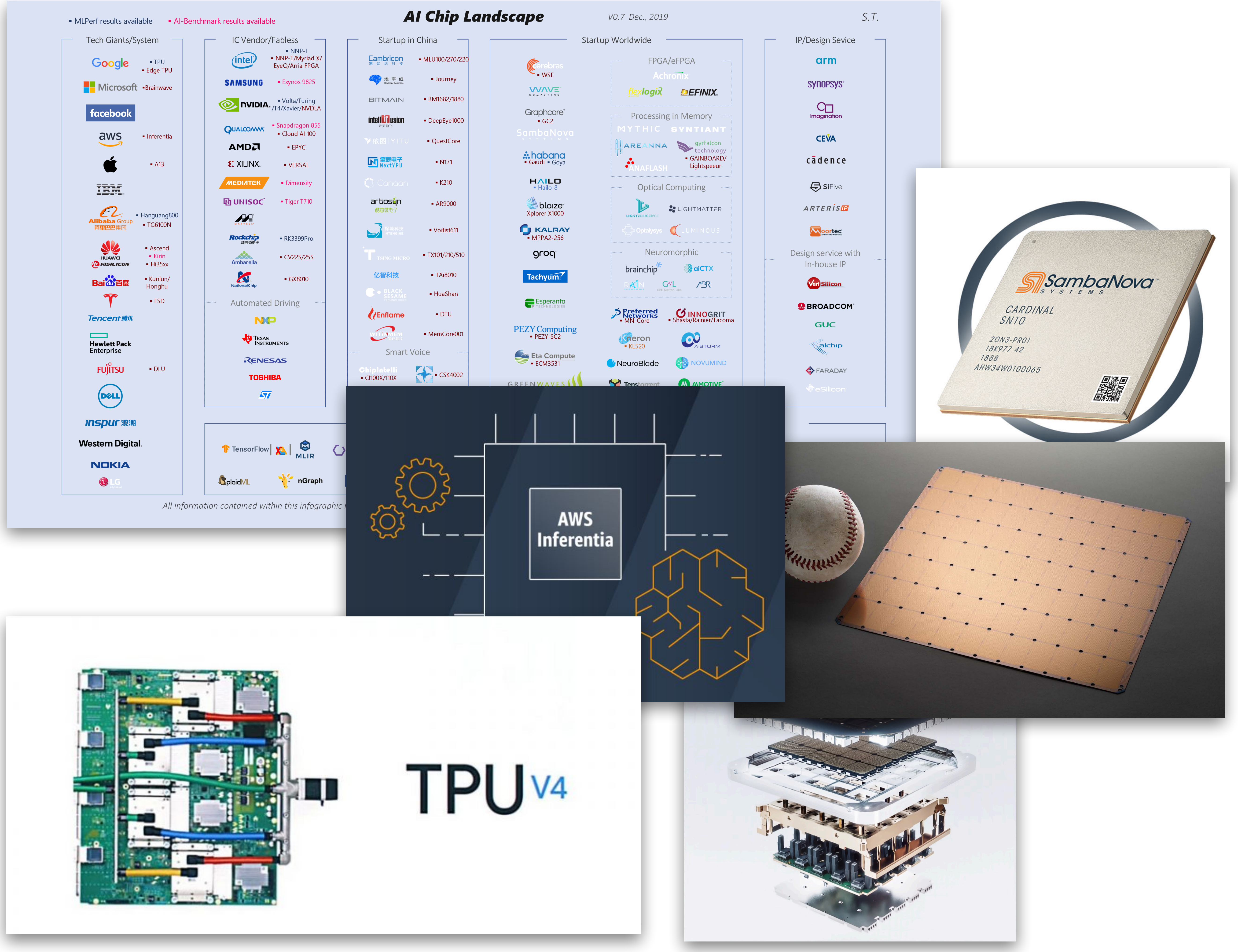
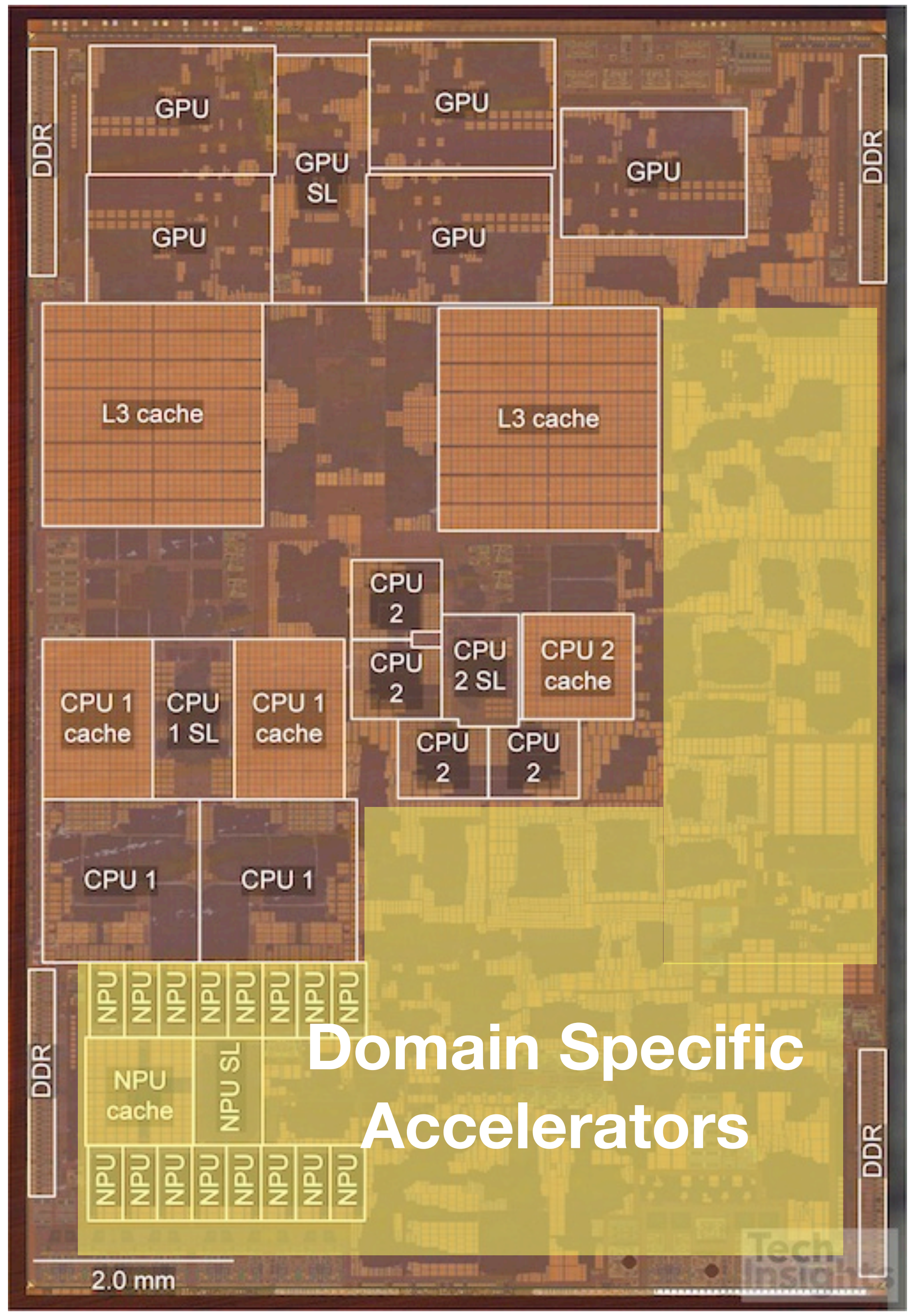
DOSA: Differentiable Model-Based One-Loop Search for DNN Accelerators

Charles Hong (UC Berkeley),
Qijing Huang (NVIDIA), Grace Dinh (UC Berkeley),
Mahesh Subedar (Intel Labs), Yakun Sophia Shao (UC Berkeley)

Hardware Acceleration is Everywhere

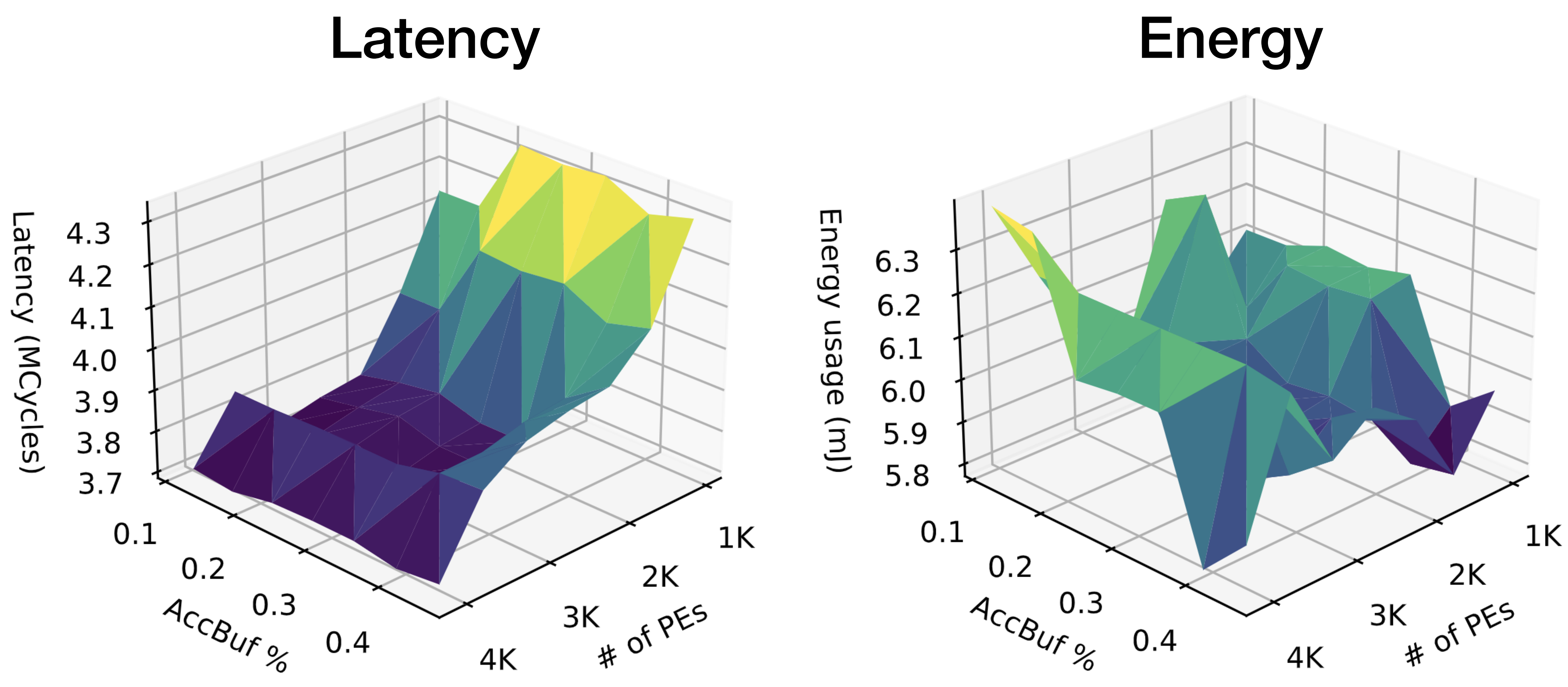




Apple A15

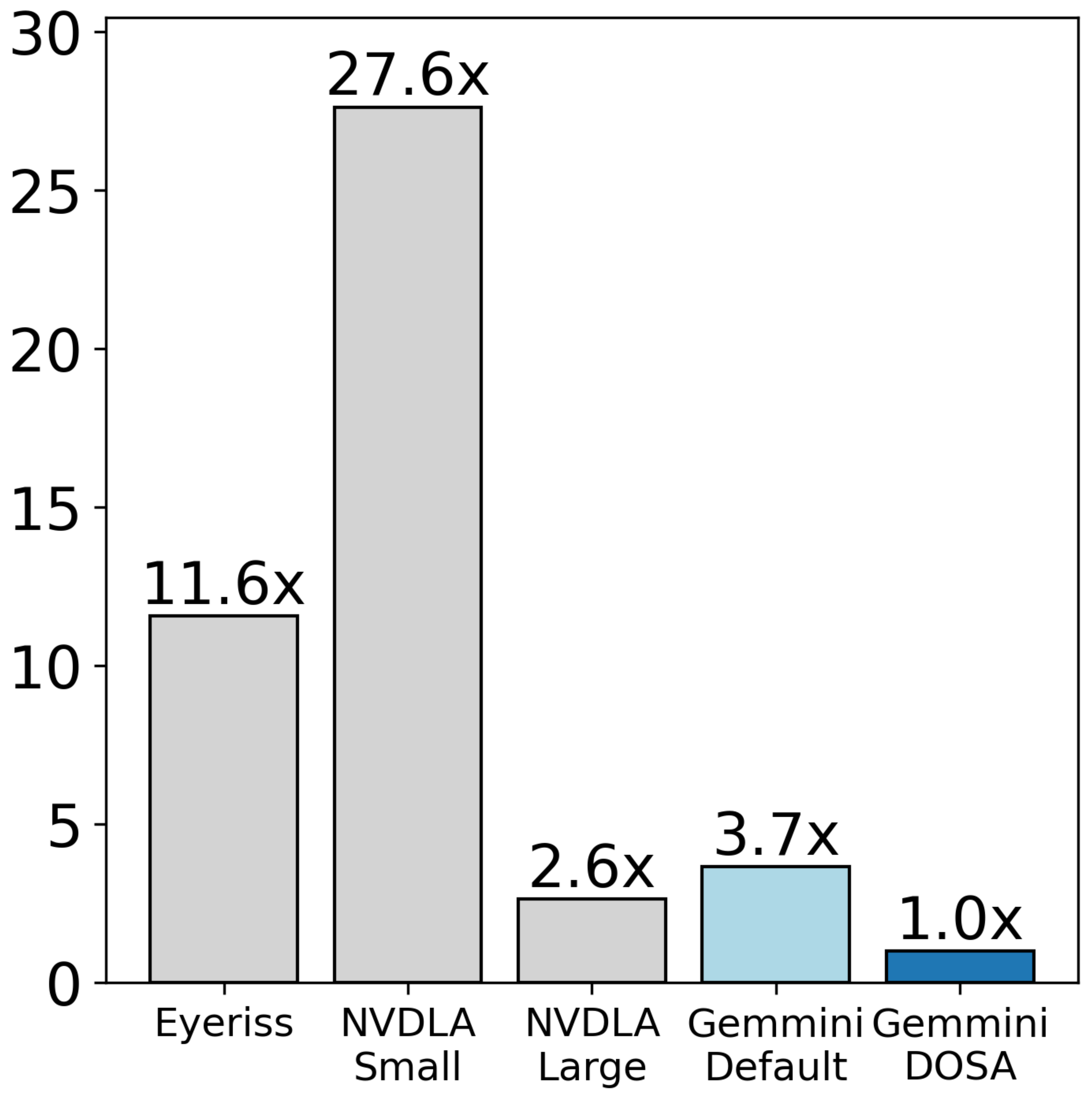
*TechInsights.com Apple iPhone 13 teardown

Accelerator Design Space Exploration is Challenging



Performance of ResNet-50 as # of PEs and accumulator size change

… but impactful.



Normalized EDP (lower is better)

Conventional DSE

Two nested loops:

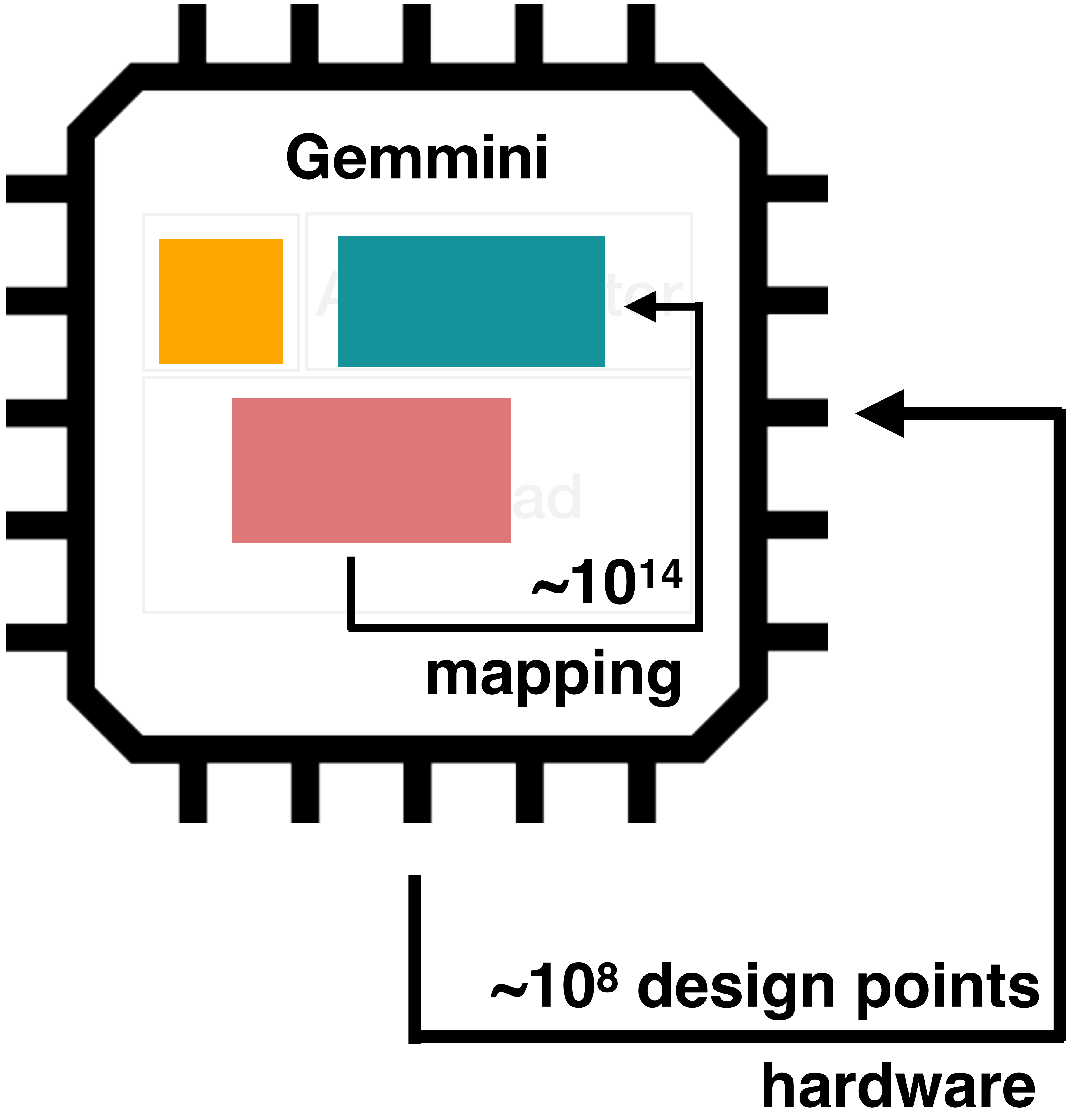
1. Hardware loop
   2. Mapping loop

1 design point

= up to hours/days of hardware simulation

~10^{22} design points

= quadrillions of years!



