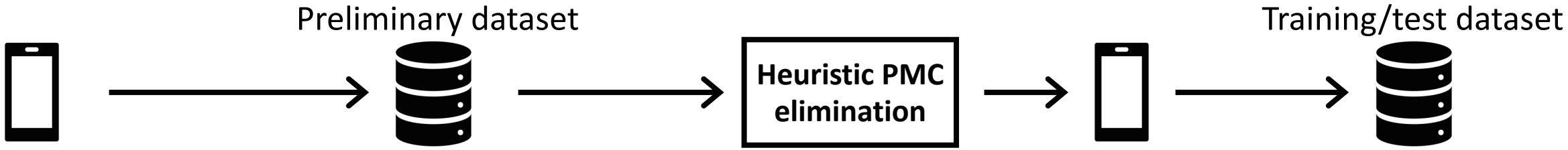
Overall Procedure



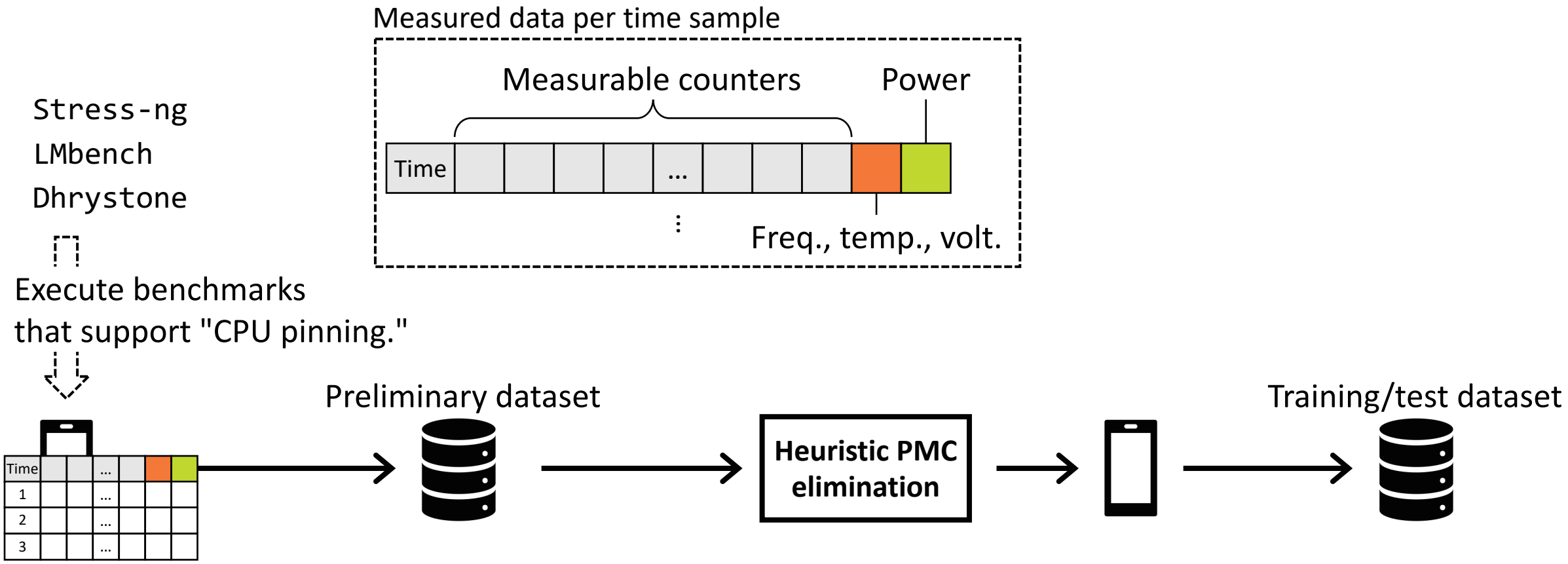
Dataset Construction

Challenge: A limited number of measurable PMC counters

- Each core can monitor only a **limited number of PMCs** simultaneously.
  - ARM Cortex-X1: **6 counters** at a time
  - Intel Skylake: **11 counters** at a time
- PMCs must be carefully selected to holistically and accurately capture diverse application behaviors between device power and the counters.

