



SHANGHAI, CHINA

# ISCAS 2026

IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS

## Performance Pragma-Based Design Space Pruning and Exploration for High-Level Synthesis

Donghao Guo<sup>1</sup>, Zhe Lin<sup>2</sup>, Jianwang Zhai<sup>1</sup>, and Kang Zhao<sup>1</sup>

{guodonghao, zhajiw, zhaokang}@bupt.edu.cn, linzh235@mail.sysu.edu.cn

<sup>1</sup>Beijing University of Posts and Telecommunications

<sup>2</sup>Sun Yat-sen University



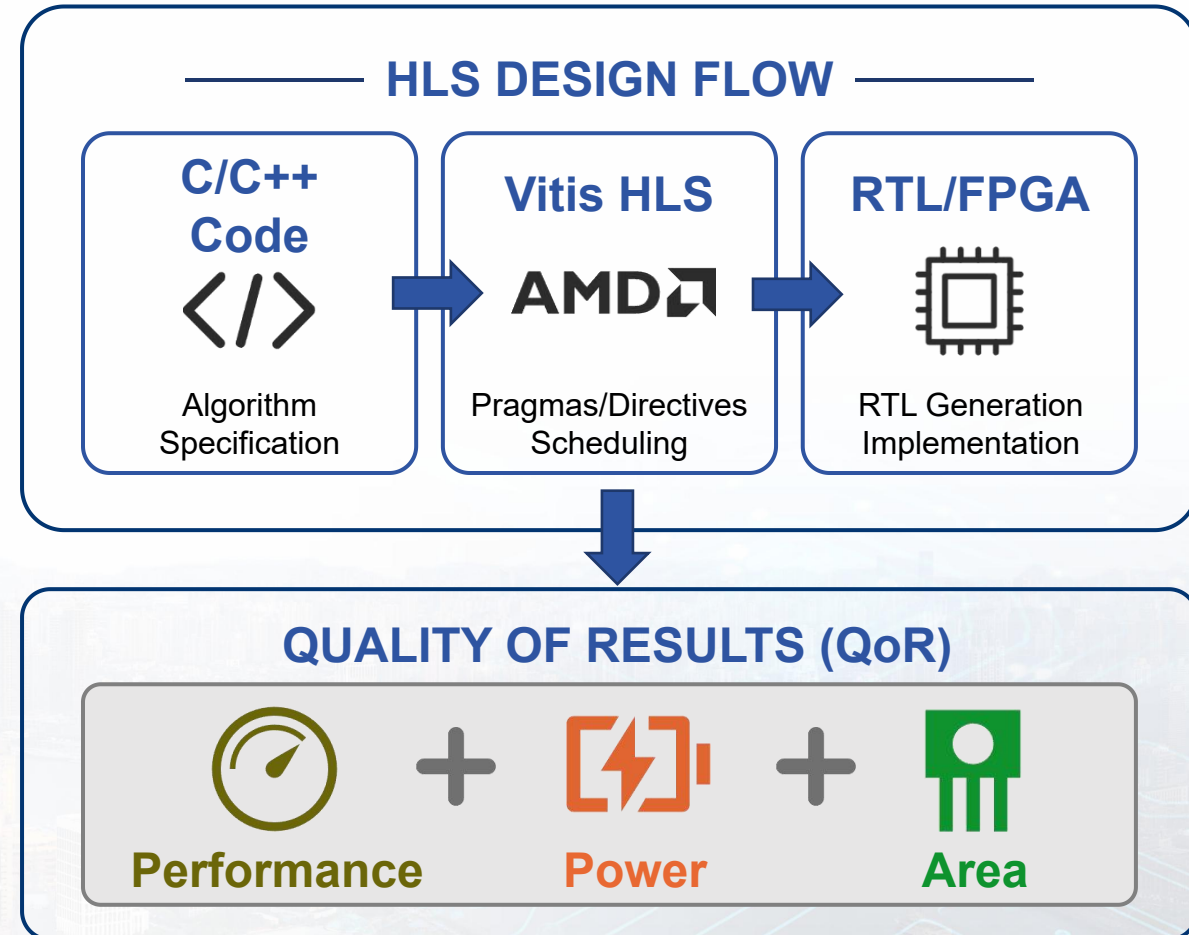
# Contents

- **Introduction**
- **Preliminaries**
- **Methodology**
- **Experiment**
- **Conclusion**

# Introduction

# Background and Motivation

- **High-level synthesis (HLS)** translates C/C++ descriptions into RTL, greatly improving FPGA/ASIC design productivity.
- HLS tools expose pragmas/directives that tune **performance, power, and area (PPA)**.
- Different pragma **types, positions, and parameters** create a huge design space.
- Efficient **design space exploration (DSE)** is therefore essential to find high-quality implementations.