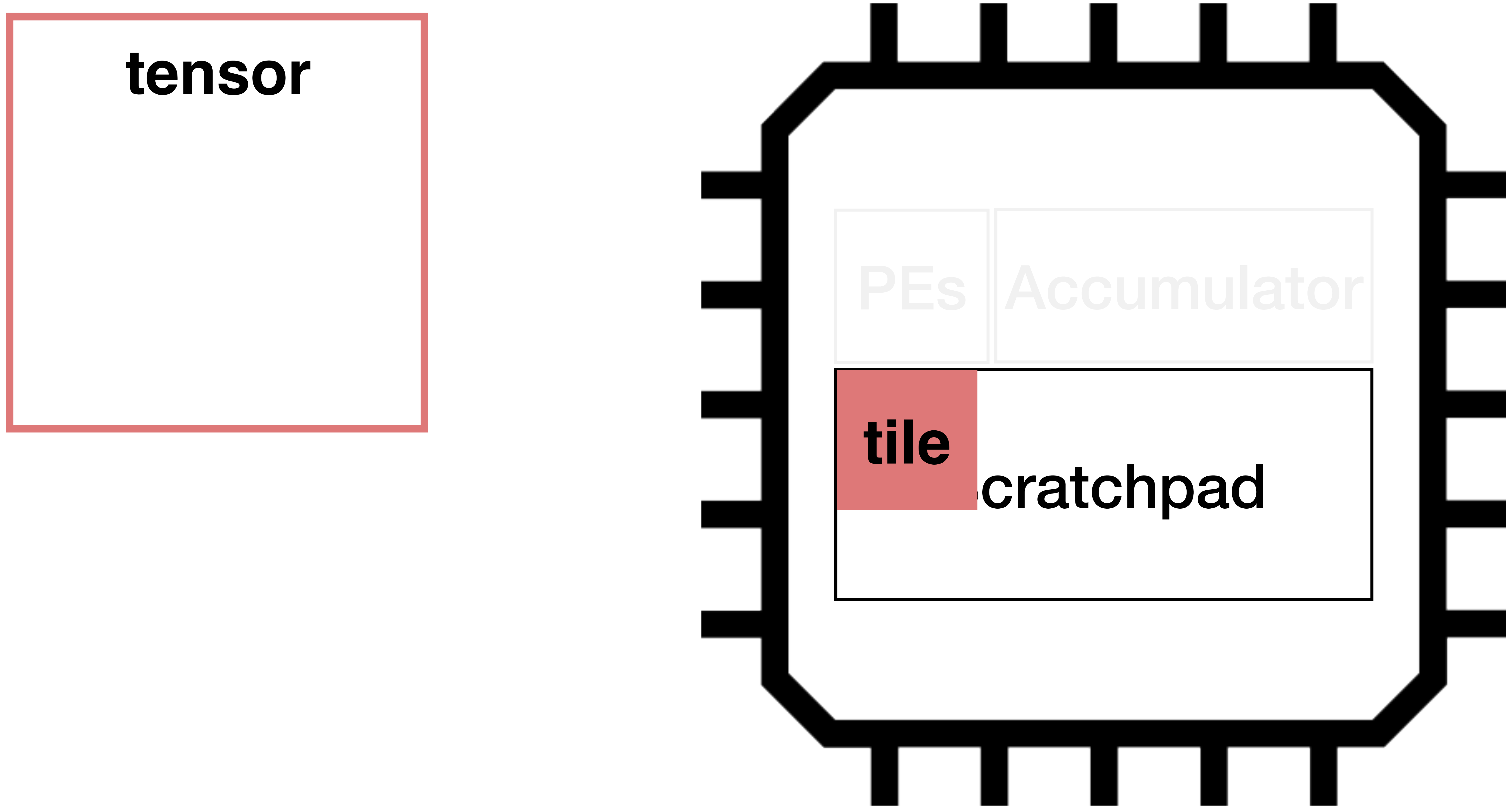
How does DOSA tackle this ~10^{22} search space?

1. Do mapping-first search.

2. Use differentiable, interpretable performance models.

3. Apply deep learning to bridge the gap between models and RTL.

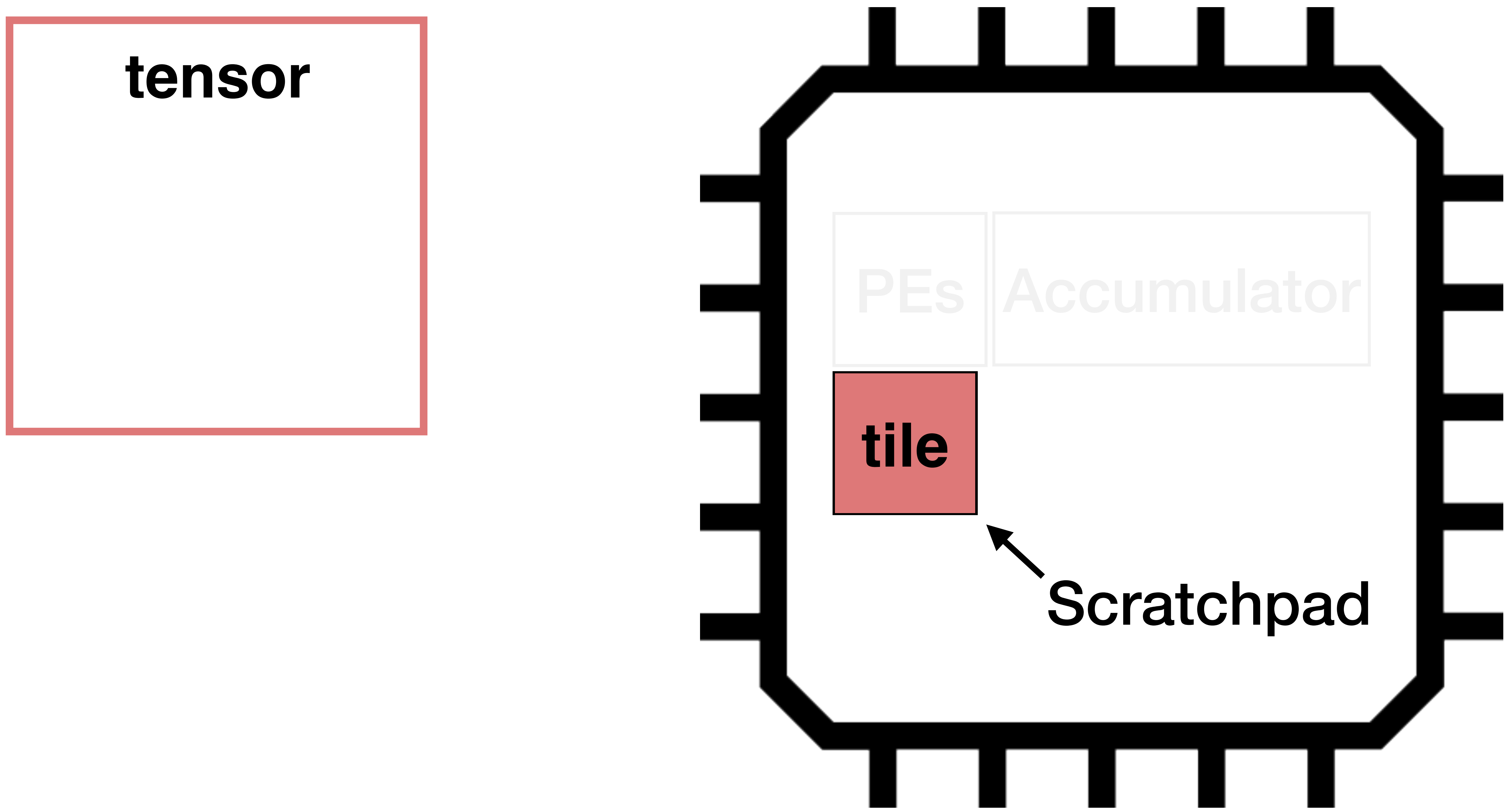
Inefficiencies of Hardware-First Search



Scratchpad too big:

Unnecessary energy and area consumption.

Fit the Hardware to the Mapping



Observation:

- You can infer optimal scratchpad size from mapping.

How does DOSA tackle this ~10^{22} search space?

1. Do mapping-first search.

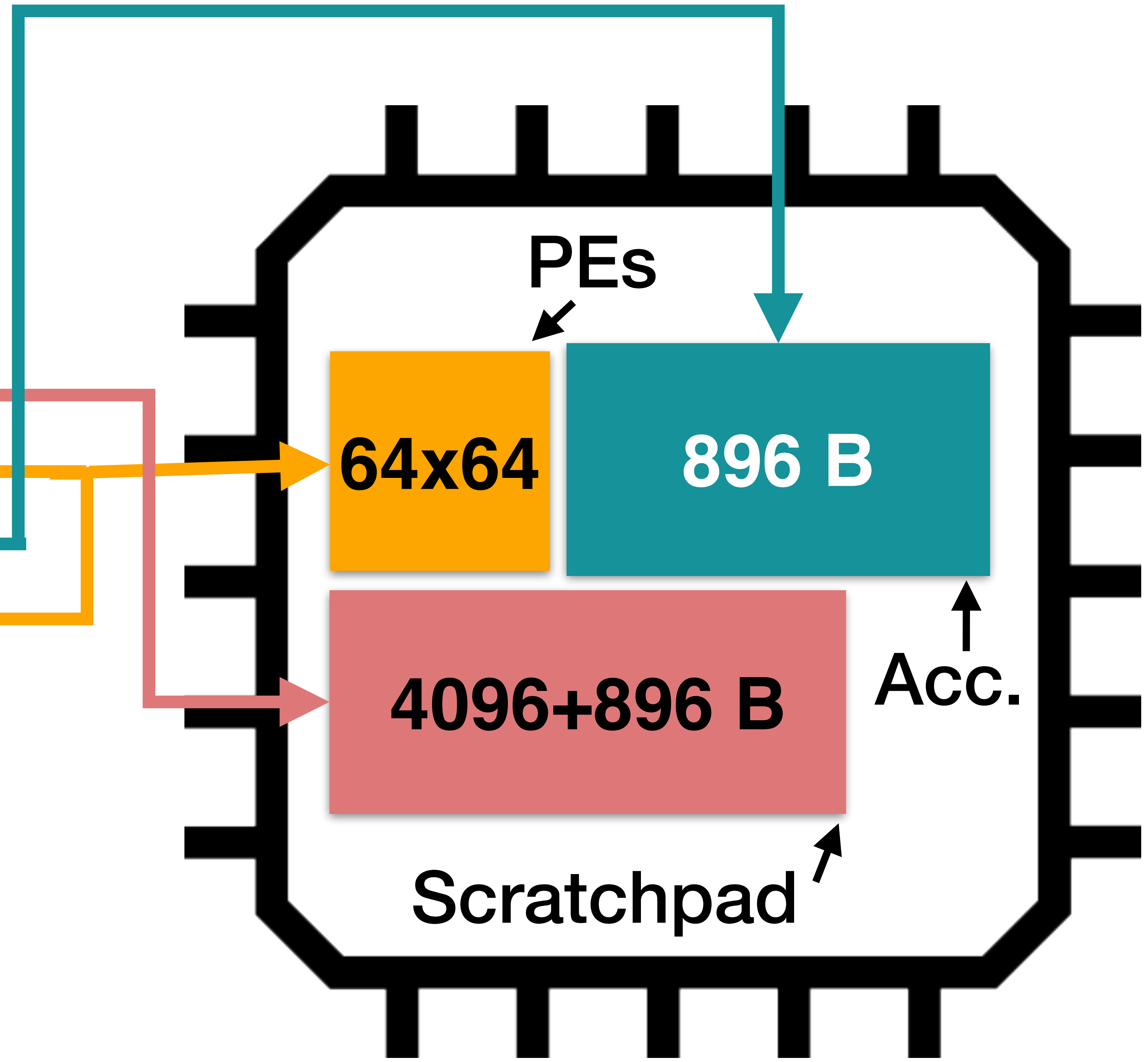
2. Use differentiable, interpretable performance models.

3. Apply deep learning to bridge the gap between models and RTL.

Mapping-First Search

Mapping Nested Loop:

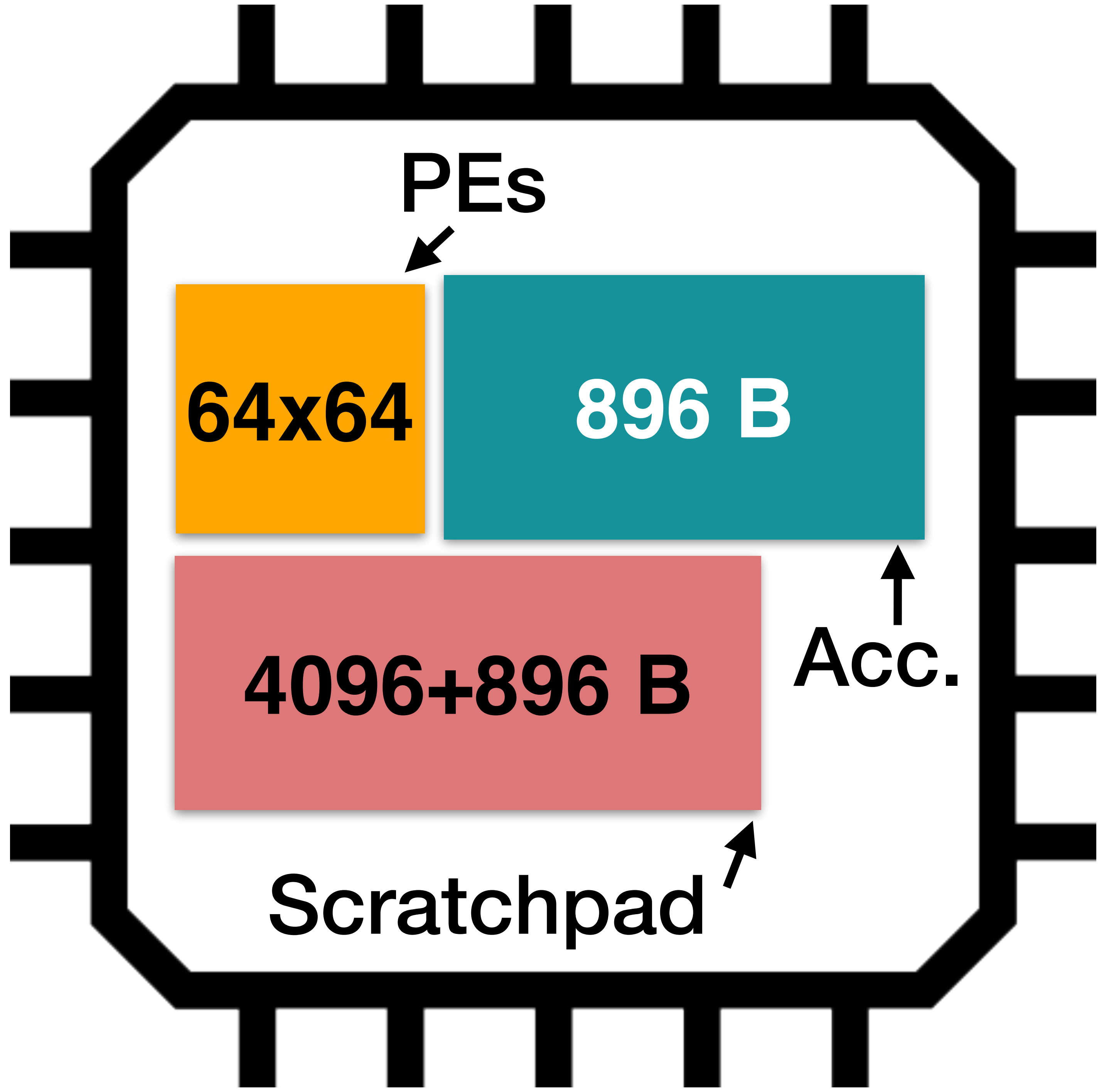
```
...
  // Scratchpad (Weights:4096, Inputs: 896)
  spatial_for k2 in [0:64):
    // Accumulator (Outputs:896)
    spatial_for c1 in [0:64):
      // Registers (Weights: 4096)
      for q0 in [0:14):
        ...
```



Mapping-First Search

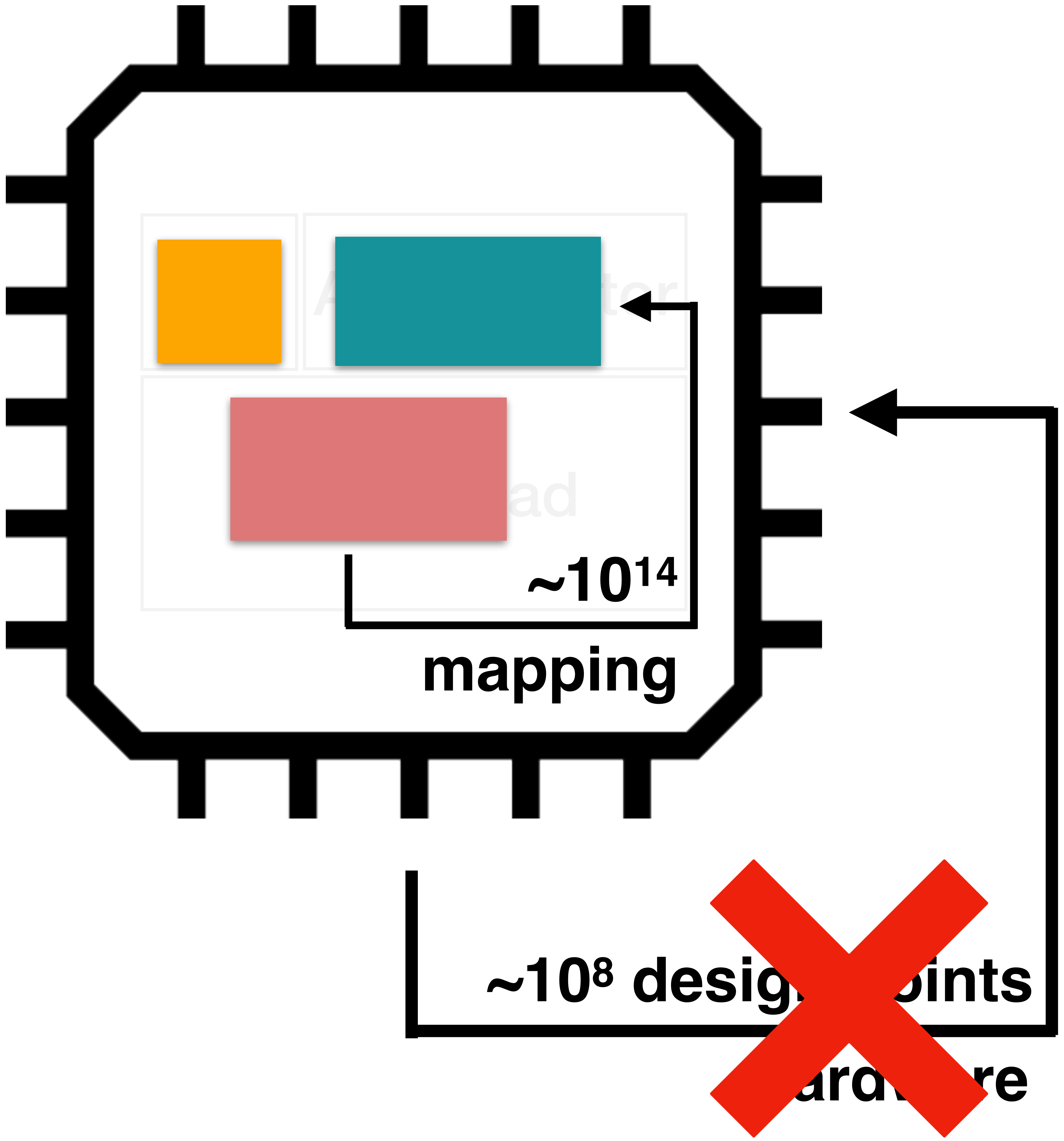
Mapping-first search:

- Search mappings, unconstrained
- Infer optimal hardware

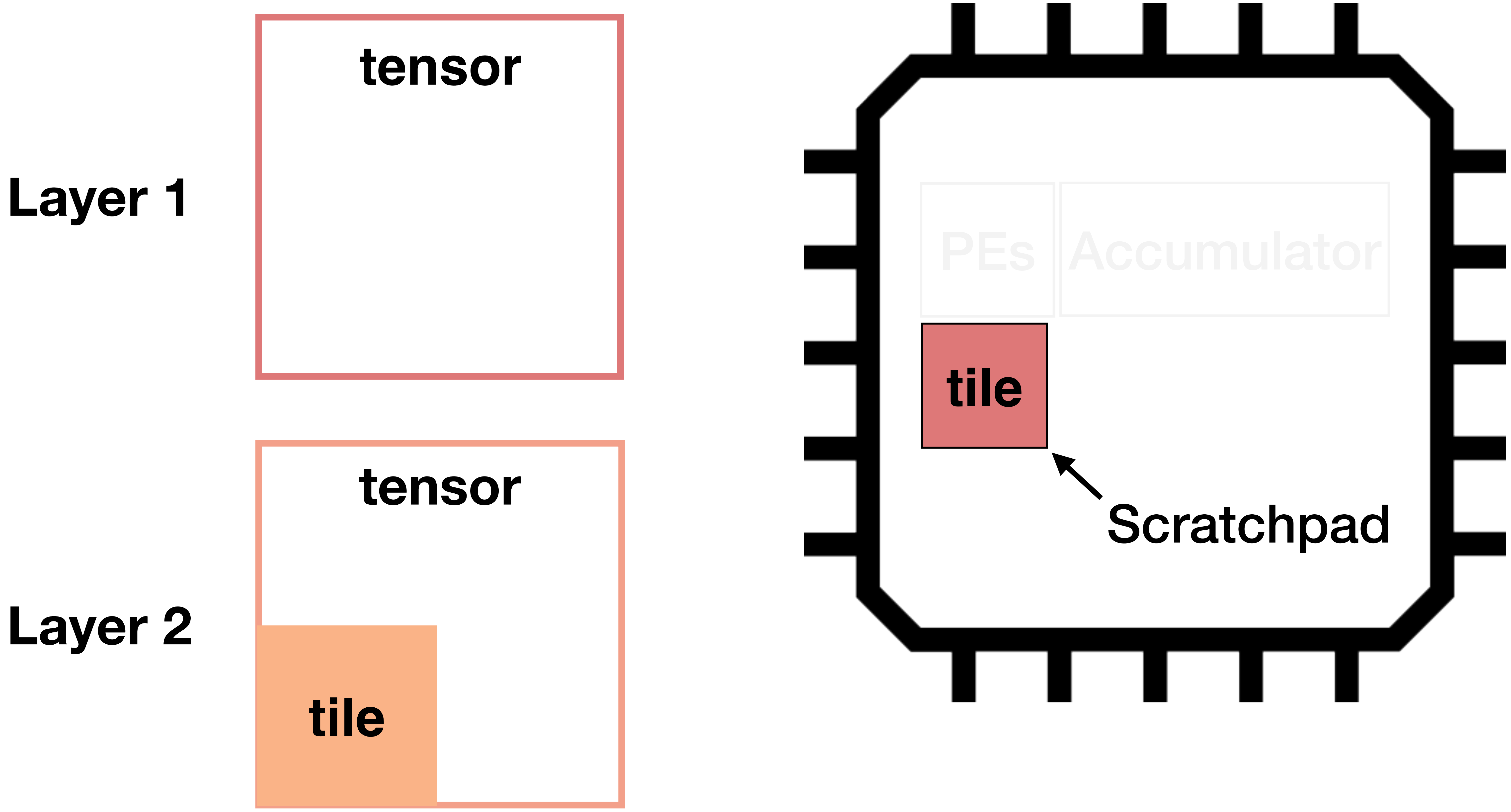


Mapping-First Search

Explore the design space with *one* loop.