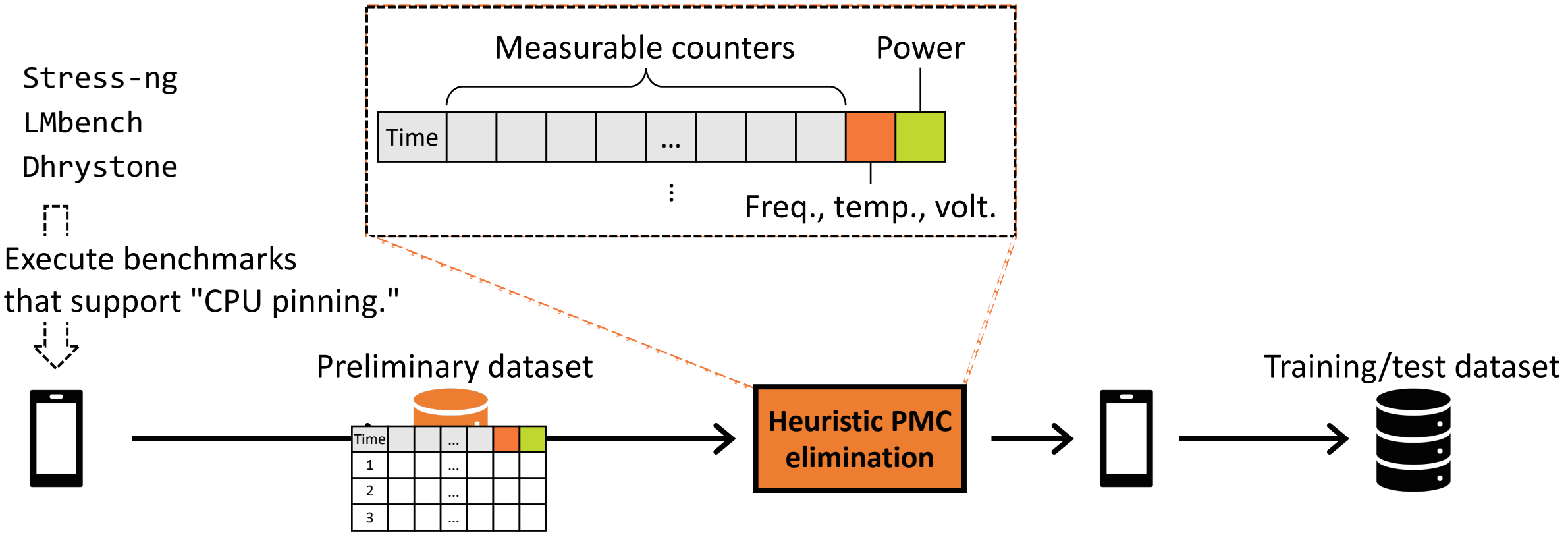


Dataset Construction



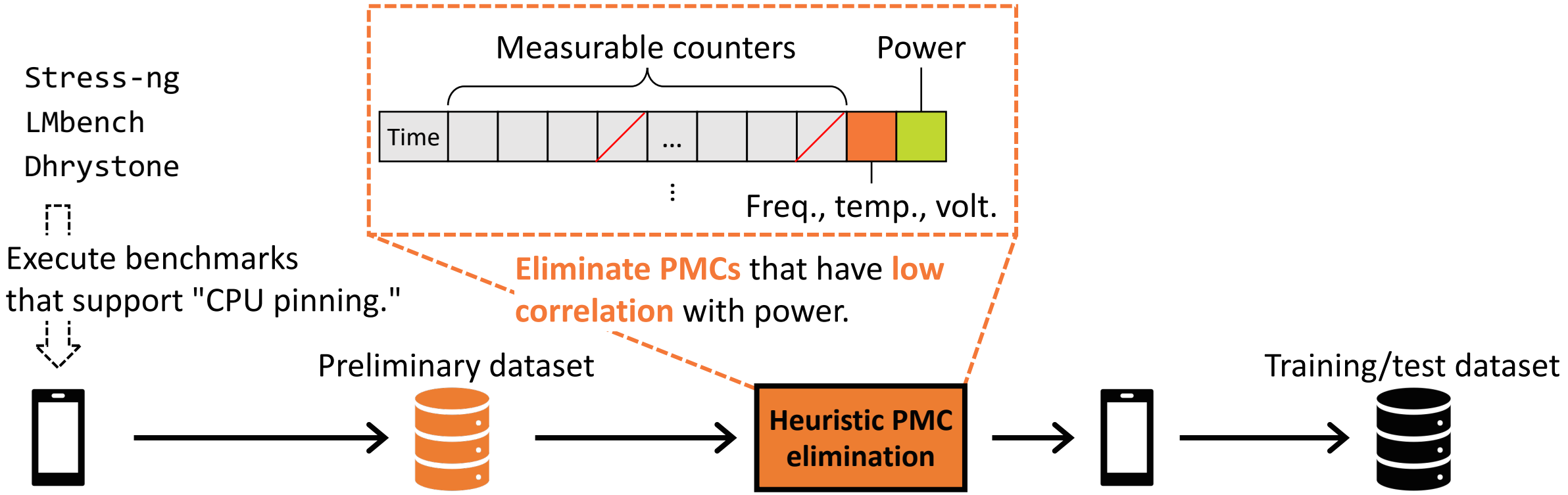
Stage 1. Constructing preliminary dataset

Dataset Construction



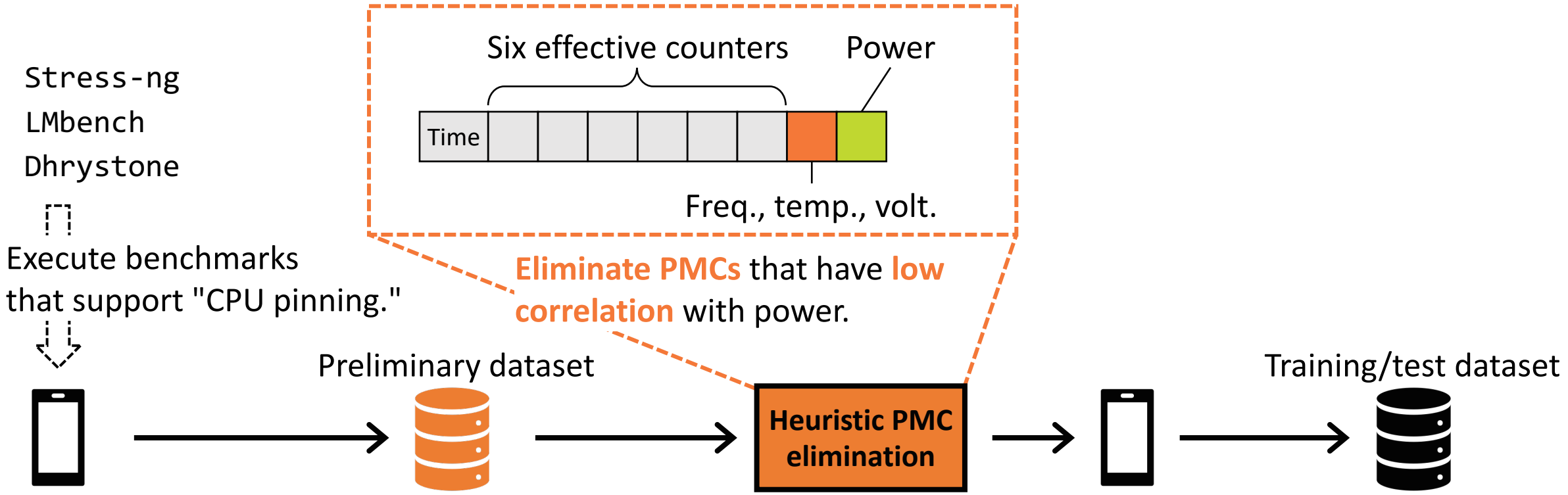
Stage 2. Selecting effective PMC counters

Dataset Construction



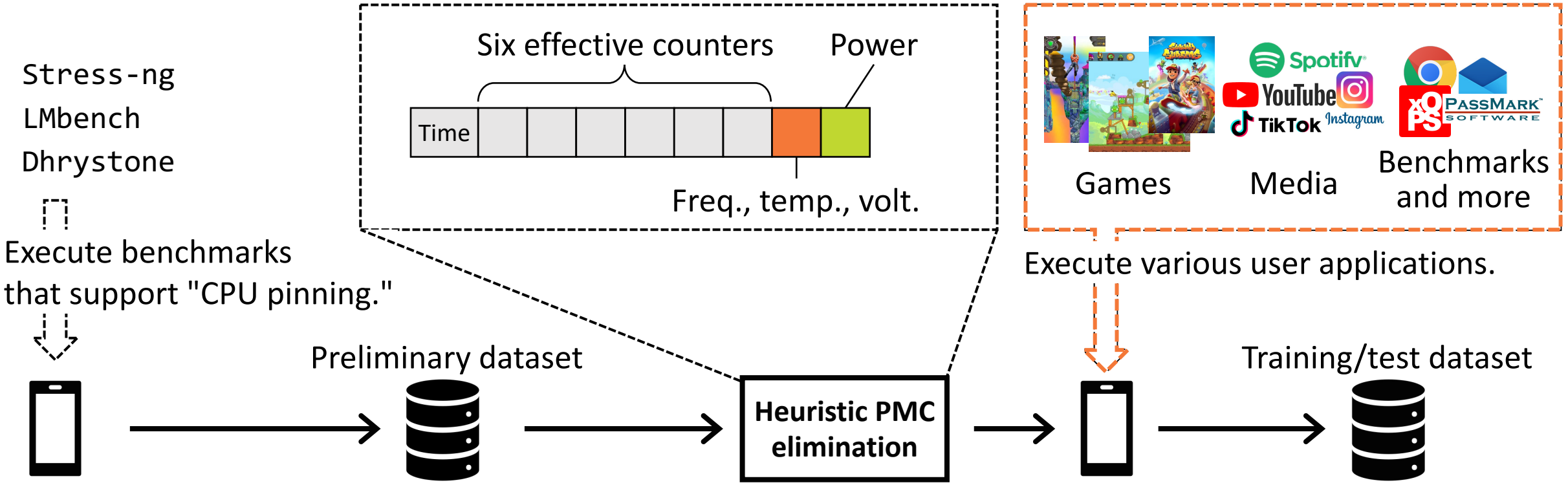
Stage 2. Selecting effective PMC counters