# Problem Statement and Key Challenges

- **Define the design space:** choose effective pragmas without excessive dimensionality.
- **Prune unreasonable designs:** remove conflicting, invalid, or redundant pragma combinations.
- **Explore efficiently:** avoid repeated expensive HLS implementation evaluations.

a



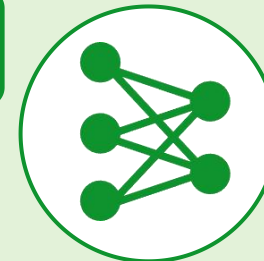
Replace traditional loop pragmas with the **PERFORMANCE** pragma.

b



Prune redundant **TARGET\_TI** values and reduce the search space.

c



Combine **GNN-based** PPA prediction with **Bayesian optimization**.

# Preliminaries

# Design Space Definition and HLS DSE Basics

## Traditional Design Space

 PIPELINE

 UNROLL

 ARRAY\_PARTITION



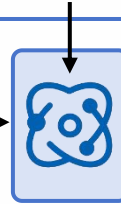
Combinations of pragma type + position + parameter quickly explode.

## PERFORMANCE Pragma

$$target\_ti = trip\_count \times II$$



C/C++  
Code



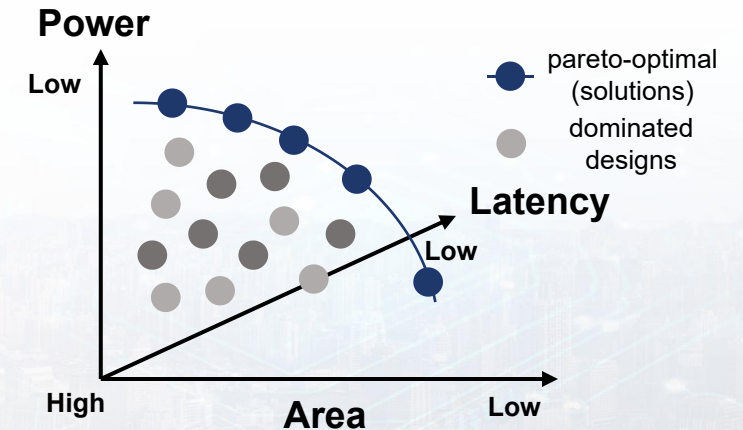
PIPELINE  
UNROLL  
ARRAY\_PARTITION



Vitis HLS infers low-level pragmas automatically.

## Multi-Objective HLS DSE

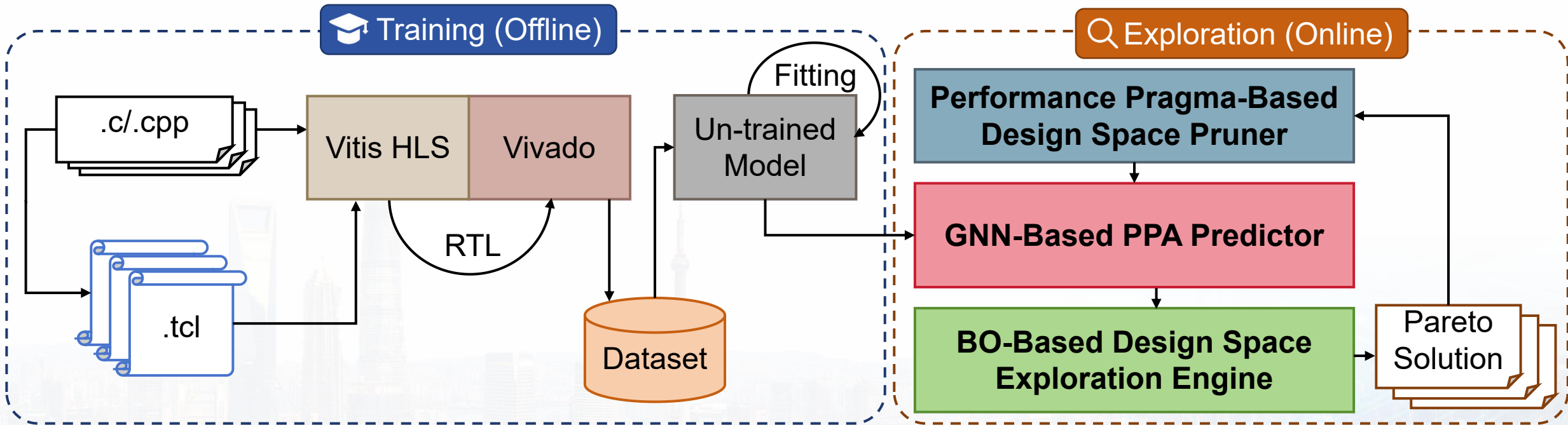
Optimize PPA, seek pareto-optimal solutions.



**Key insight:** A better design space definition can improve both **exploration efficiency** and **solution quality**.

# Methodology

# Proposed HLS DSE Framework



## 1 Construct the pruned search space

Use performance pragmas to eliminate unpromising regions.

## 2 Train a QoR predictor

Build a GNN model using HLS synthesis data samples.

## 3 Explore promising candidates

Leverage BO with the predictor to efficiently search the space.