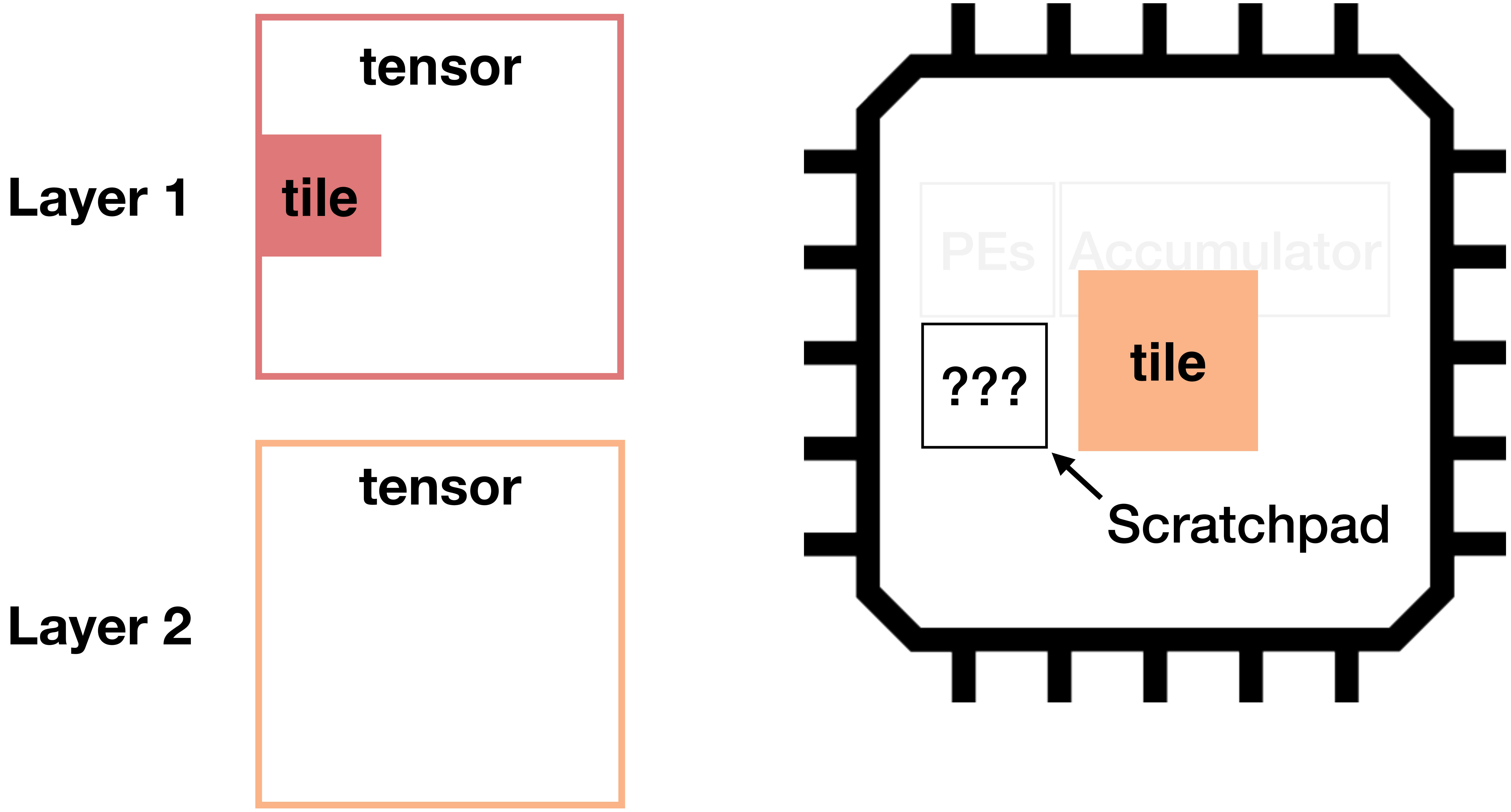


Multi-Layer DSE



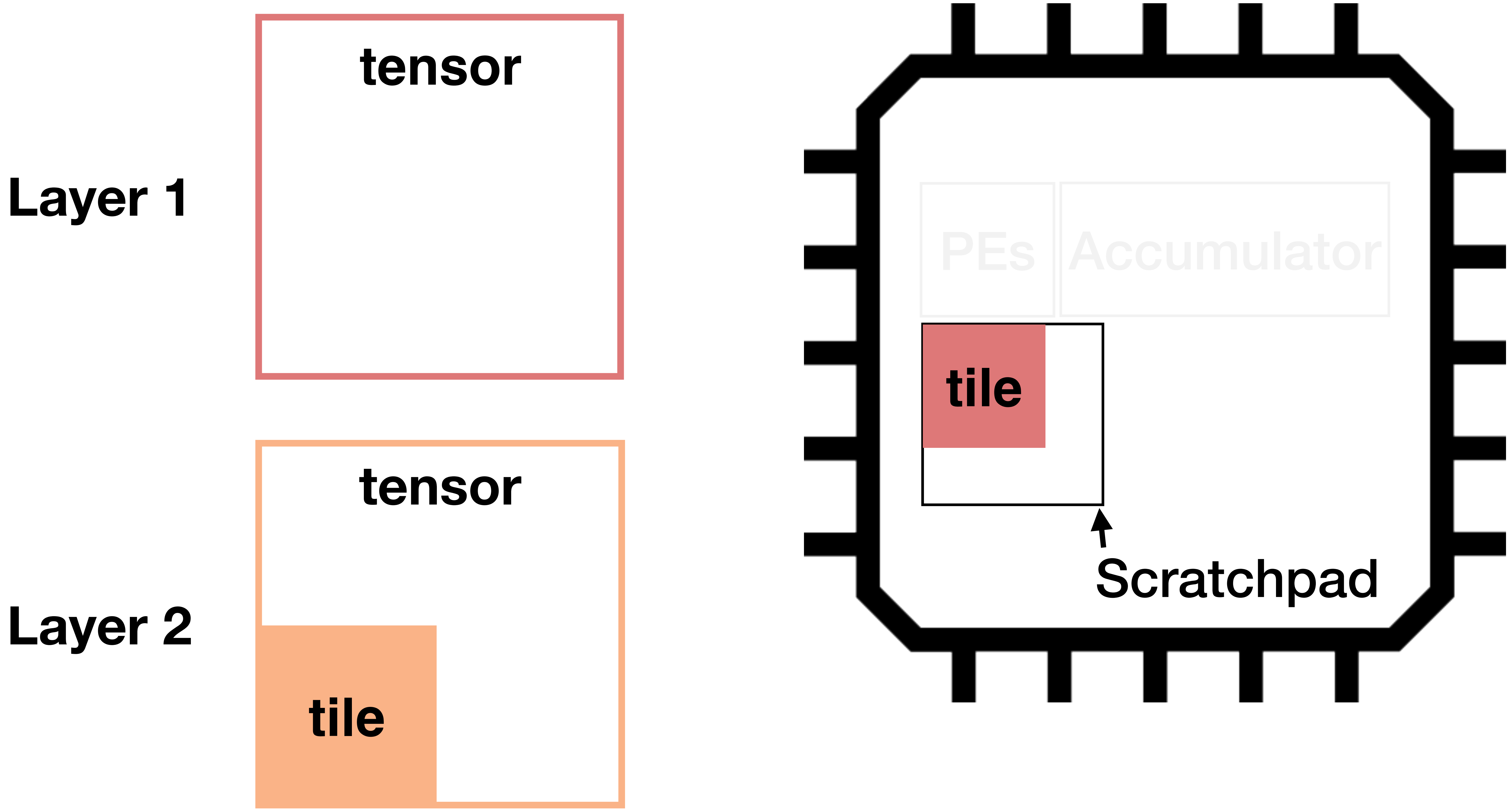
What happens when we get to the next layer?

Multi-Layer DSE



Scratchpad has to fit tile sizes of all layers.

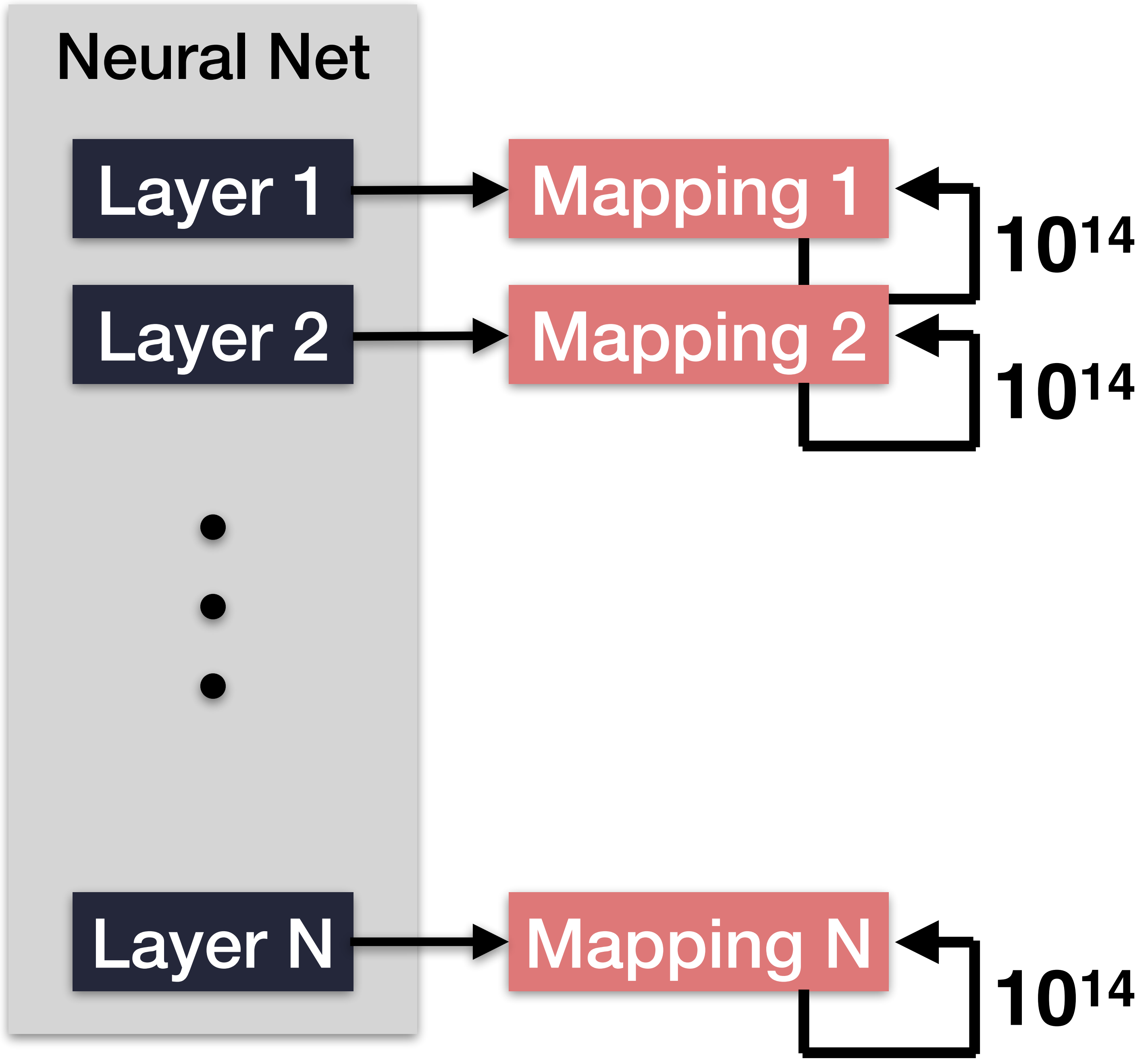
A High-Dimensional Optimization Problem



Resizing the scratchpad affects the energy of all layers.

⟹ Search mappings for all layers together.

A Very High-Dimensional Optimization Problem



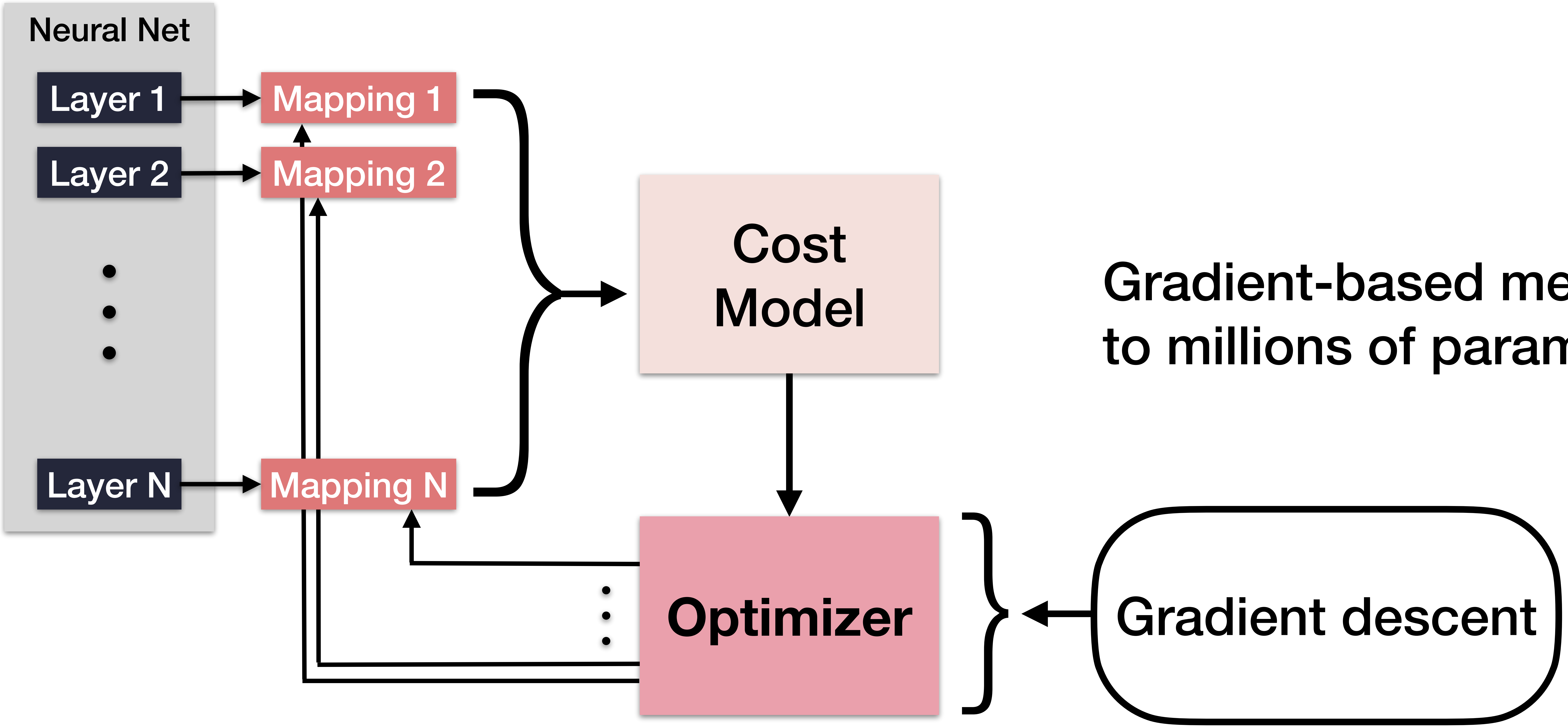
Unique layers per DNN model: 5-25

Mapping variables per layer: 40

Up to 1000 input variables to optimize!

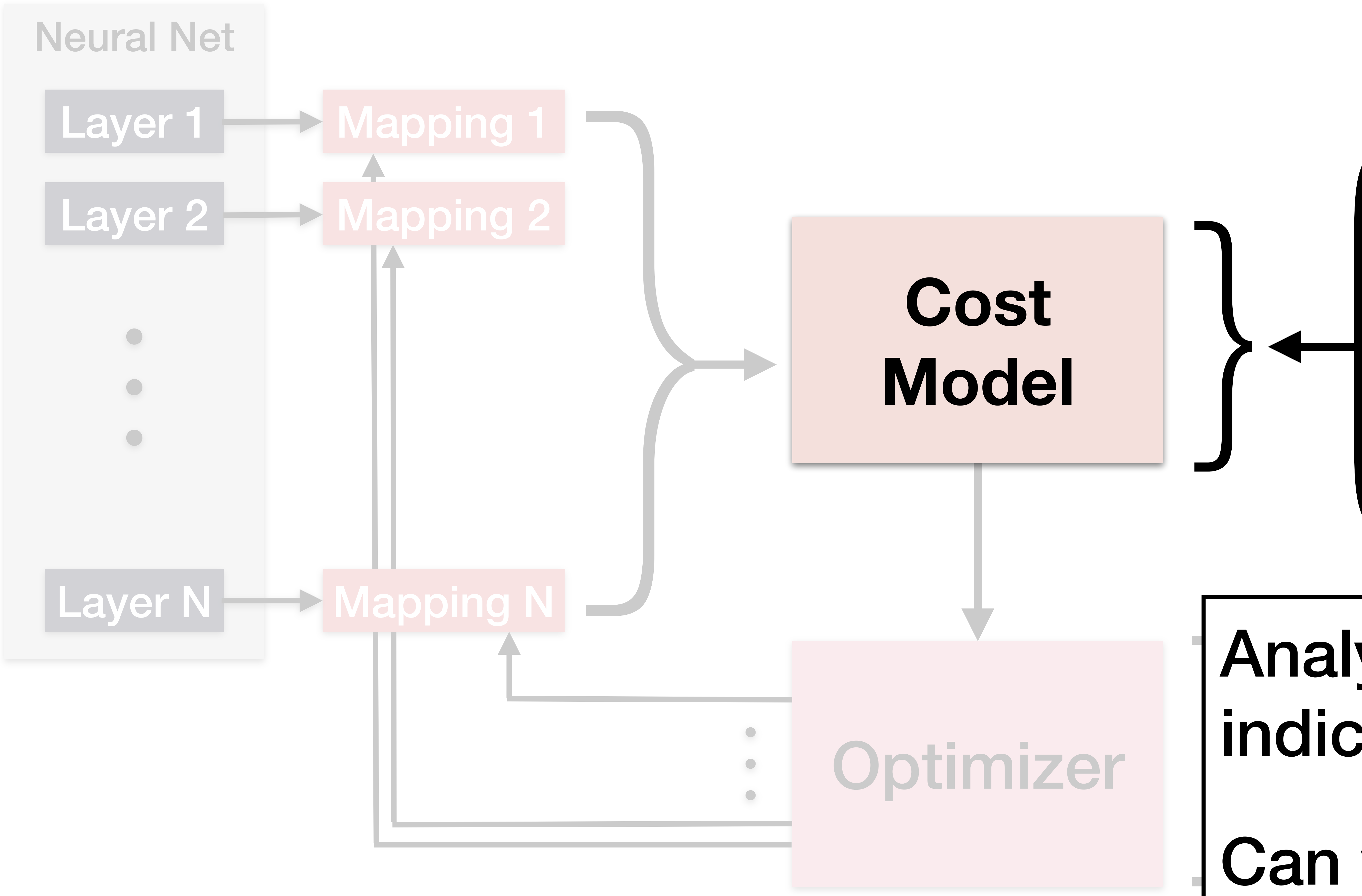
$10^{14} \times 10^{14} \times \ldots \times 10^{14} = (10^{14})^N$ search space!