Dataset Construction



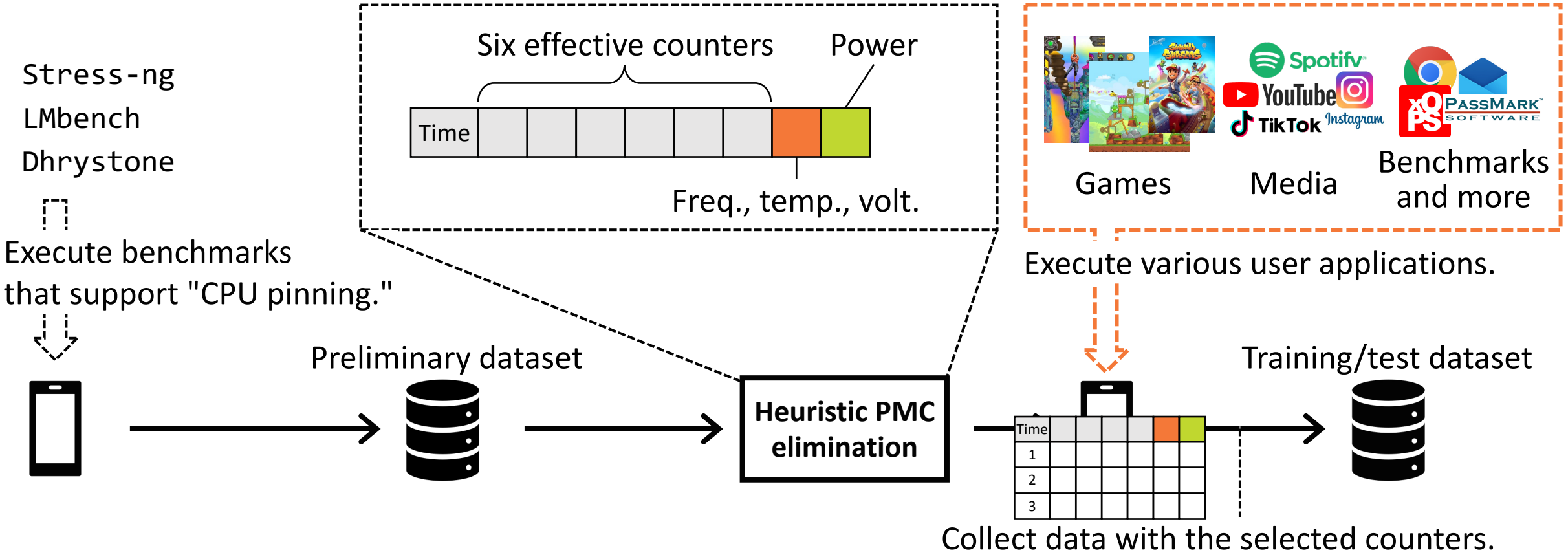
Stage 2. Selecting effective PMC counters

Dataset Construction



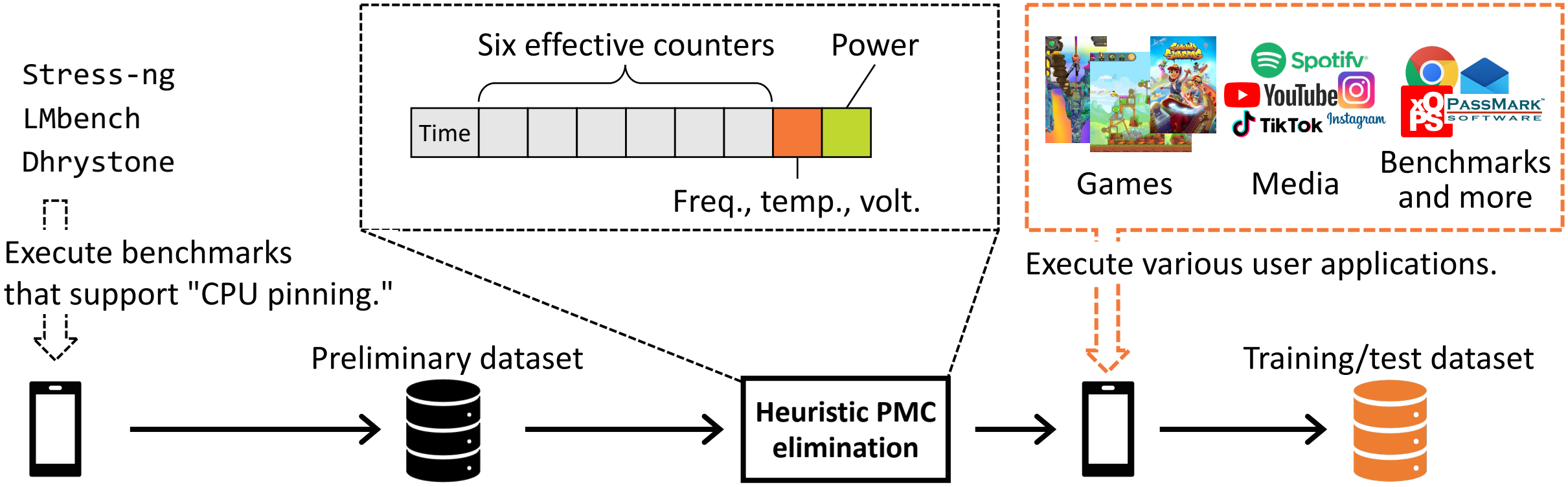
Stage 3. Constructing training/test datasets