Choice of Optimizer



Gradient-based methods scale to millions of parameters.

Choice of Cost Model



- Must be fast
- Must be accurate
- Must be ***differentiable***

Analytical models provide a fast indicator for accelerator performance.

Can we construct a differentiable analytical model?

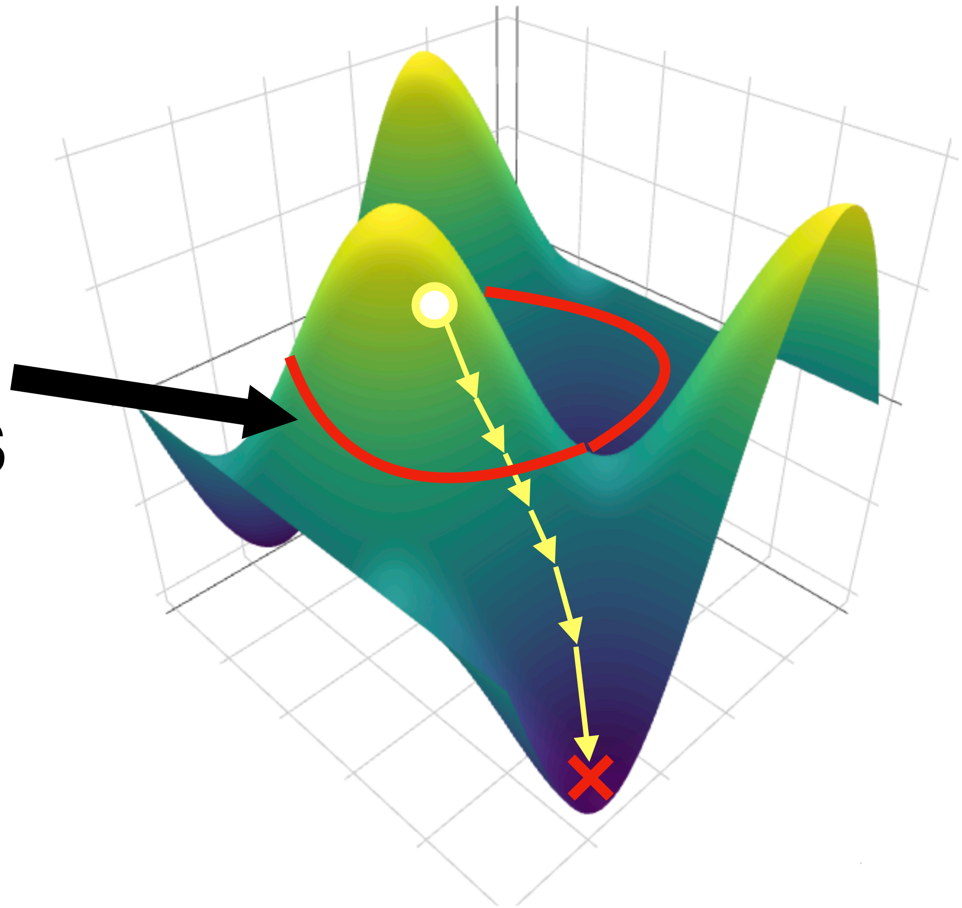
Differentiable Functions

Differentiable function: “a function whose derivative exists at each point in its domain.”

- Must be fast
- Must be accurate
- Must be ***differentiable***

Hardware-First Search (conventional)

Hardware Constraints



Mapping Search Space

Mapping-First Search

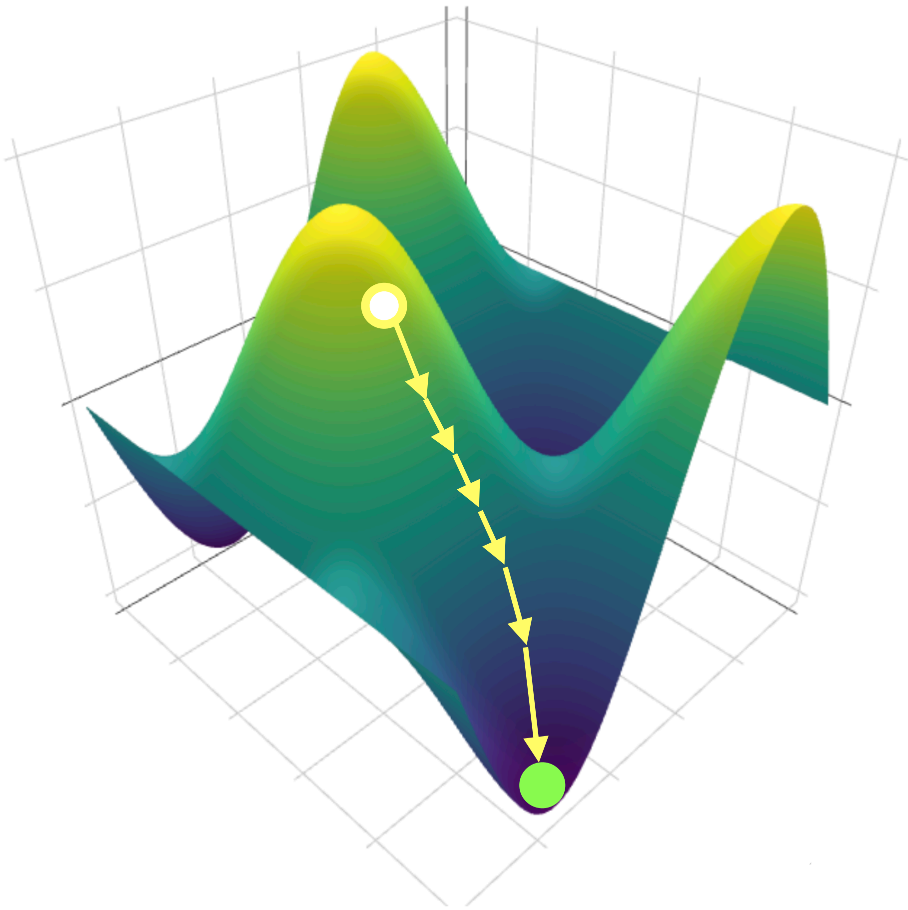
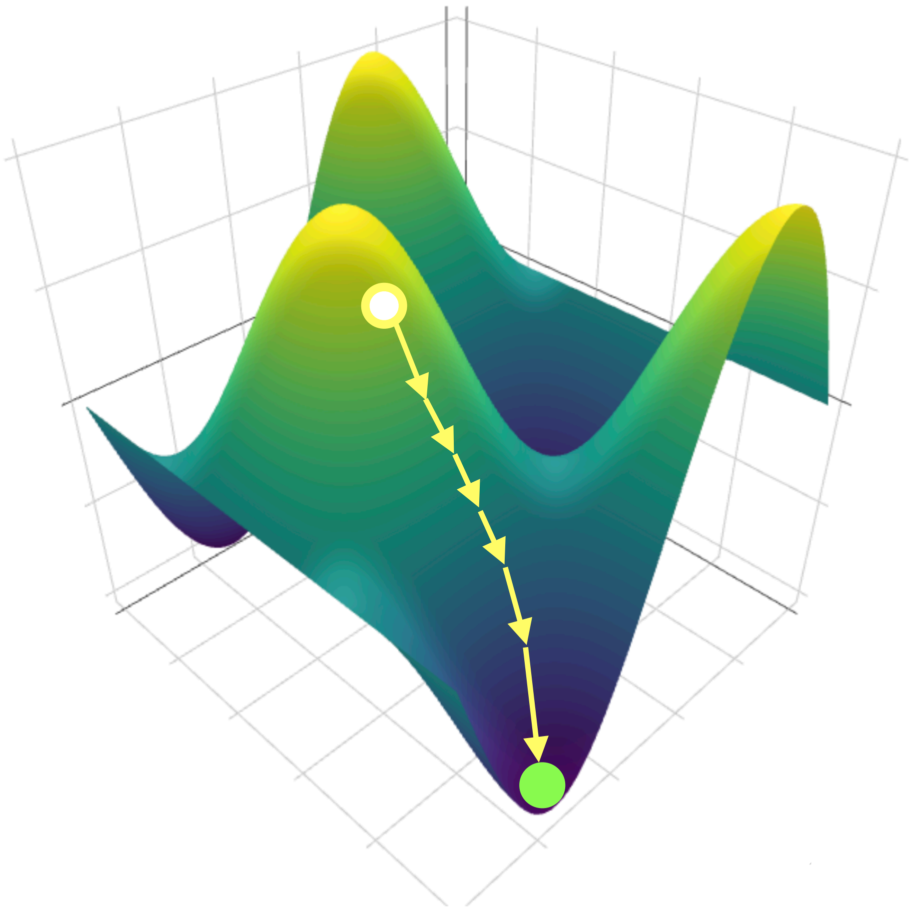


*Illustration from plotly.com

Differentiable Functions

💡 Mapping-first search is well-suited to gradient-based optimization!

Mapping-First Search



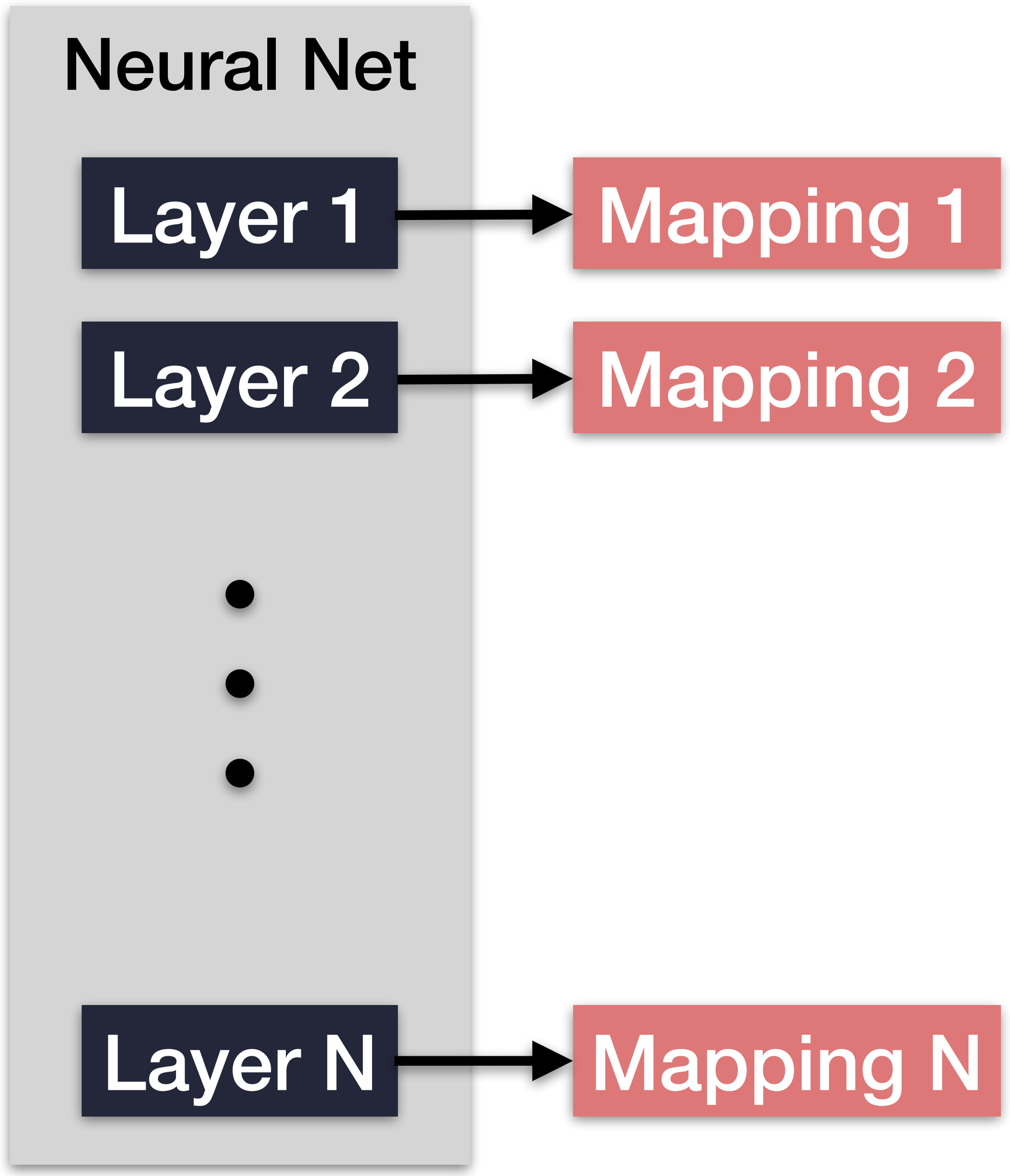
How does DOSA tackle this ~10^{22} search space?

1. Do mapping-first search.

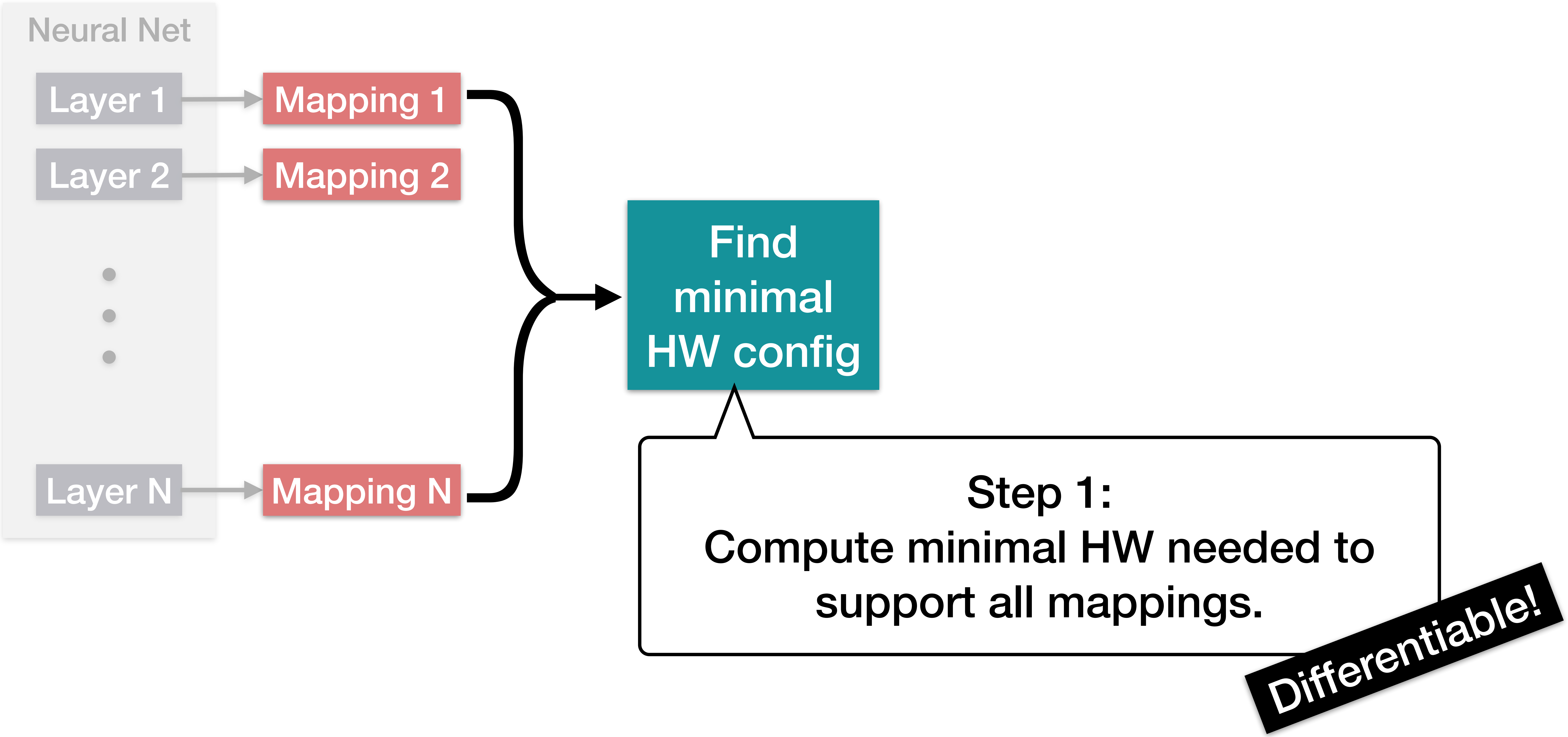
2. Use differentiable, interpretable performance models.