Dataset Construction



Stage 3. Constructing training/test datasets

Dataset Construction

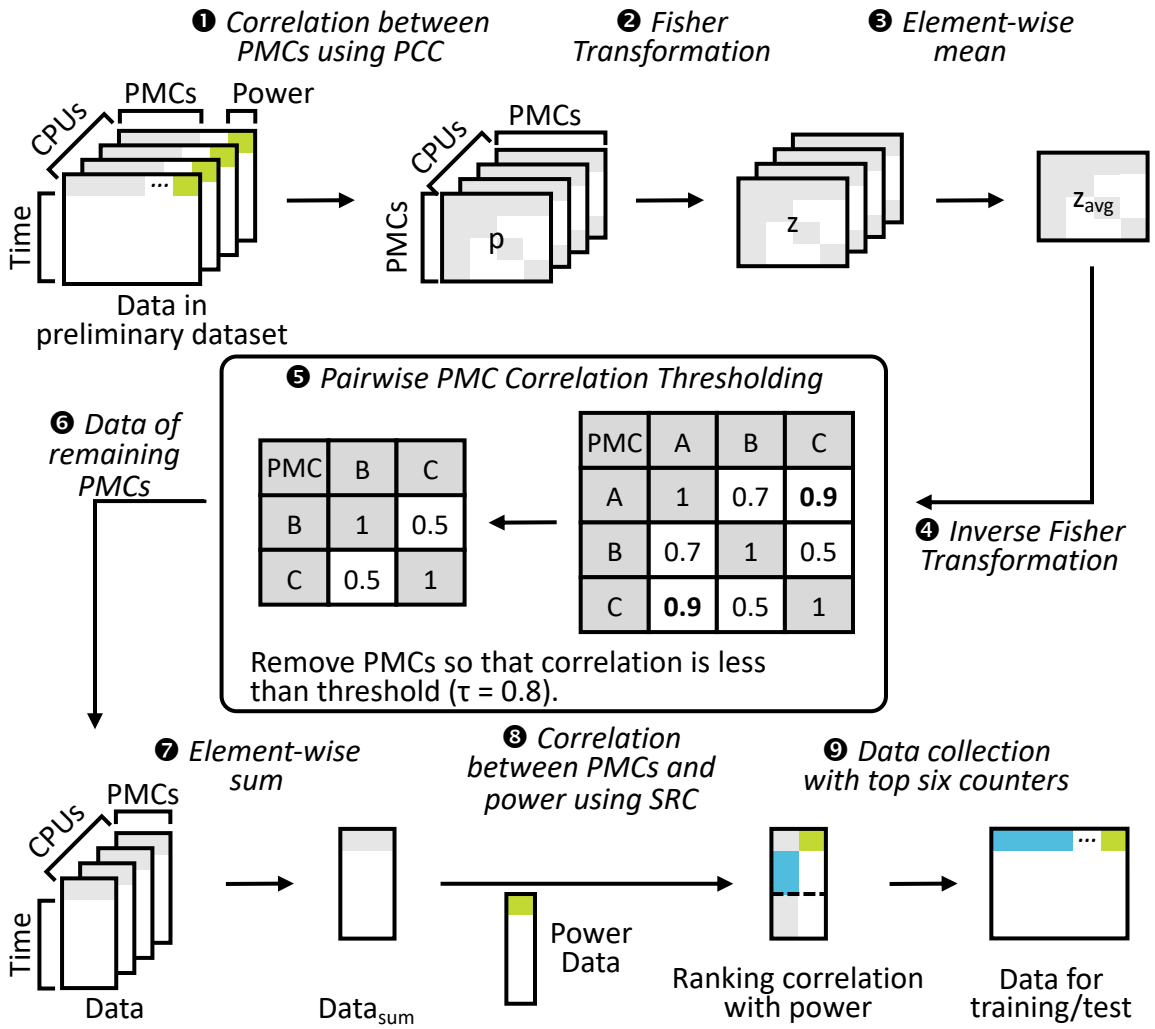


Stage 1. Constructing Preliminary dataset

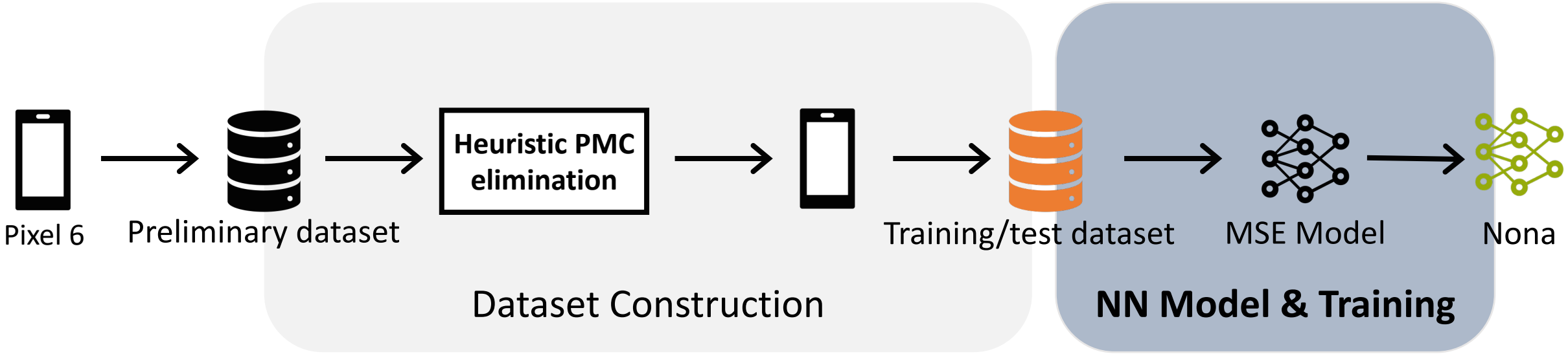
Stage 2. Selecting effective PMC counters

Stage 3. Constructing training/test datasets

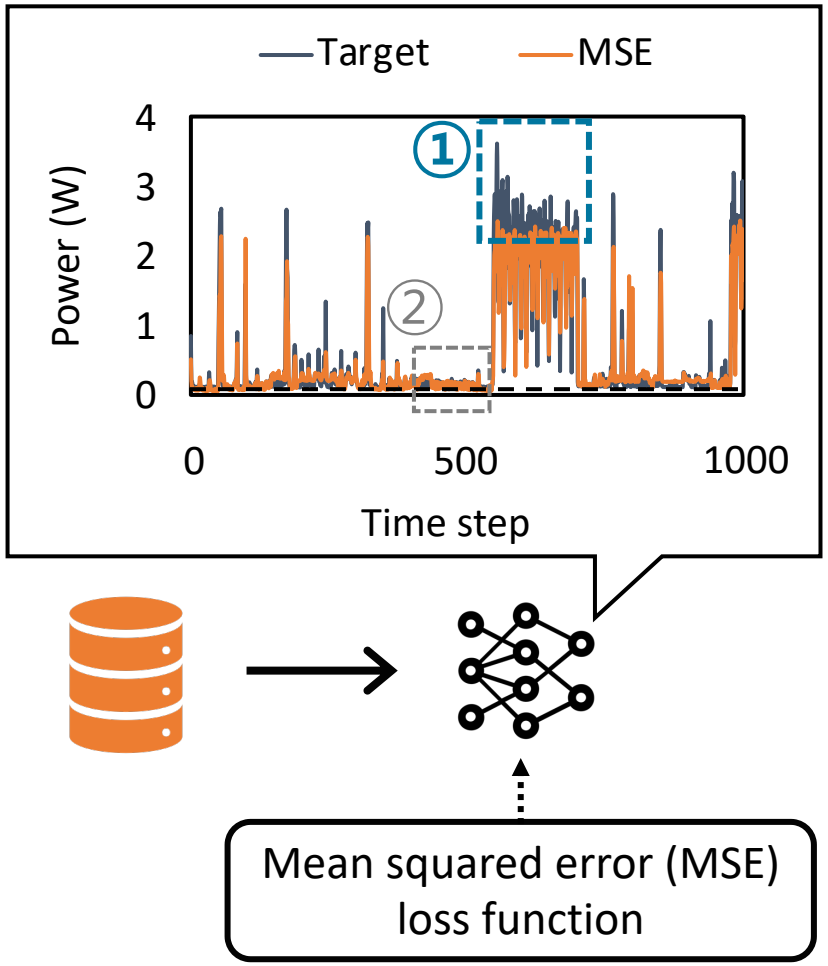
Dataset Construction: Heuristic PMC Elimination



Overall Procedure

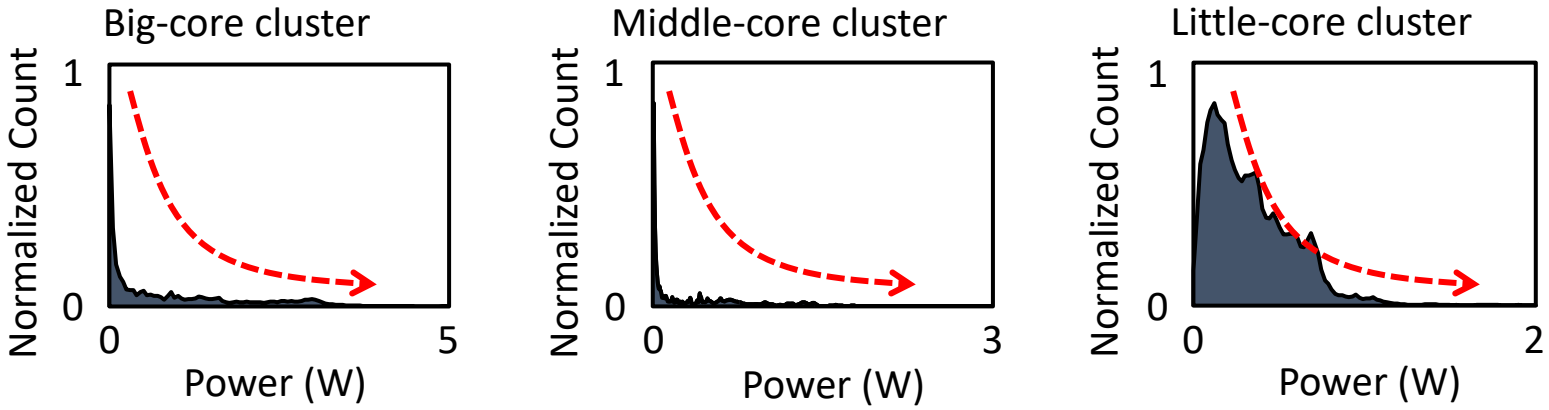


Neural Network: Data Imbalance

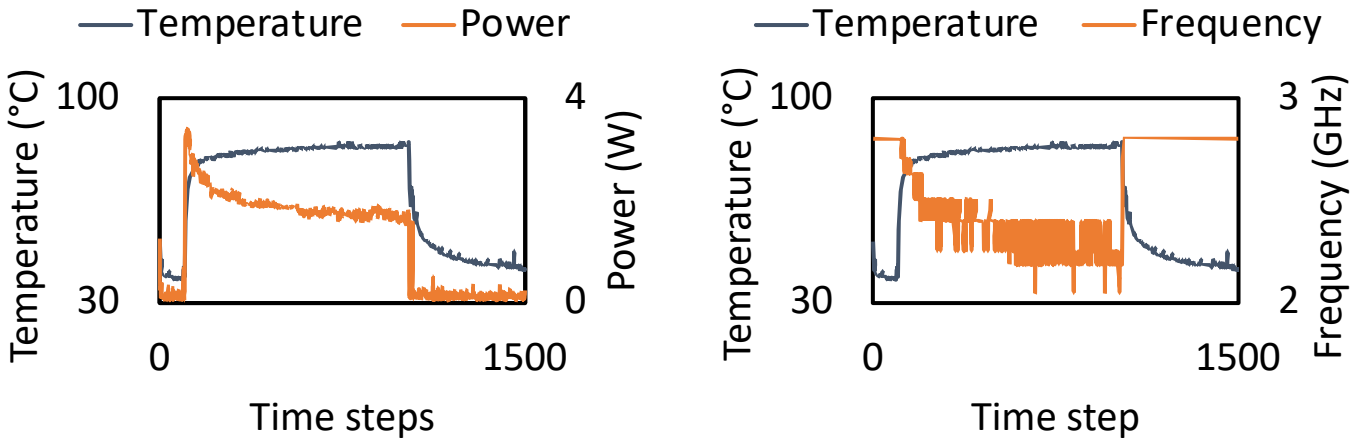


Challenge 1: Long tail distribution of collected dataset

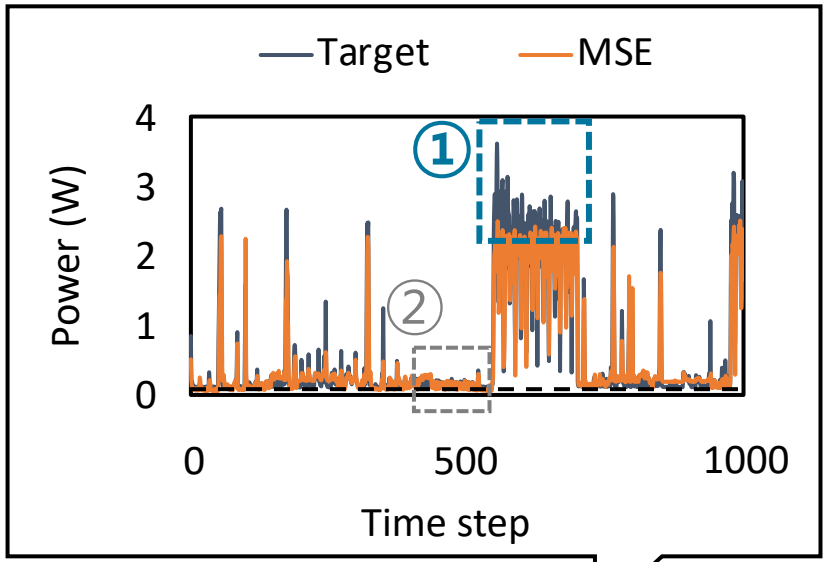
Long-tail sample distributions of power for each core cluster

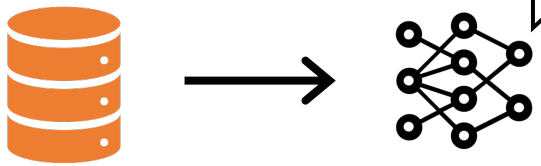


* Collecting high-power data is obstructed by thermal throttling.