3. Apply deep learning to bridge the gap between models and RTL.

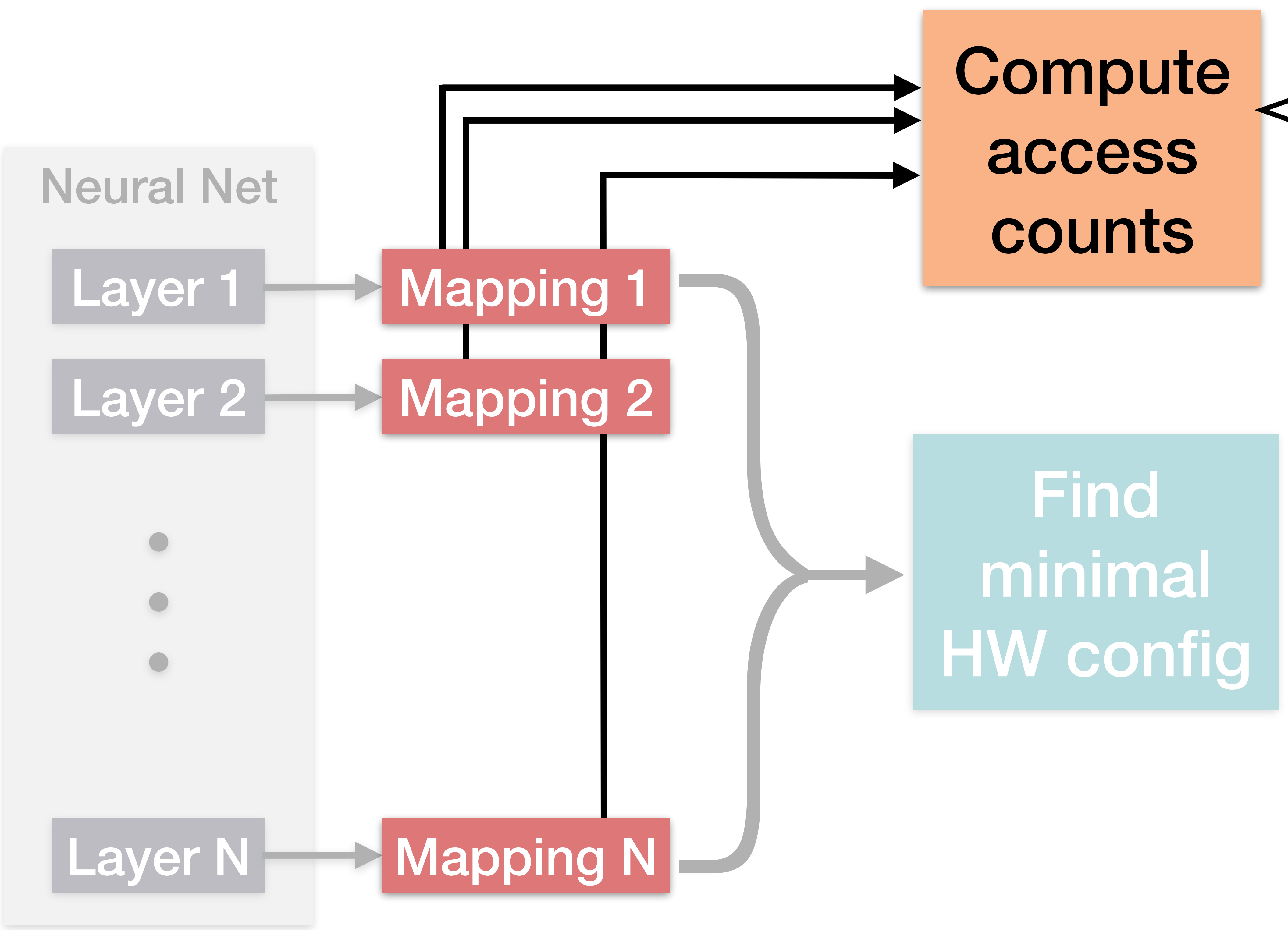
DOSA Differentiable Model



DOSA Differentiable Model



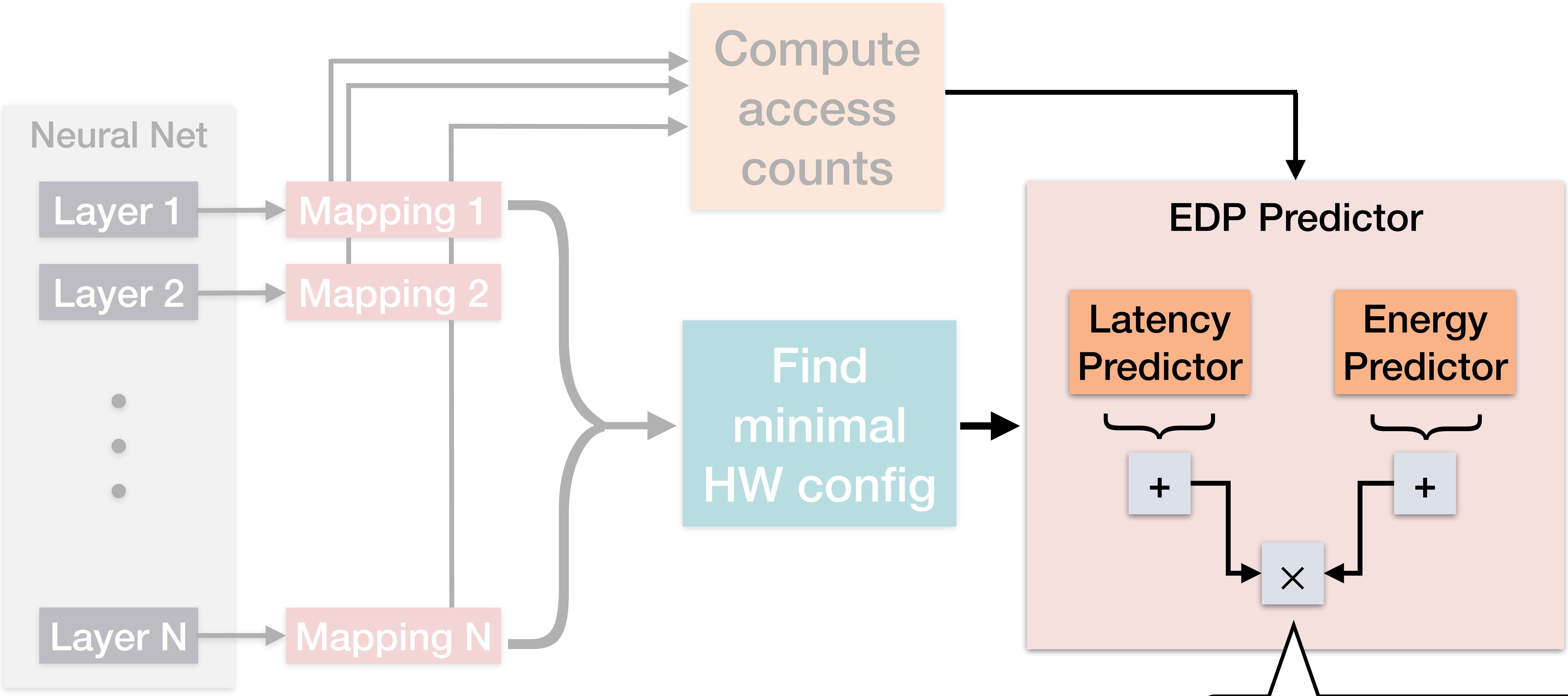
DOSA Differentiable Model



Step 2:
Using tiling factors and loop ordering, compute number of accesses/operations for each hardware component.

Differentiable!

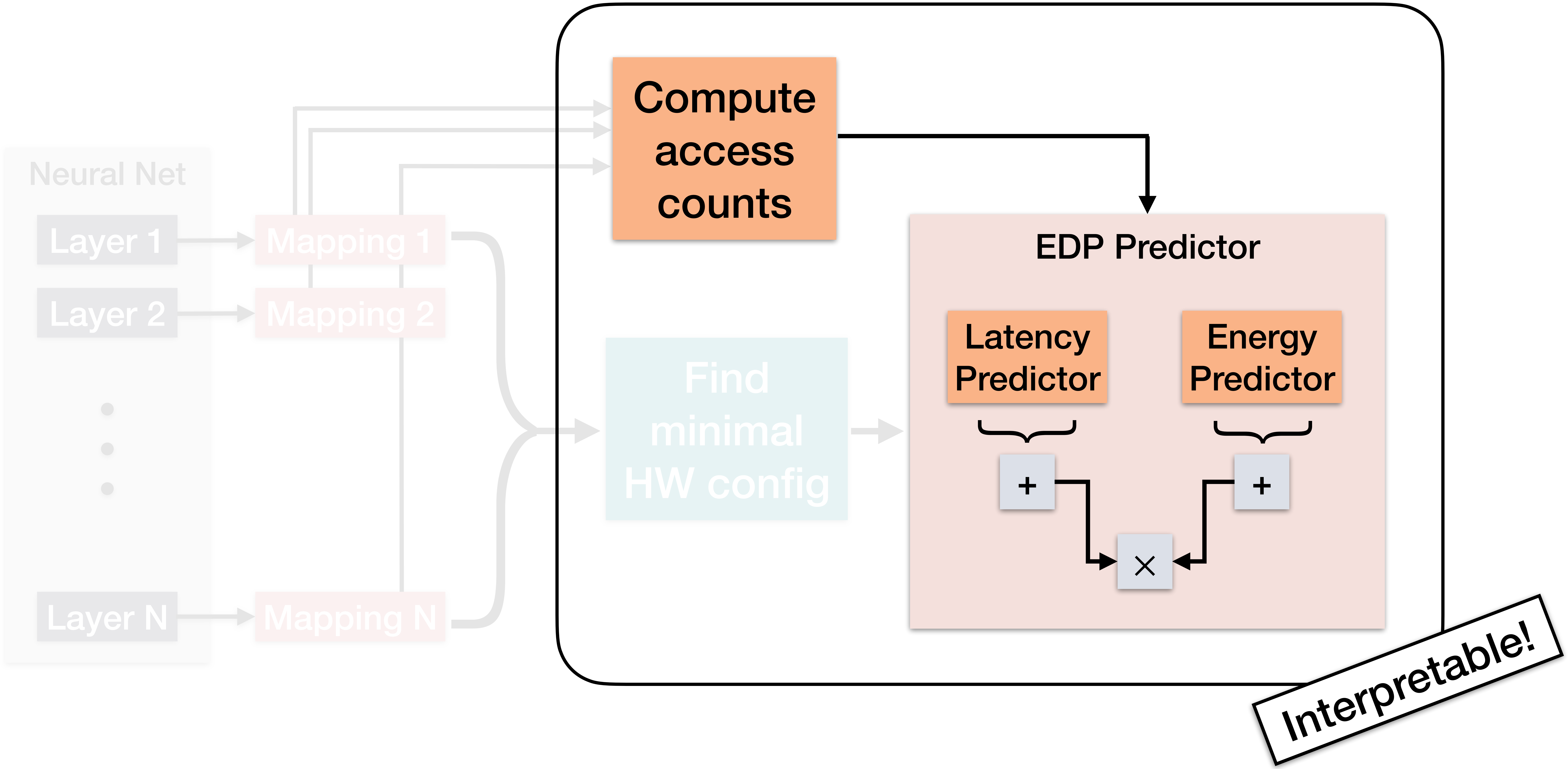
DOSA Differentiable Model



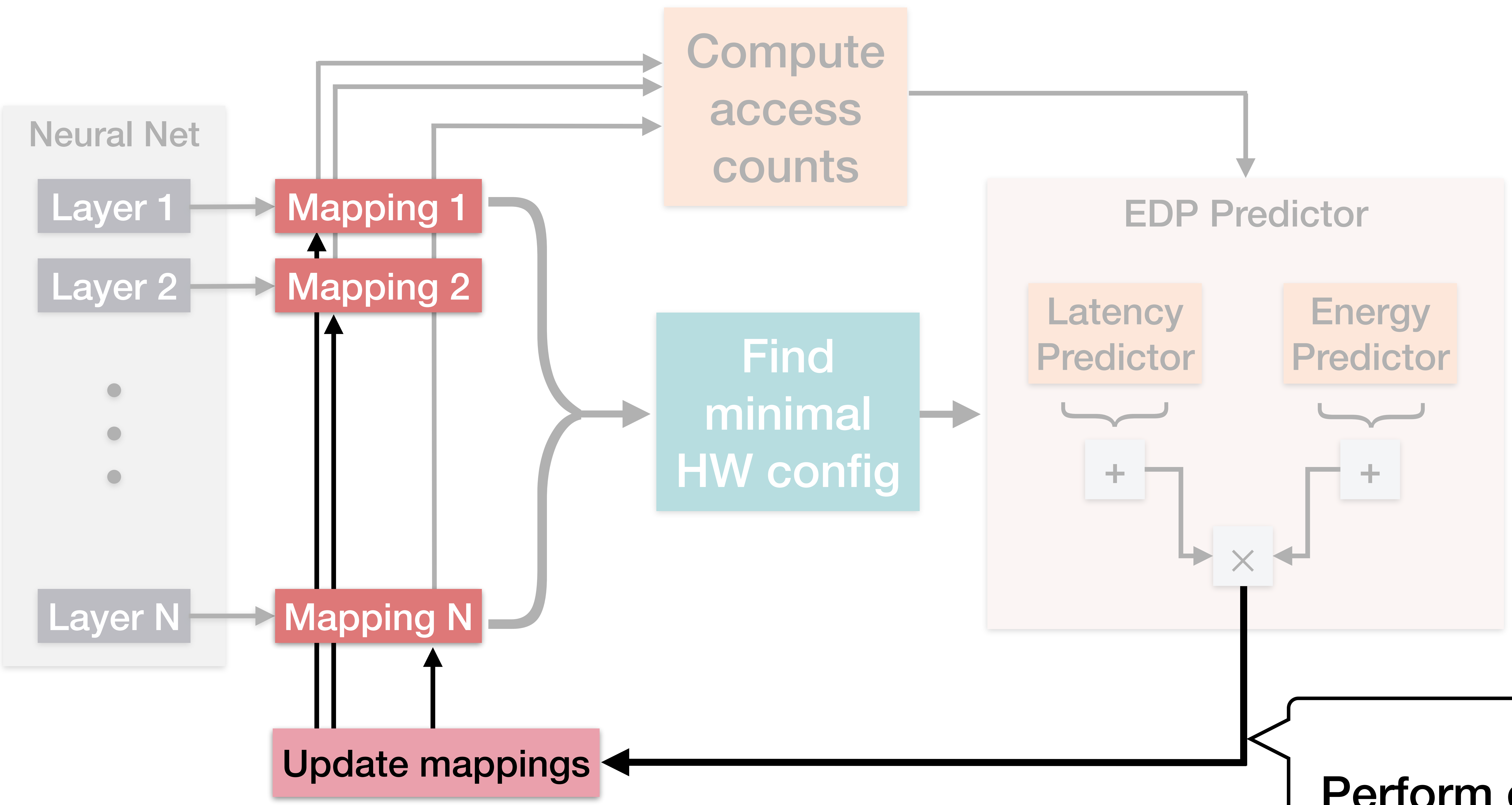
Step 3:
Predict latency and energy.

Differentiable!

DOSA Differentiable Model

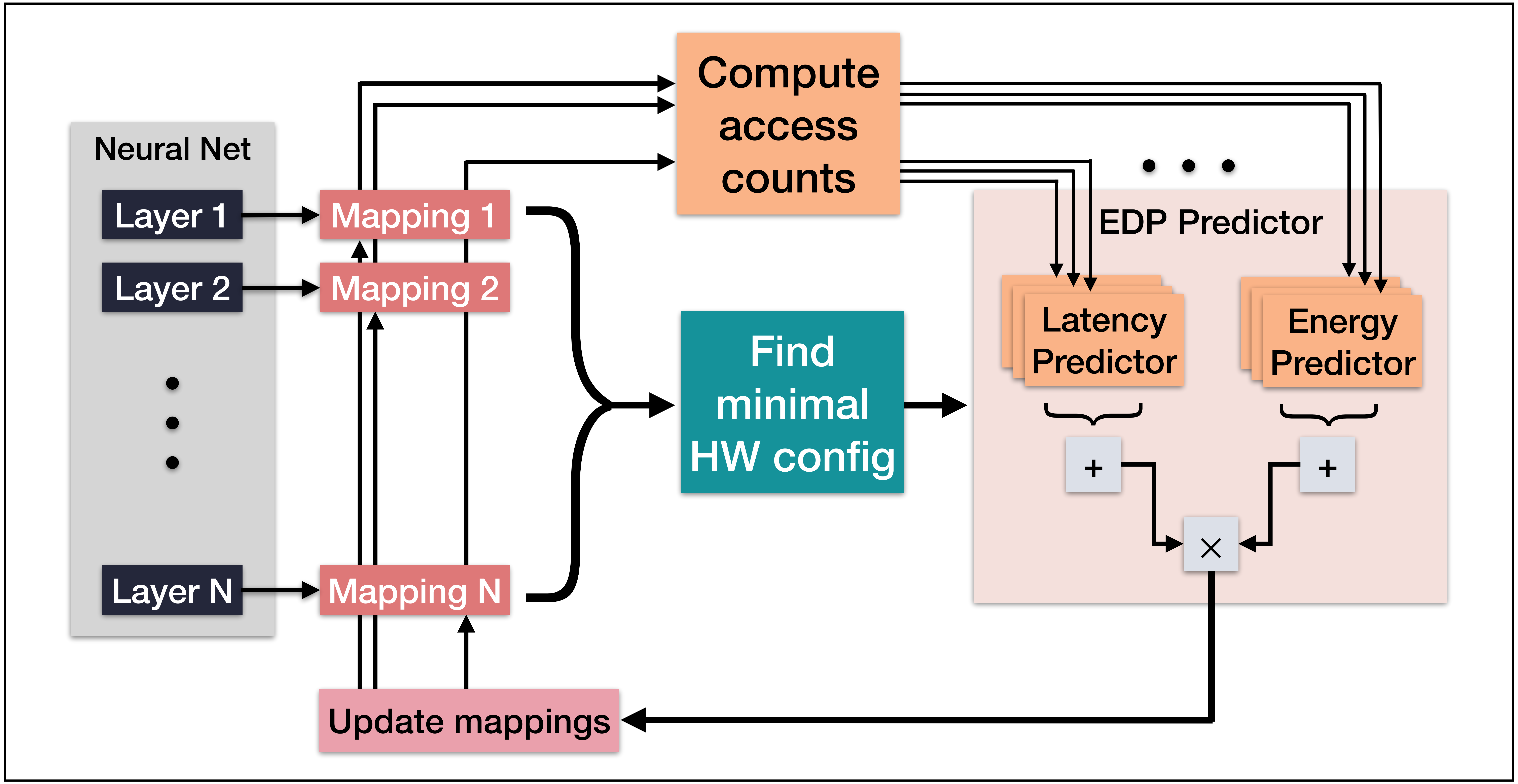


DOSA Differentiable Model

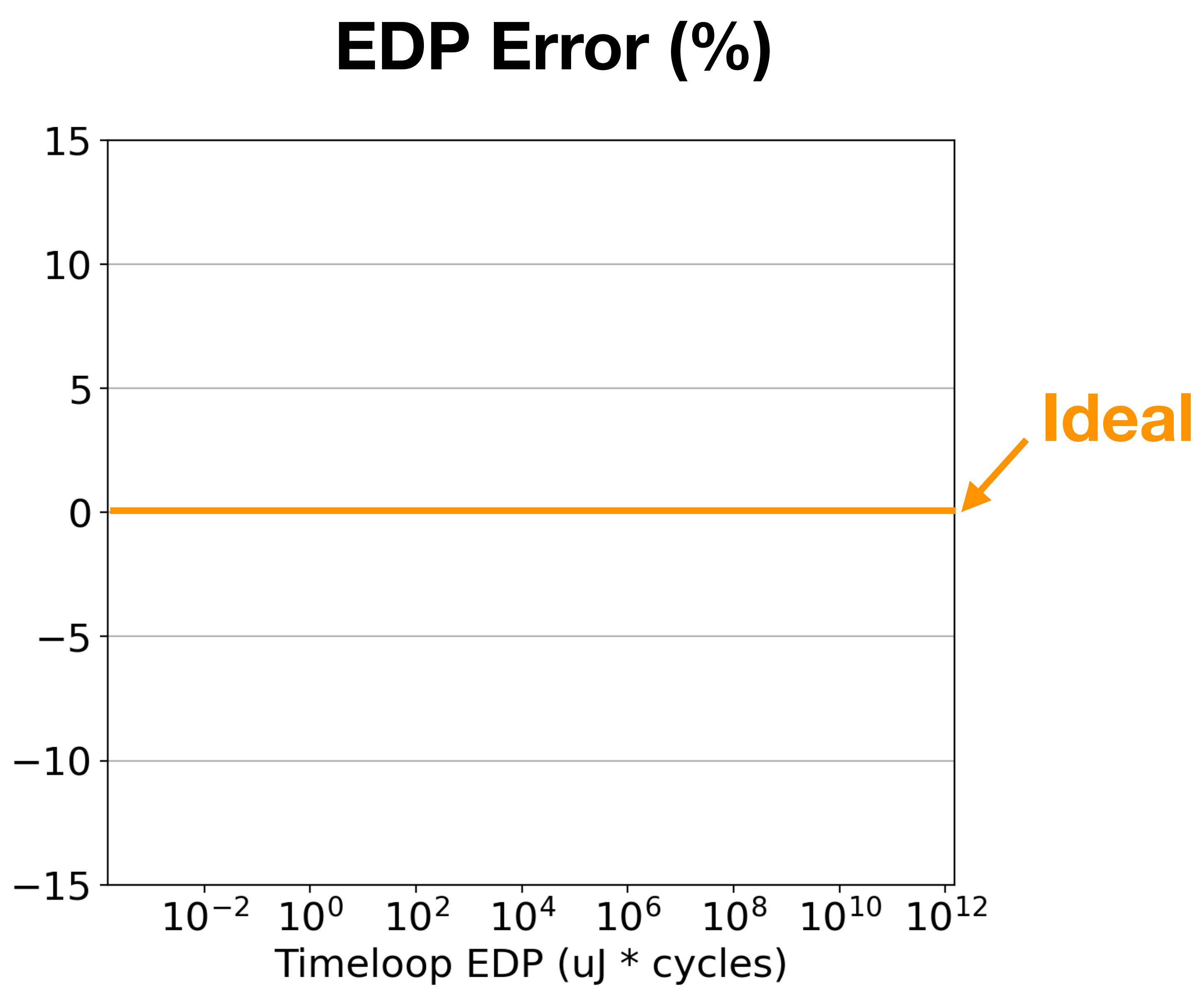


Step 4:
Perform gradient descent to minimize loss (e.g. energy-delay product).

DOSA: Differentiable Model-Based One-Loop Search for DNN Accelerators

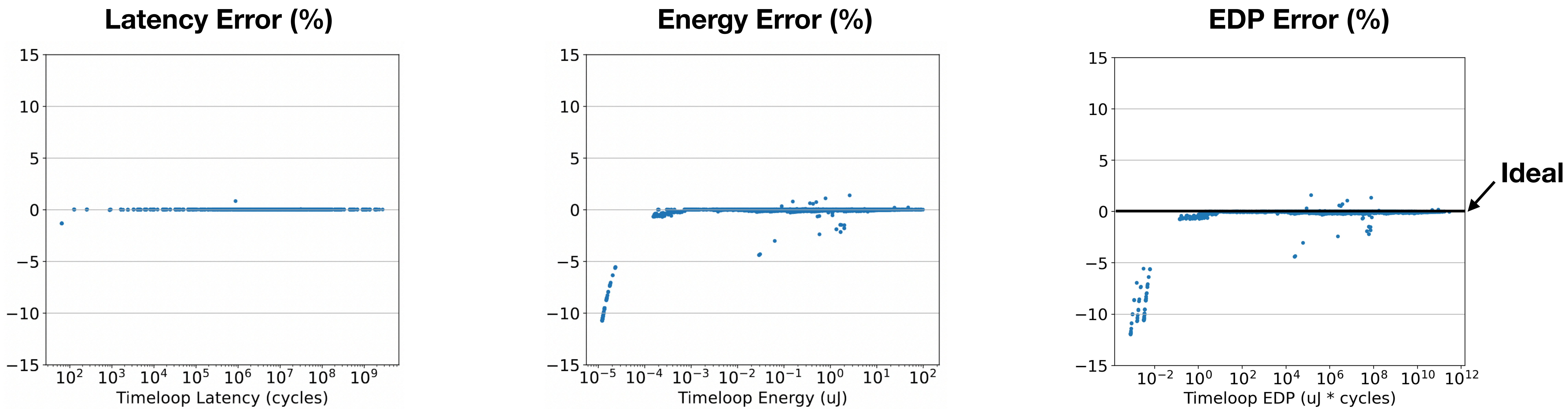


DOSA Differentiable Model: Correlation to Timeloop*



*Parashar, et al. "Timeloop: A systematic approach to DNN accelerator evaluation," International Symposium on Performance Analysis of Systems and Software (ISPASS), 2019.

DOSA Differentiable Model: Correlation to Timeloop*



Over 10,000 mappings from 73 unique layers, model is:

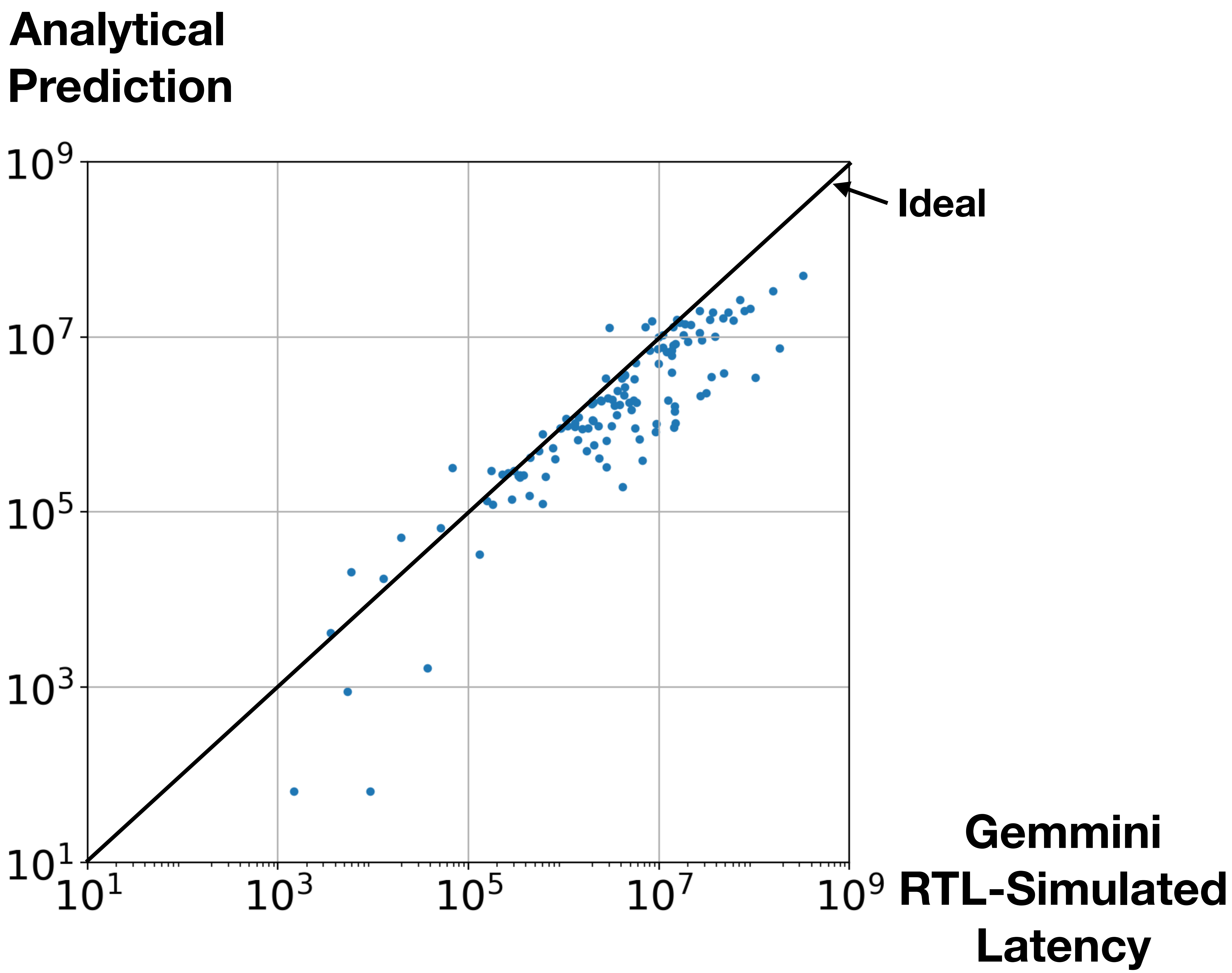
- On average, within **0.18%** of Timeloop
- Within 1% of Timeloop, for **98.3% of mappings**

*Parashar, et al. "Timeloop: A systematic approach to DNN accelerator evaluation," International Symposium on Performance Analysis of Systems and Software (ISPASS), 2019.

Going Beyond Architectural Modeling

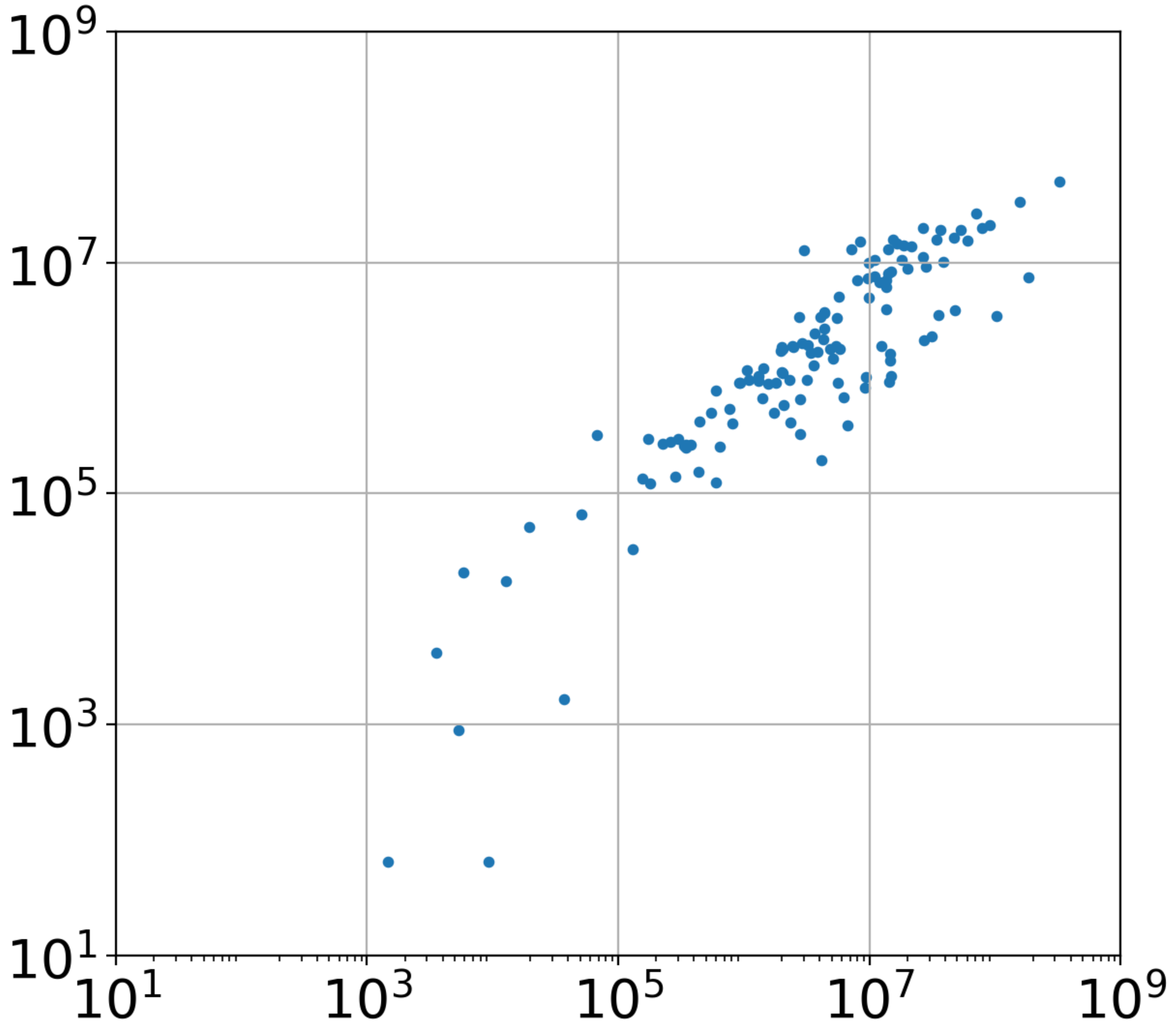
Right: Prediction accuracy of analytical model vs Gemmini RTL implementation.

- Analytical models don’t fully capture real-world performance.
- How can we improve the accuracy of our model?



Deep Learning-Based Latency Predictor: Accuracy

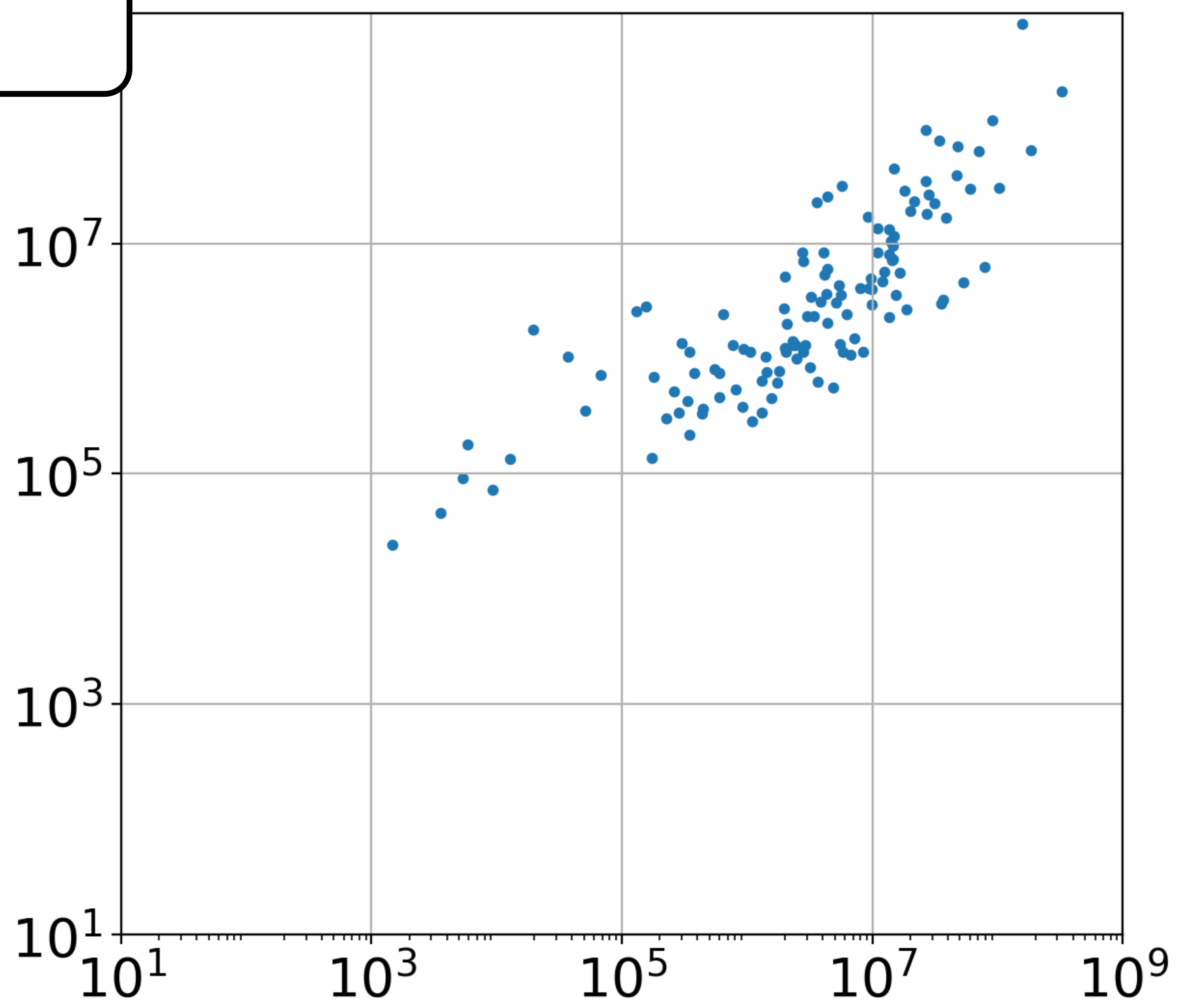
DOSA Analytical Model Prediction
Correlation = 87%



Gemmini RTL-Simulated Latency

Prediction accuracy does not improve.

DNN Model Prediction
Correlation = 84%

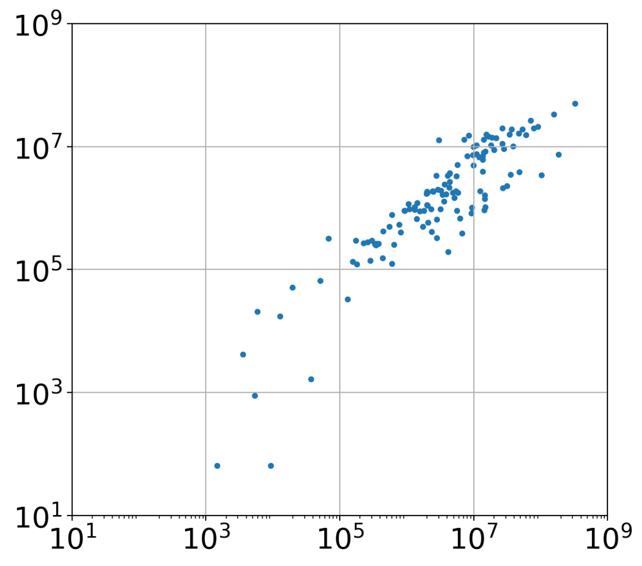


Gemmini RTL-Simulated Latency

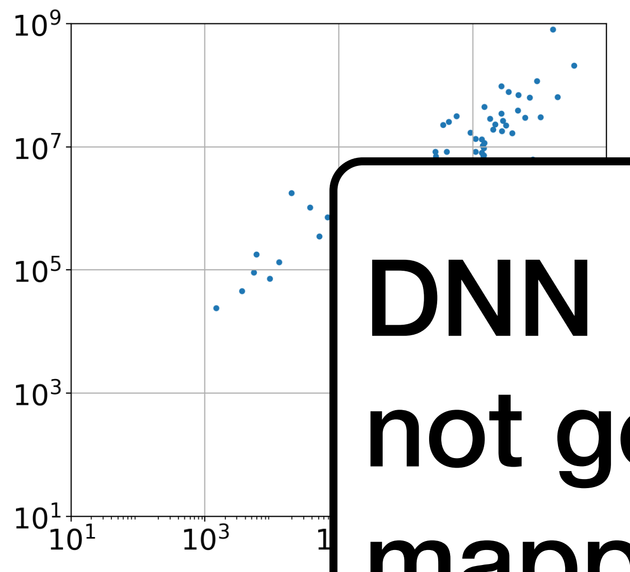
Deep Learning-Based Latency Predictor: Generalization

Training layers, test mappings

DOSA Analytical Model Prediction
Correlation = 87%

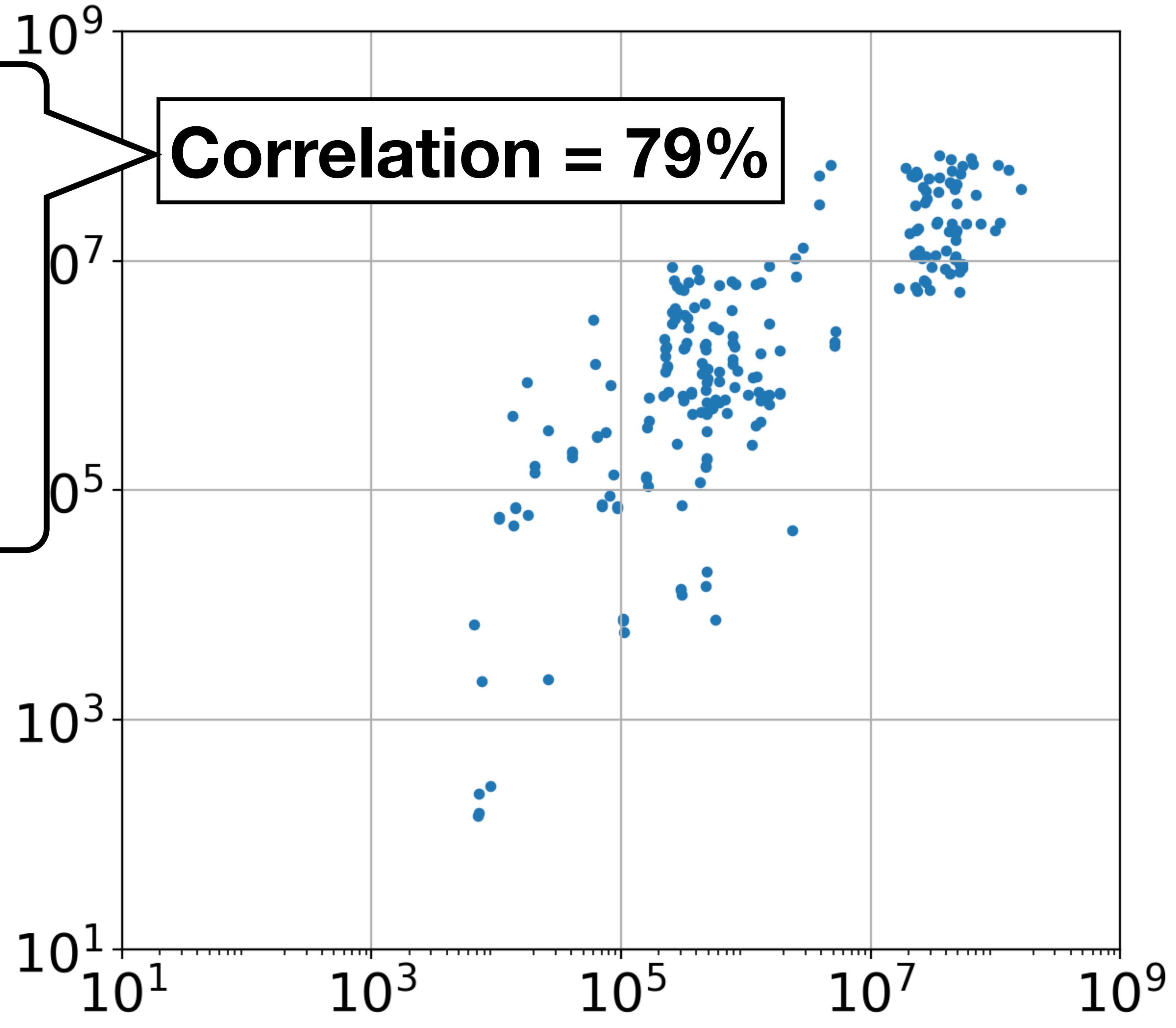


DNN Model Prediction
Correlation = 84%



Test layers, test mappings

DNN Model Prediction



DNN models also do not generalize well to mappings for unseen layers.