Neural Network: Data Imbalance

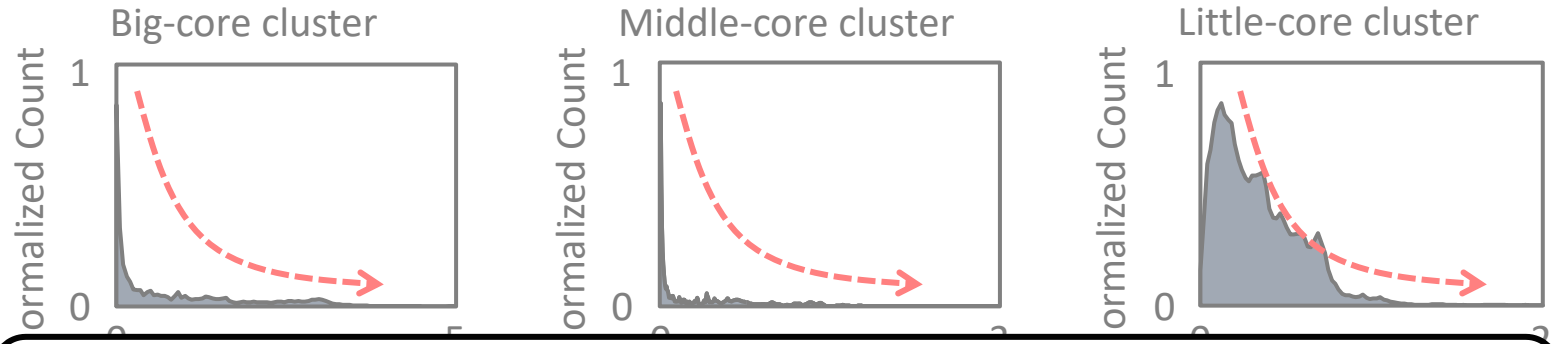




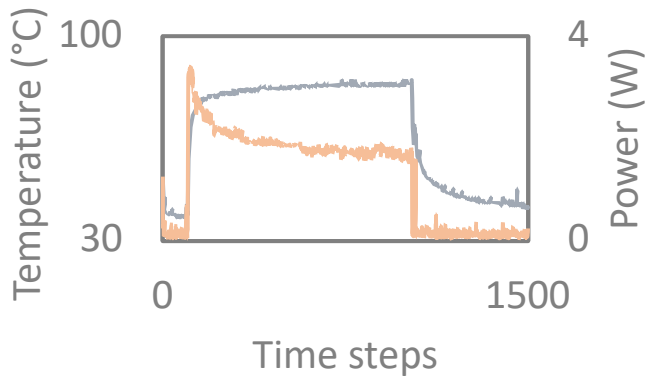
Mean squared error (MSE) loss function

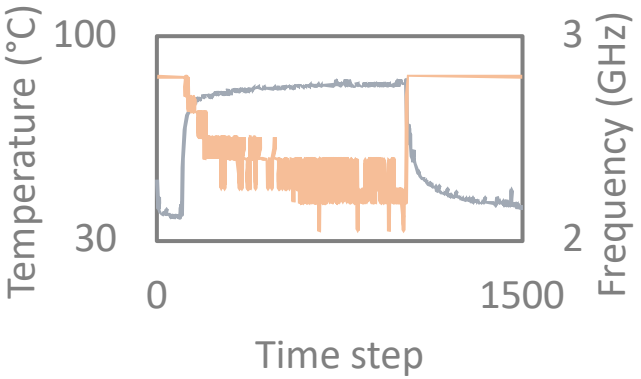
Challenge 1: Long tail distribution of collected dataset