How does DOSA tackle this ~10^{22} search space?

1. Do mapping-first search.

2. Use differentiable, interpretable performance models.

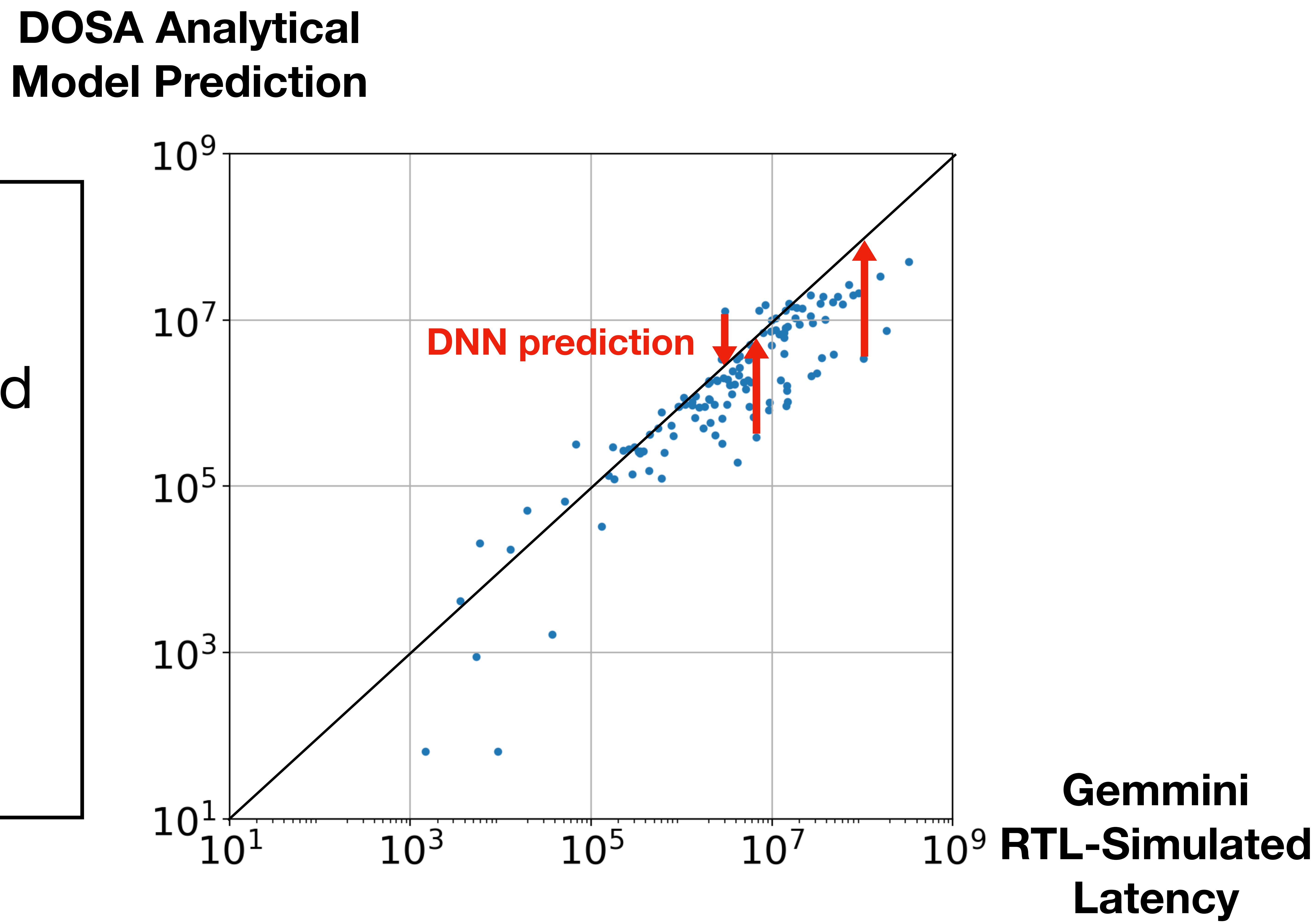
3. Apply deep learning to bridge the gap between models and RTL.

Combining Analytical and DNN Predictors

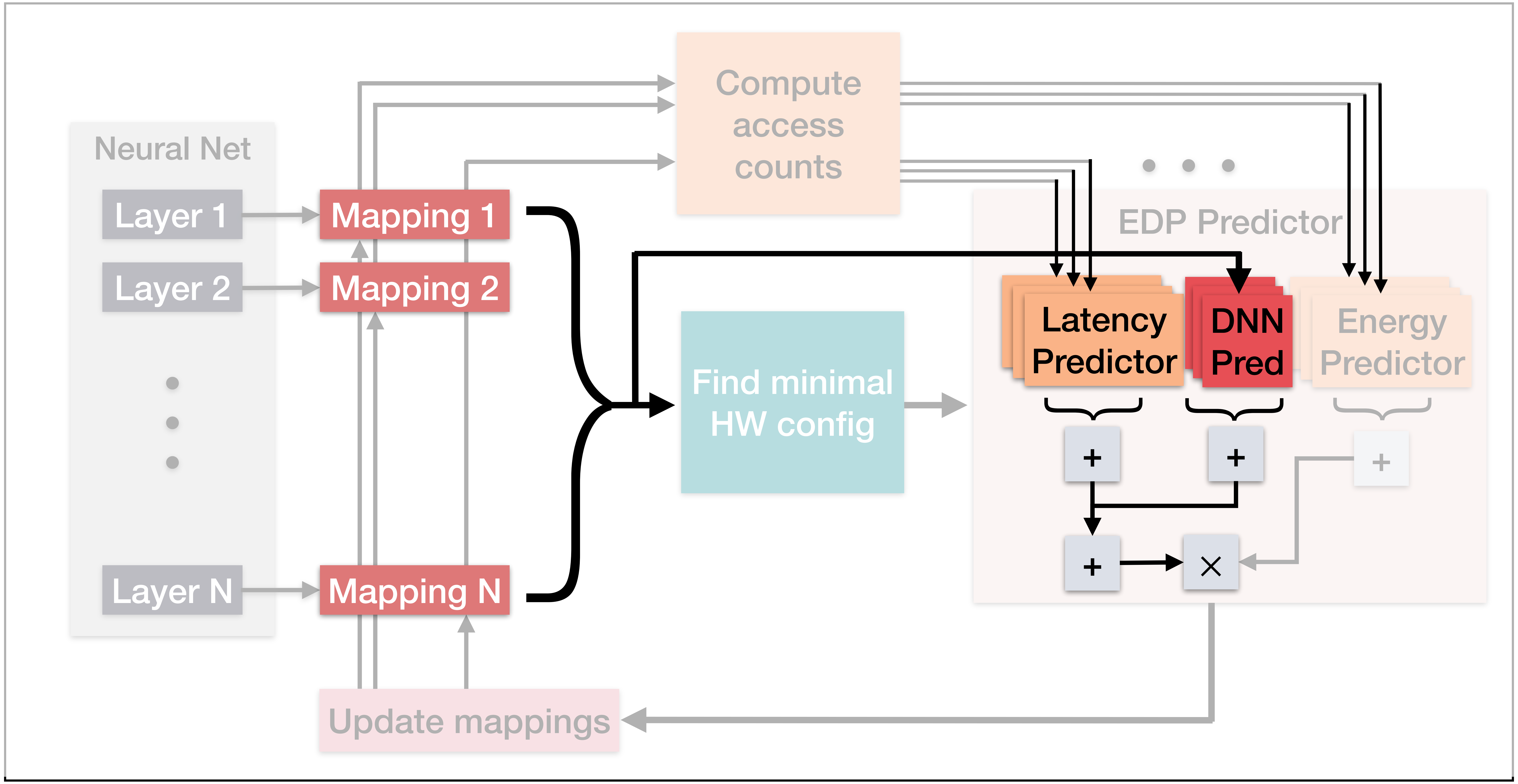
DNN model **predicts the difference** between analytical model and Gemmini RTL-simulated latency.

Trained on ~1500 mappings from:

- AlexNet, ResNeXt50-32x4d, VGG-16, DeepBench



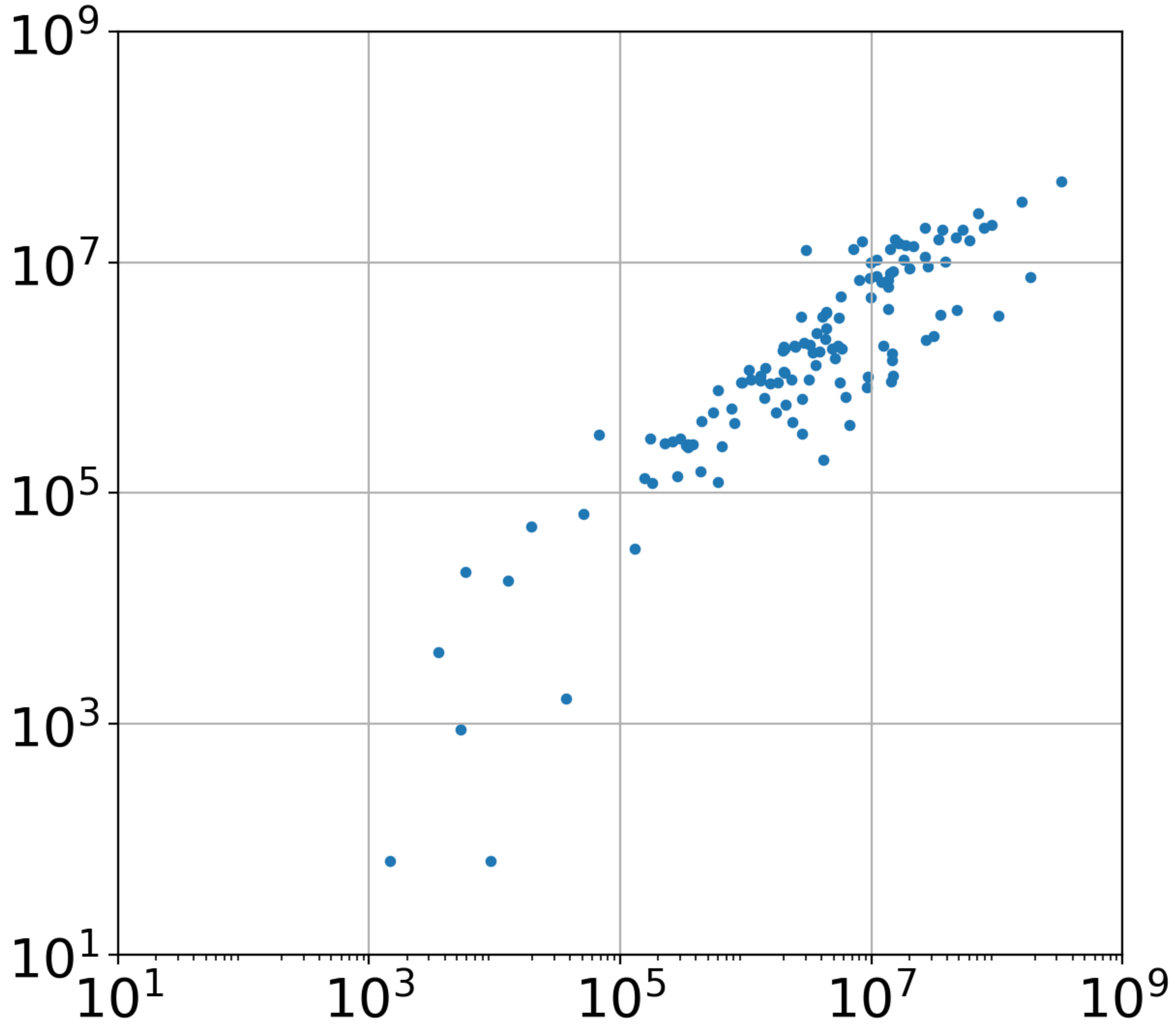
DOSA: Differentiable Model-Based One-Loop Search for DNN Accelerators



Combining Analytical and DNN Predictors: Accuracy

Training layers, test mappings

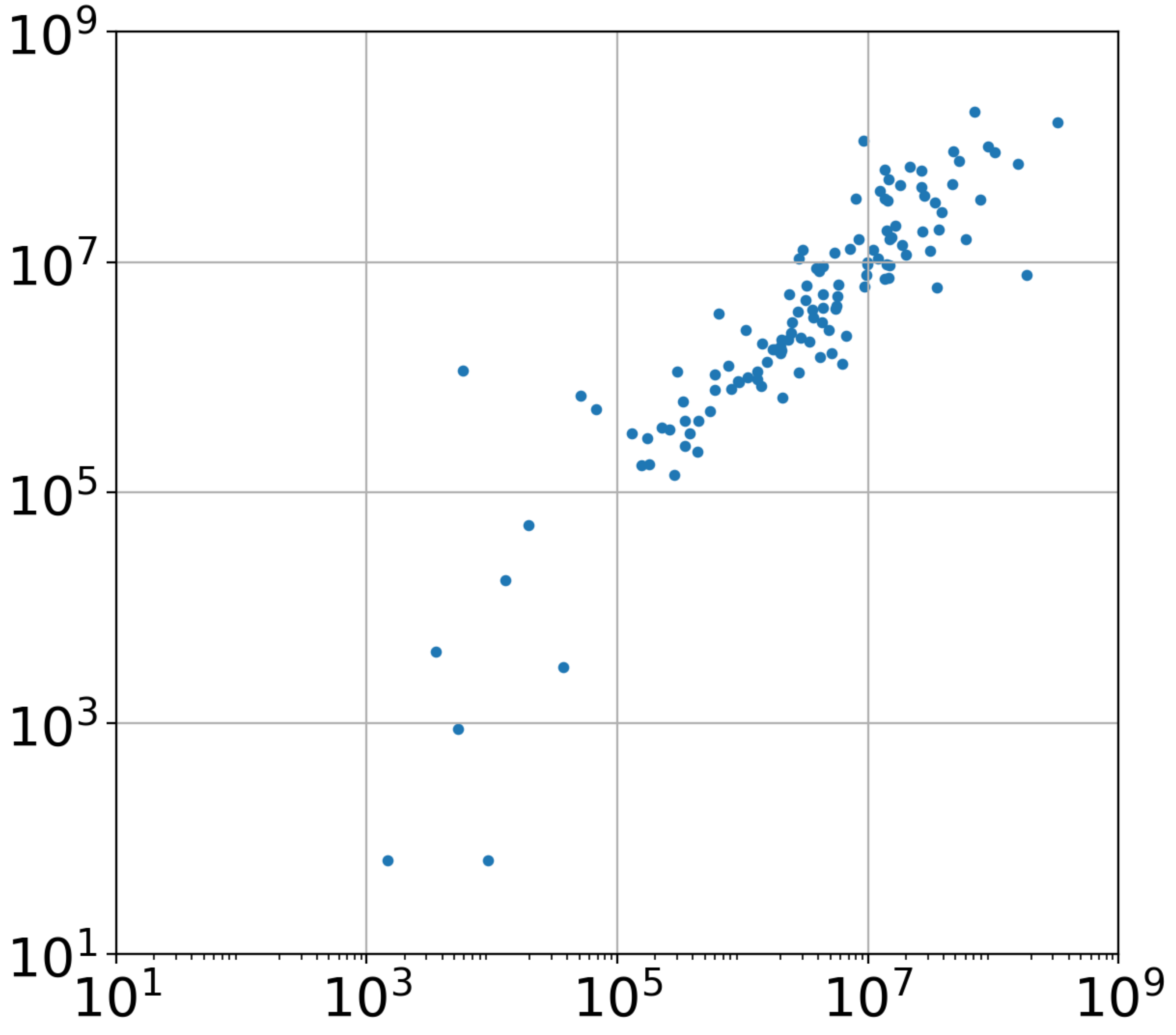
Analytical Model Prediction
Correlation = 87%