Long-tail sample distributions of power for each core cluster

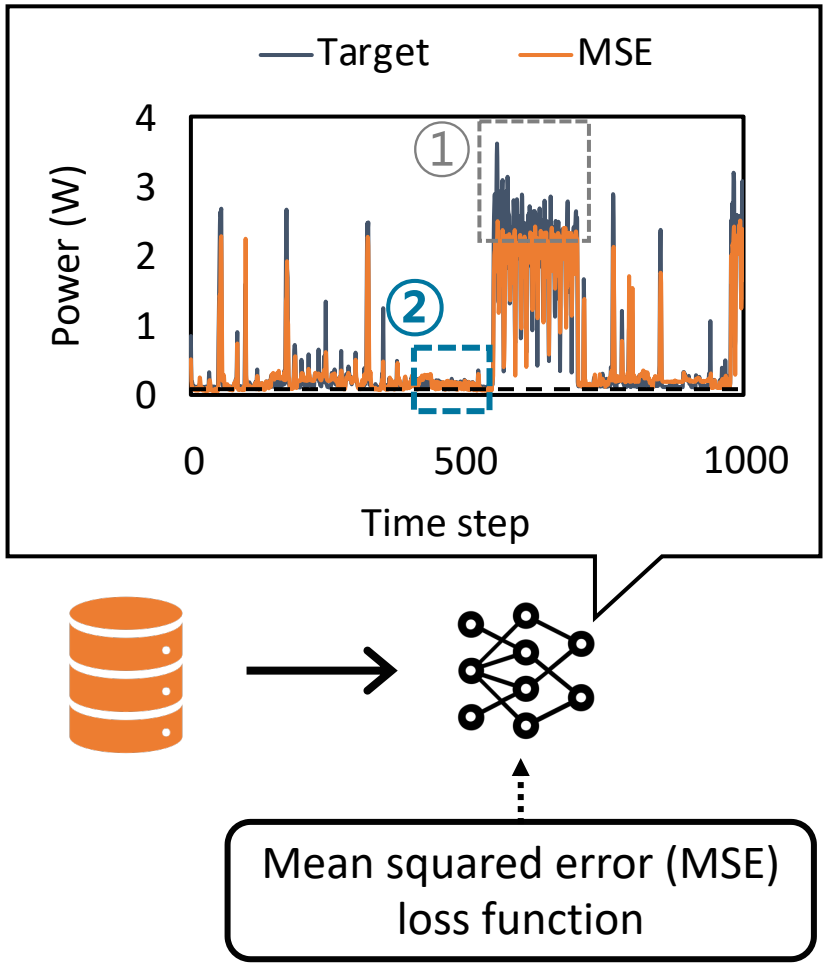


1. We need a term that encourages the NN model to predict high power states **despite the data imbalance**.



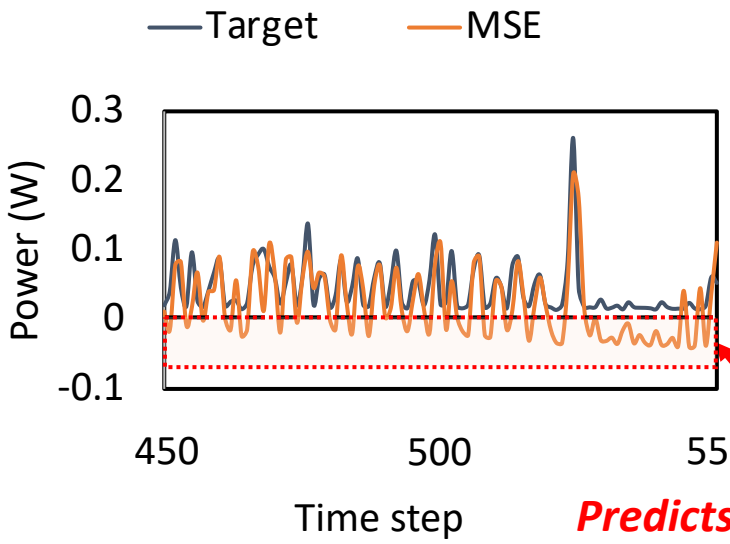


Neural Network: Negative Power Predictions



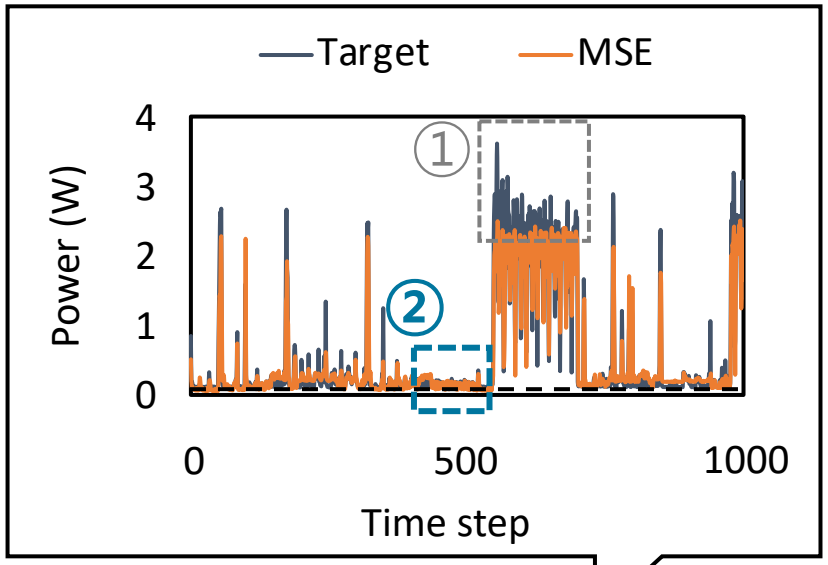
Mean squared error (MSE) loss function

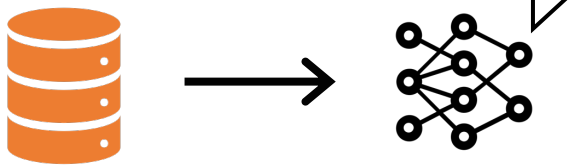
Challenge 2: Negative power predictions



- Since **MSE** is applied only to minimize residuals, it does **not prevent the negative-number predictions** from occurring.

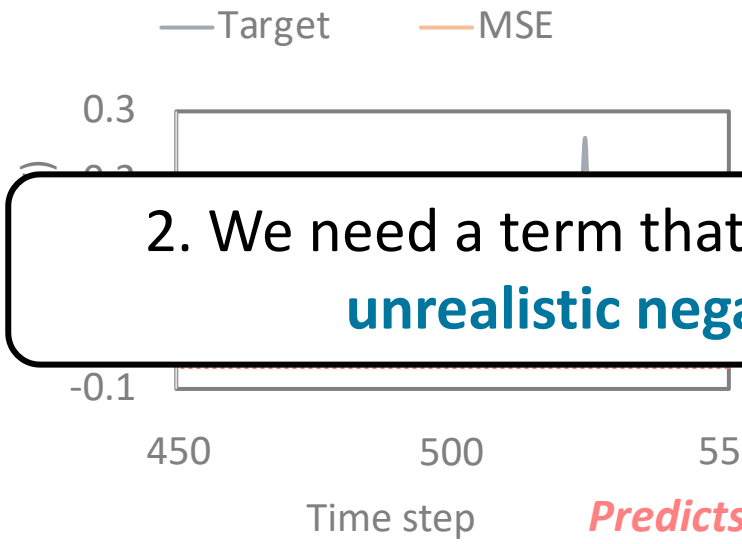
Neural Network: Negative Power Predictions





Mean squared error (MSE) loss function

Challenge 2: Negative power predictions



- Since **MSE** is applied only to minimize residuals, it does

2. We need a term that prevents the prediction of **unrealistic negative power states**.

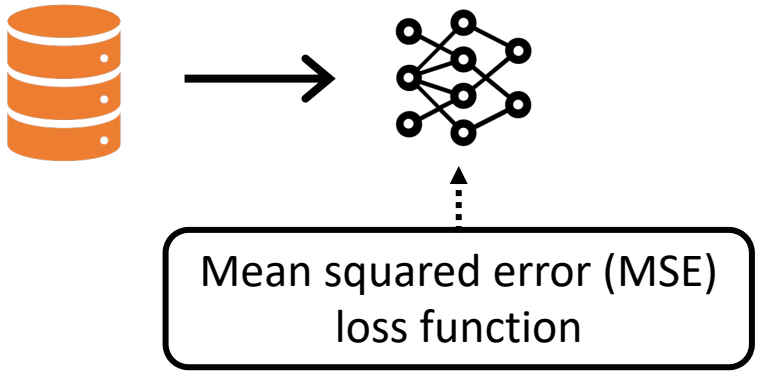
Systematic loss function

Encouraging high power prediction

$$\mathcal{L}_{\text{sys}} = \frac{1}{N}\sum_{i=1}^{N}\left(\lambda_{\text{exp}} e^{\left(a_{\text{exp}} y_i\right)}(y_i - \hat{y}_i)^2 + \lambda_{\text{neg}}\max(0, -\hat{y}_i)\right)$$