Gemmini RTL-Simulated Latency

Not only improves prediction accuracy slightly…

Analytical+DNN Model Prediction
Correlation = 92%

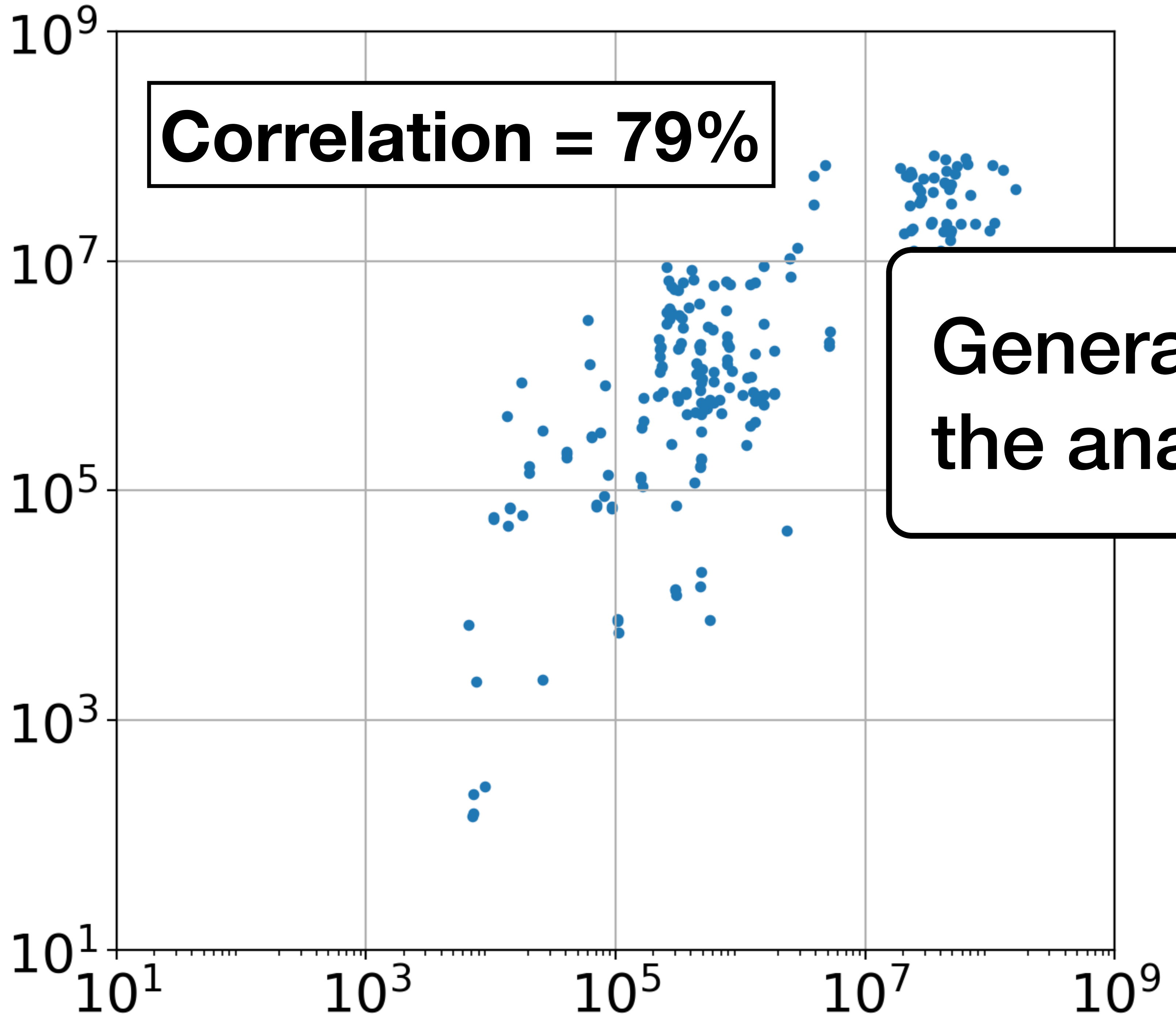


Gemmini RTL-Simulated Latency

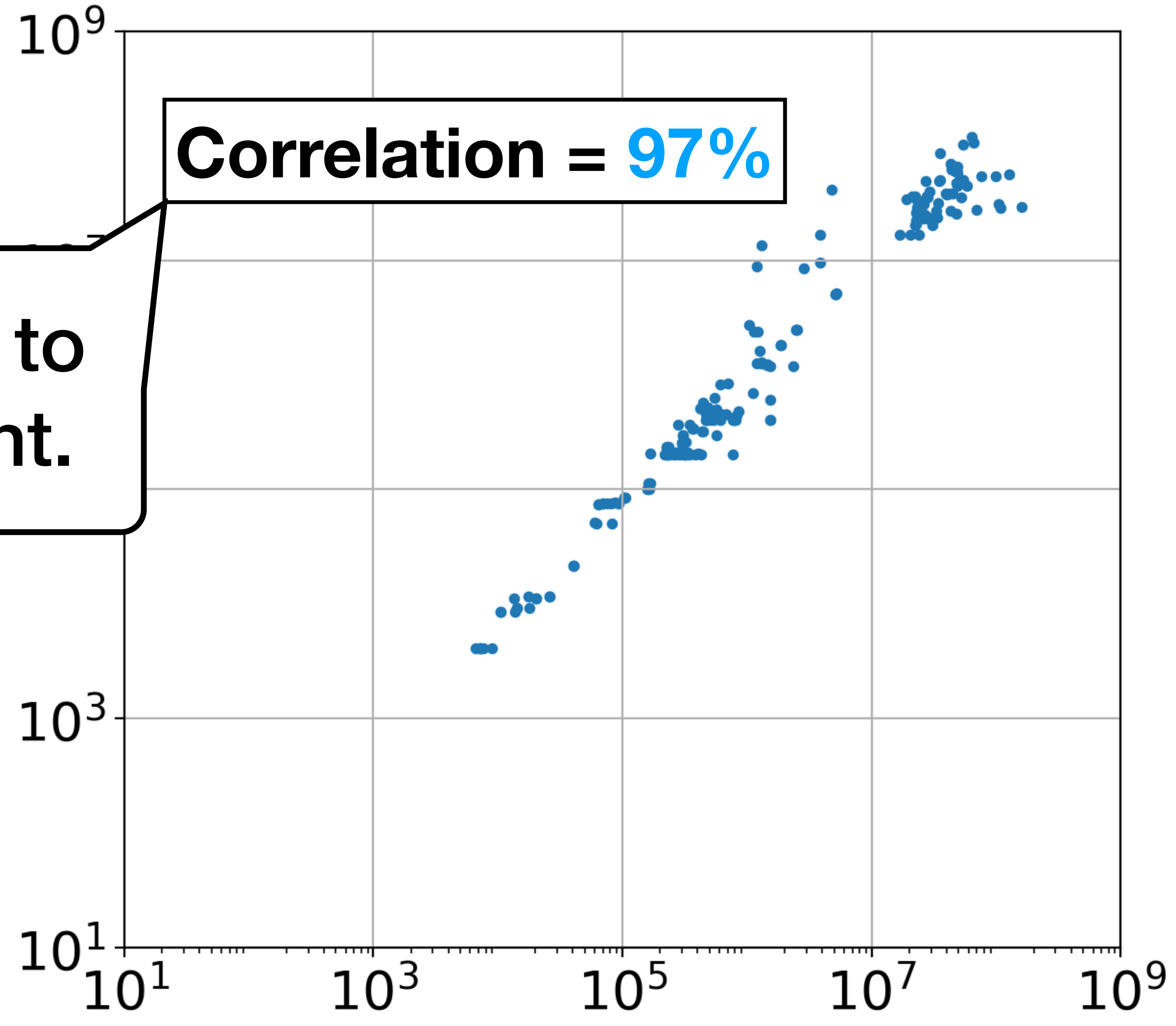
Combining Analytical and DNN Predictors: Generalization

Test layers, test mappings

DNN Model Prediction



Analytical+DNN Model Prediction



Generalizes well thanks to the analytical component.

Evaluation

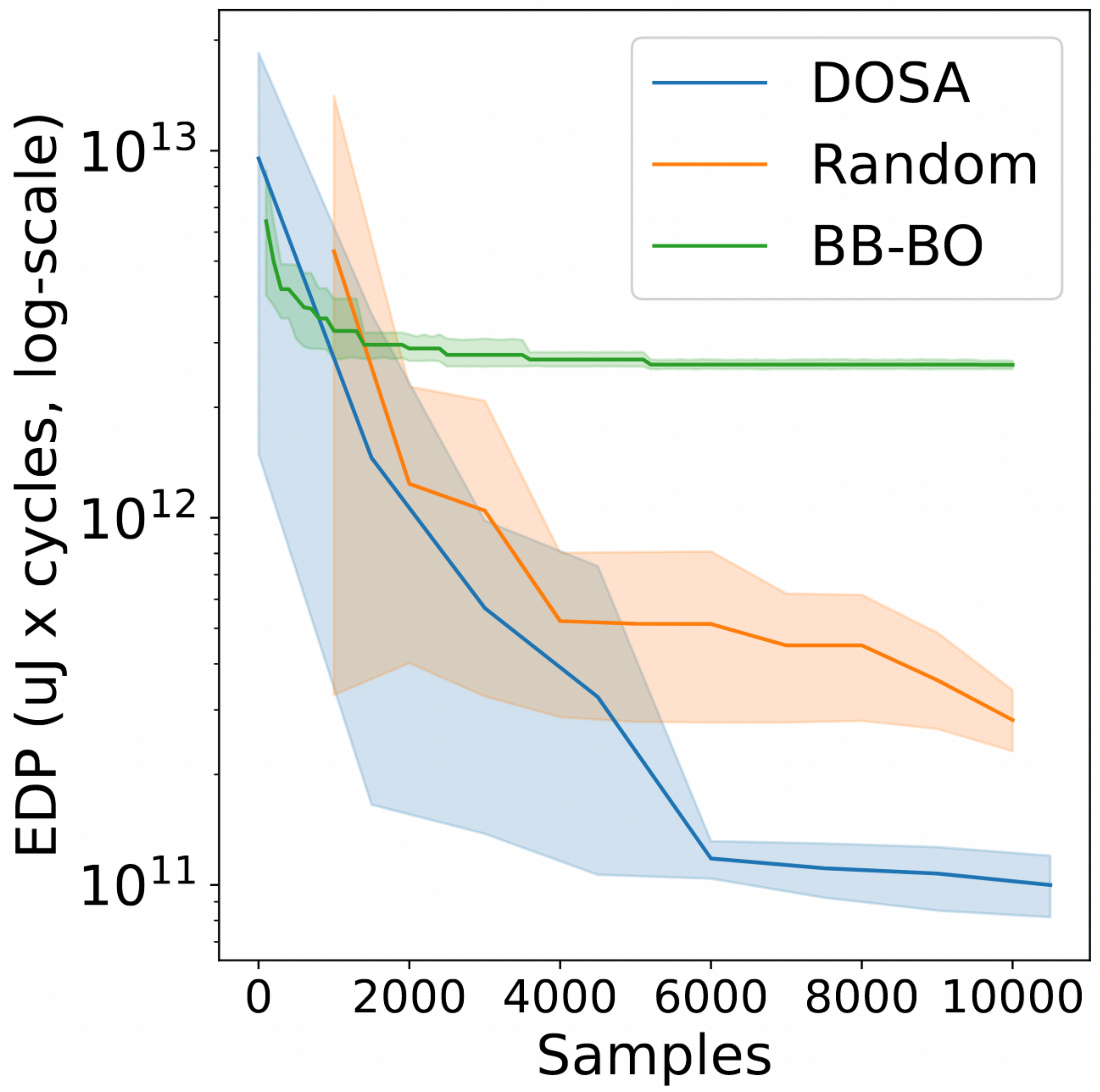
Evaluating Co-Search Performance

After 10,000 samples, DOSA finds hardware-mapping co-design points with better EDP:

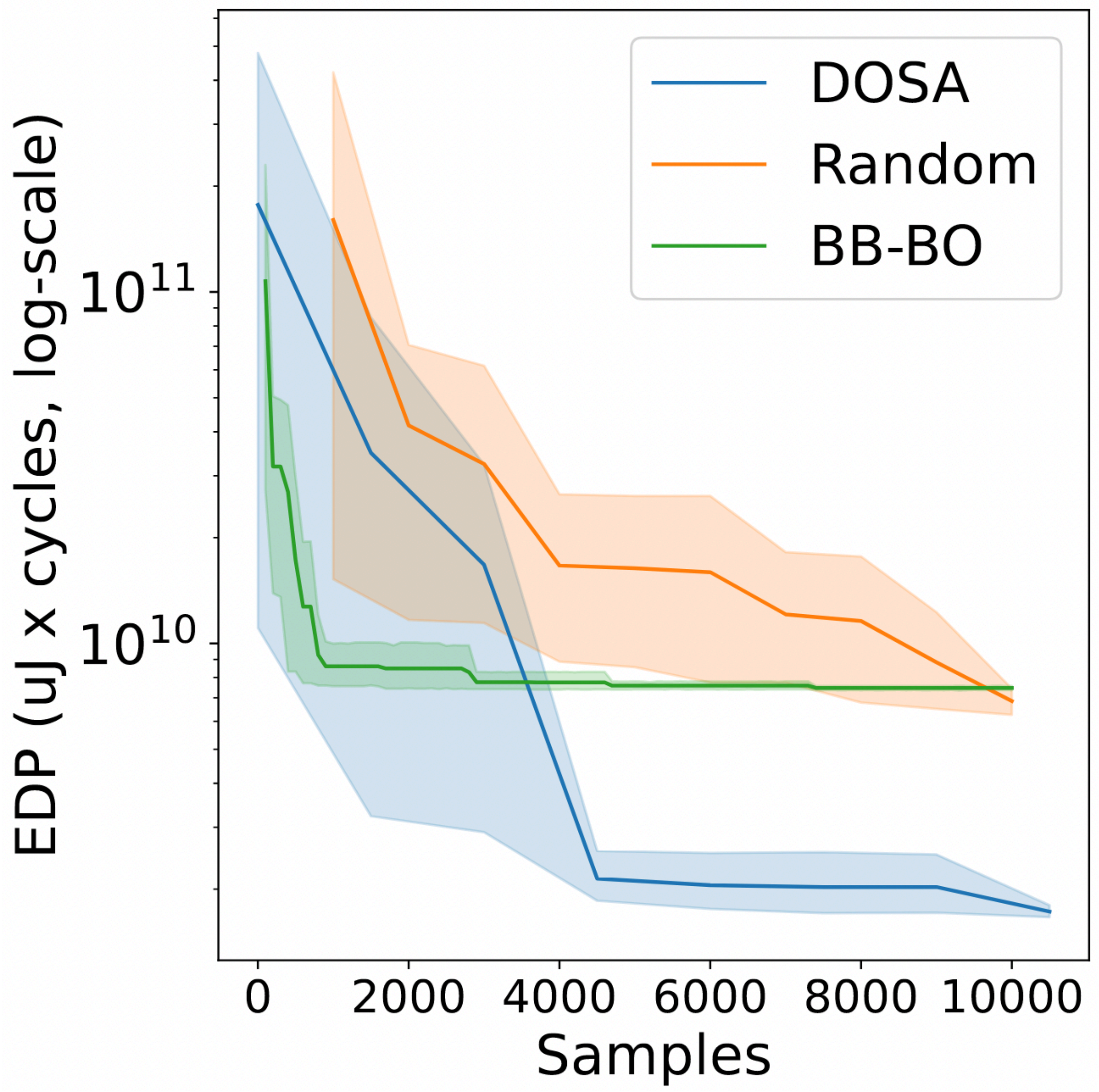
- **2.80x** vs random search
- **12.59x** vs Bayesian optimization

Latency evaluated w/ Timeloop

ResNet-50



BERT



Comparison to Baseline Accelerators

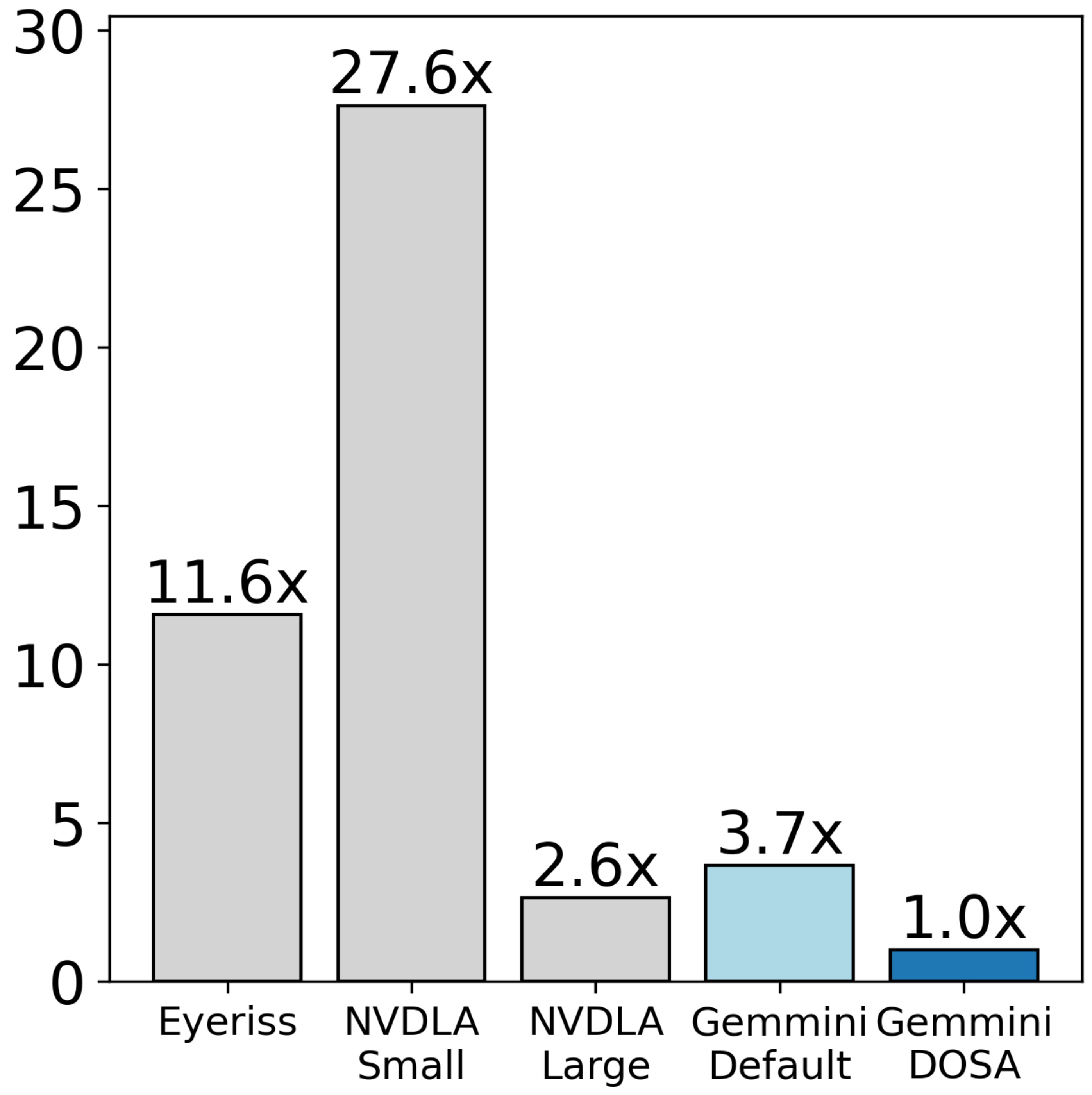
DOSA-searched design points are several times more efficient than hand-tuned baselines.

Results shown are averaged over 4 target workloads:

- U-Net, ResNet-50, BERT, RetinaNet

Latency evaluated w/ Timeloop

Normalized EDP (lower is better)

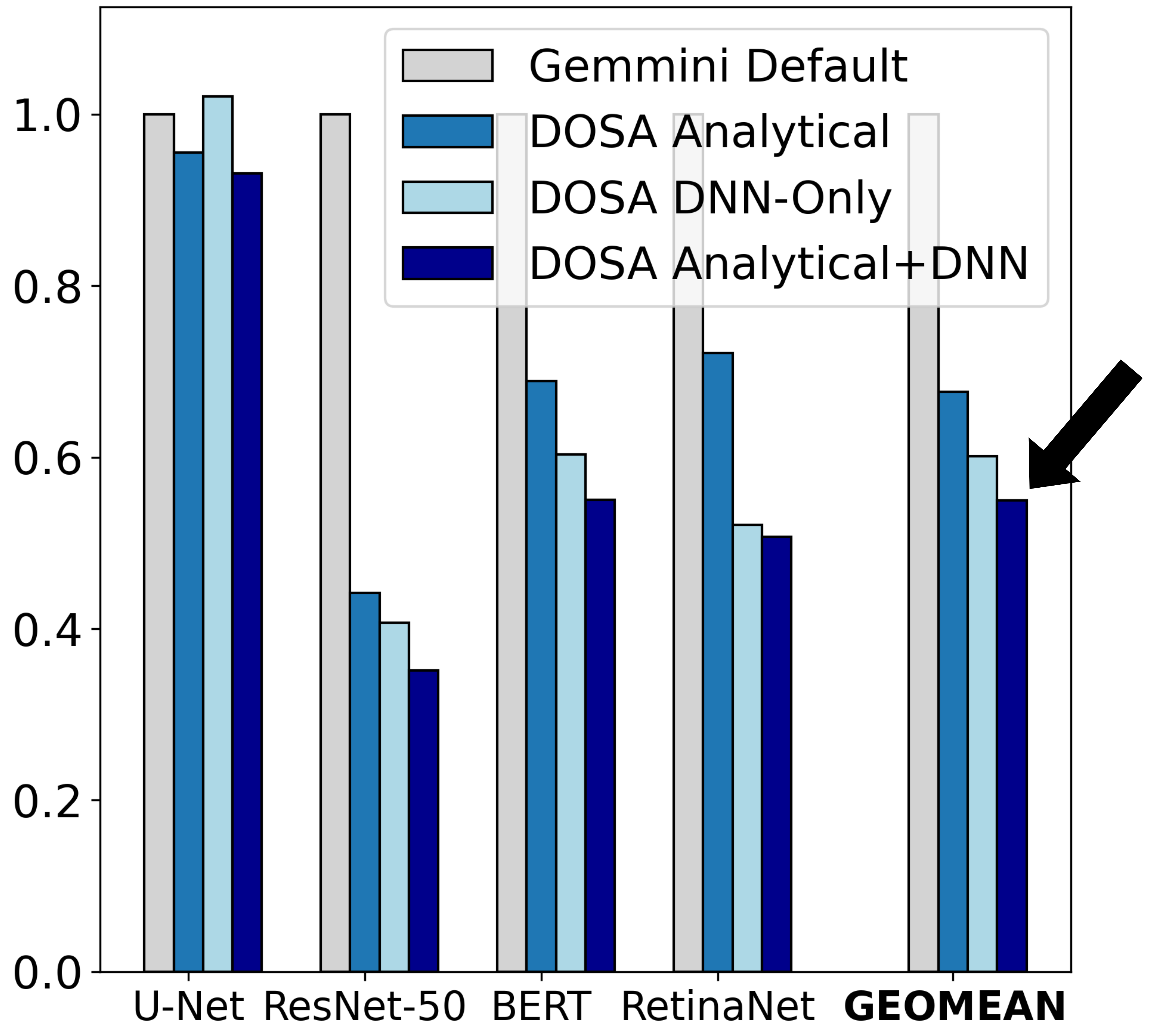


Optimizing Real Hardware (Gemmini)

With the combined analytical+DNN model, we optimize Gemmini EDP by **1.82x**.

Latency evaluated w/ RTL simulation

Normalized EDP (lower is better)



Conclusion

Paper PDF

Open-source at
github.com/ucb-bar/dosa

- ***Mapping-first one-loop*** search enables more efficient accelerator DSE.
- Accelerator performance models can be ***differentiable*** and ***interpretable***.
- We can augment analytical models with real performance data using DNNs.

- We look forward to extending DOSA to other parameters and platforms!
- Questions or feedback? charleshong@berkeley.edu