Preventing negative power prediction

Neural Network: Retrain with Systematic Loss



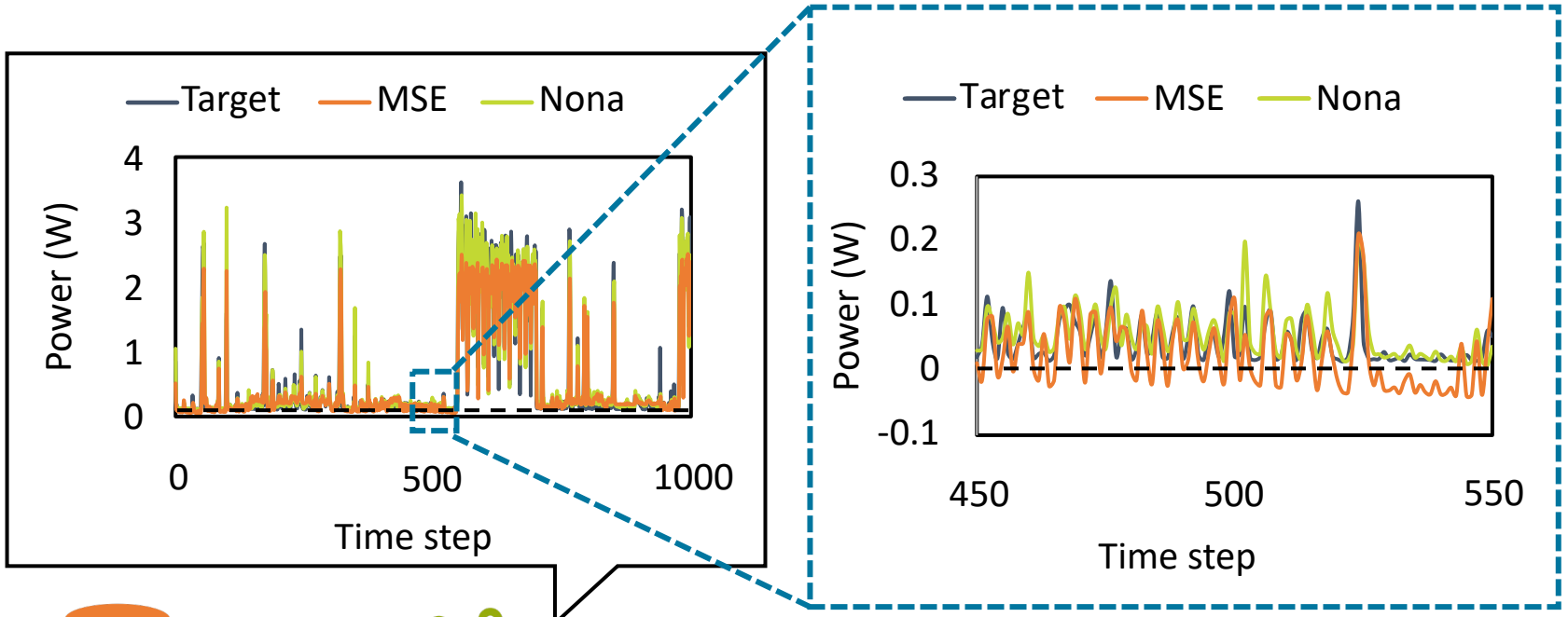
Systematic loss function

Encouraging high power prediction

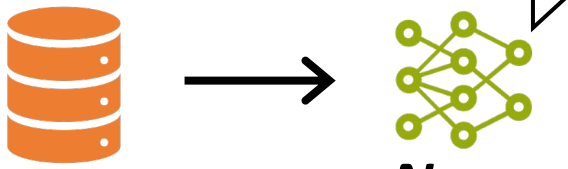
$$\mathcal{L}_{\text{sys}} = \frac{1}{N}\sum_{i=1}^{N}\left(\lambda_{\text{exp}} e^{\left(a_{\text{exp}} y_i\right)}(y_i - \hat{y}_i)^2 + \lambda_{\text{neg}}\max(0, -\hat{y}_i)\right)$$

Preventing negative power prediction

Neural Network: Retrain with Systematic Loss



- Nona traces **rapid changes in power states**, including high-power states.
- It prevents the prediction of **negative power numbers.**
- The power prediction overhead of Nona is **only 0.11ms** on the CPU.



Nona

Retraining with ***the novel systematic loss function***

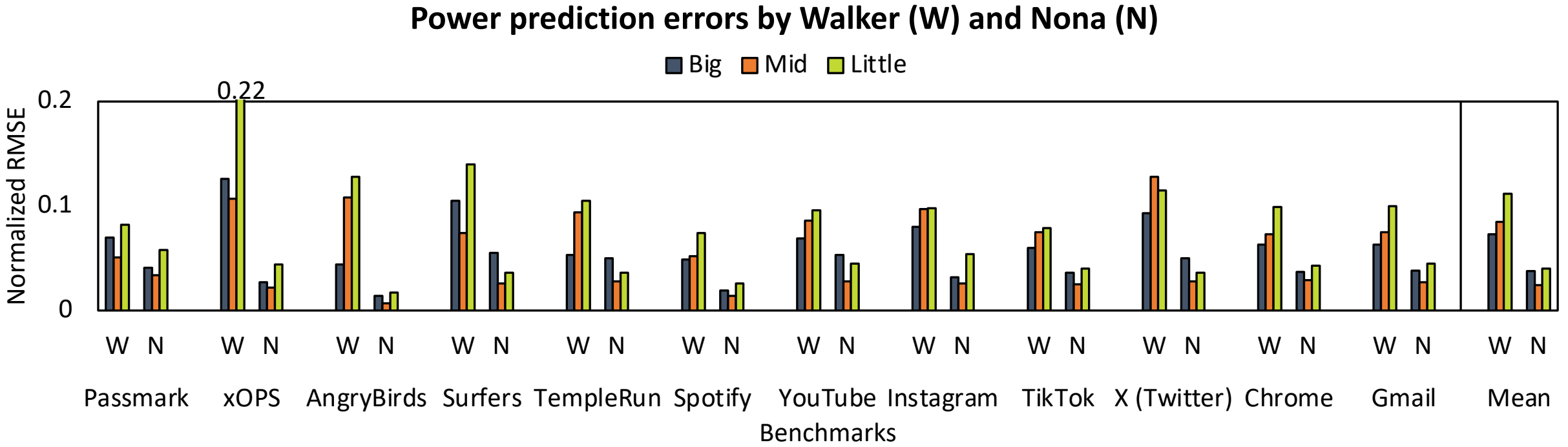
Systematic loss function

Encouraging high power prediction

$$\mathcal{L}_{\text{sys}} = \frac{1}{N}\sum_{i=1}^{N}\left(\lambda_{\text{exp}} e^{\left(a_{\text{exp}} y_i\right)}(y_i - \hat{y}_i)^2 + \lambda_{\text{neg}}\max(0, -\hat{y}_i)\right)$$

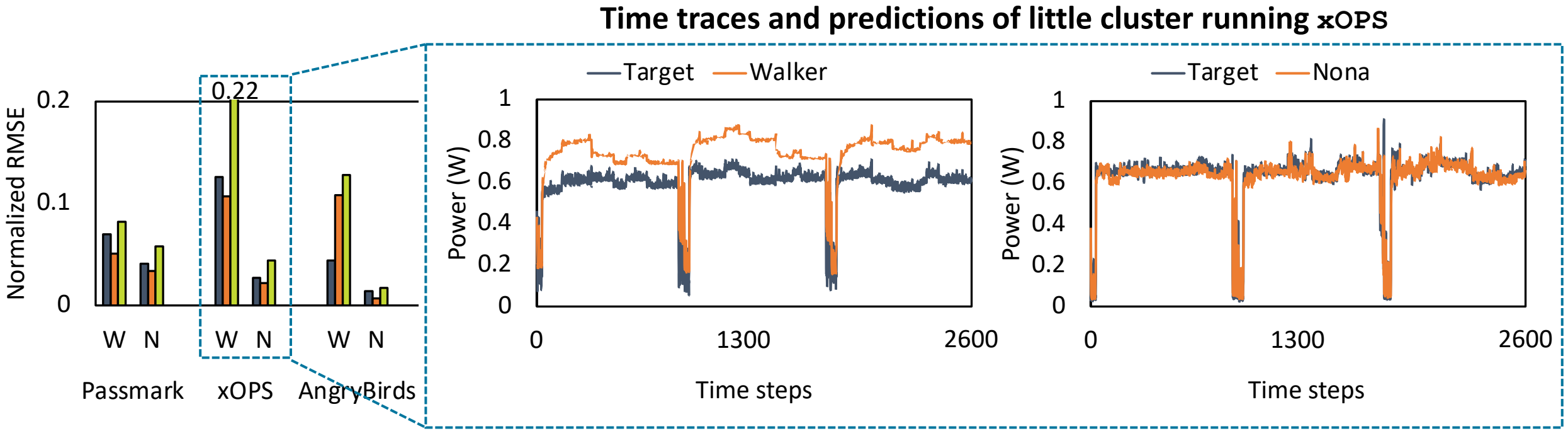
Preventing negative power prediction

Comparative Evaluation



- Nona shows an average error of **3.4%**, which is better than Walker's by **2.6x**.
- Nona has **3.8%, 2.5%, and 4.0%** prediction errors for big, middle, and little-core clusters, respectively.
- It captures the **wide range of power variations**, as shown with `xOPS` (i.e., a high-power stress benchmark).

Comparative Evaluation



- Nona shows an average error of **3.4%**, which is better than Walker's by **2.6x**.
- Nona has **3.8%, 2.5%, and 4.0%** prediction errors for big, middle, and little-core clusters, respectively.
- It captures the **wide range of power variations**, as shown with xOPS (i.e., a high-power stress benchmark).

Conclusion

- Nona is a NN-based power prediction model for heterogeneous CPUs.
- It introduces a **novel PMC selection** strategy to identify those that have the highest correlations with power.
- Nona also introduces a **systematic loss function** that enables unbiased training for imbalanced data samples and improved prediction accuracy.