## 4 Output high-quality pareto-optimal solutions

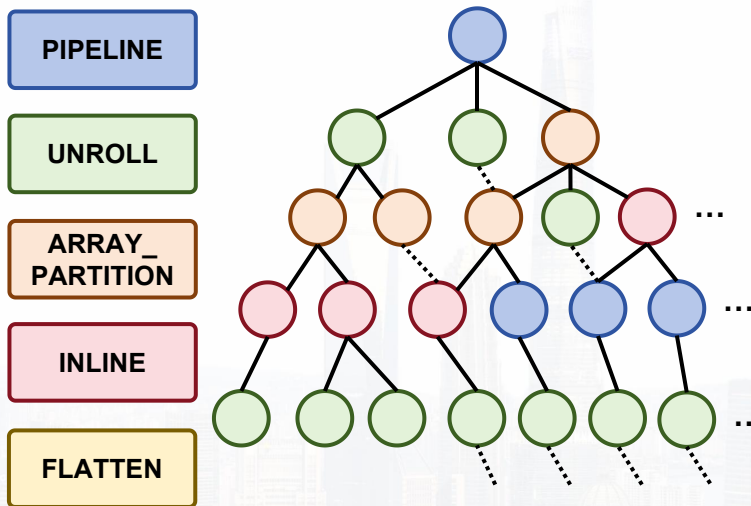
Provide trade-off optimal designs across PPA objectives.



**Pruning + Prediction + BO = Efficient HLS DSE**

# Design Space Pruner

## 1 Why traditional spaces are inefficient

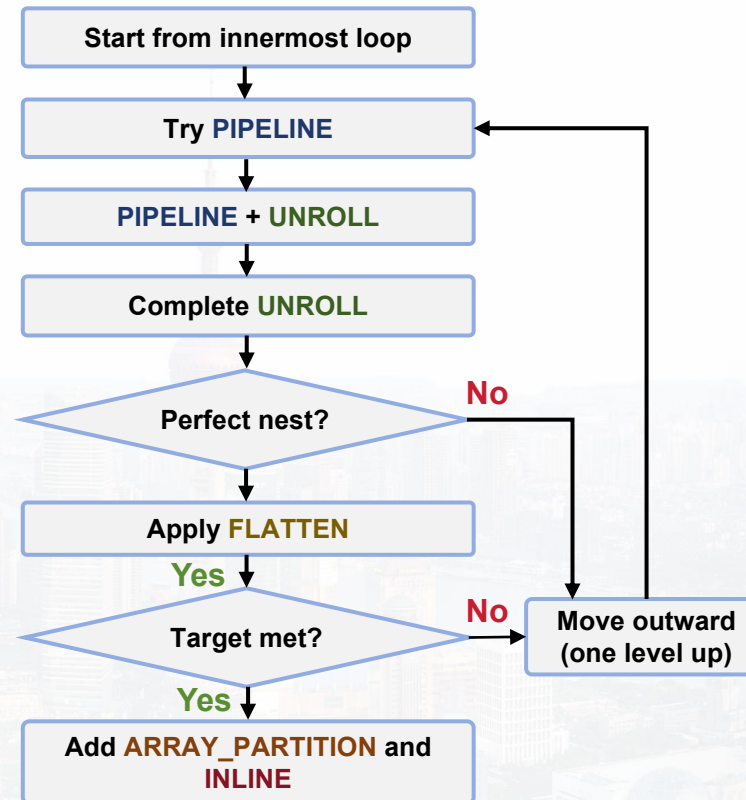


✗ unreasonable combinations

🔄 redundant designs

🔍 large search space

## 2 Low-level pragma allocation logic



## 3 Valid TARGET\_TI pruning

$$target\_ti = \frac{tripcount_i \times II_i}{\prod_{j=i}^n unroll\_factor_j}$$

🎯 Only valid target\_ti values are kept.

🗑️ Redundant values mapping to the same design are removed.

Many candidate target\_ti → Valid target\_ti kept

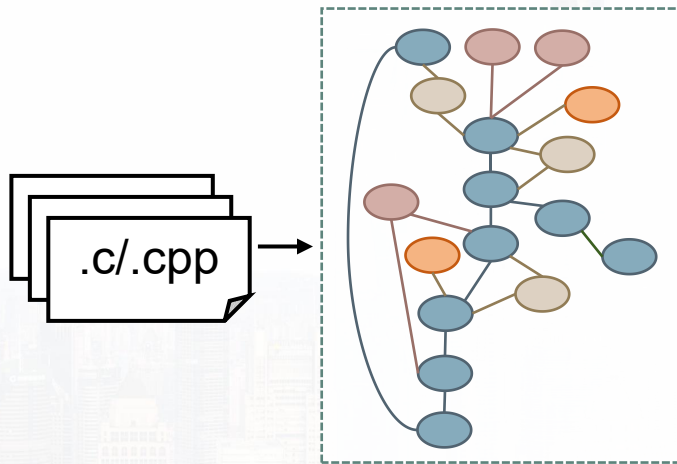


**Outcome:** A substantially reduced design space scale.

# GNN-Based PPA Predictor

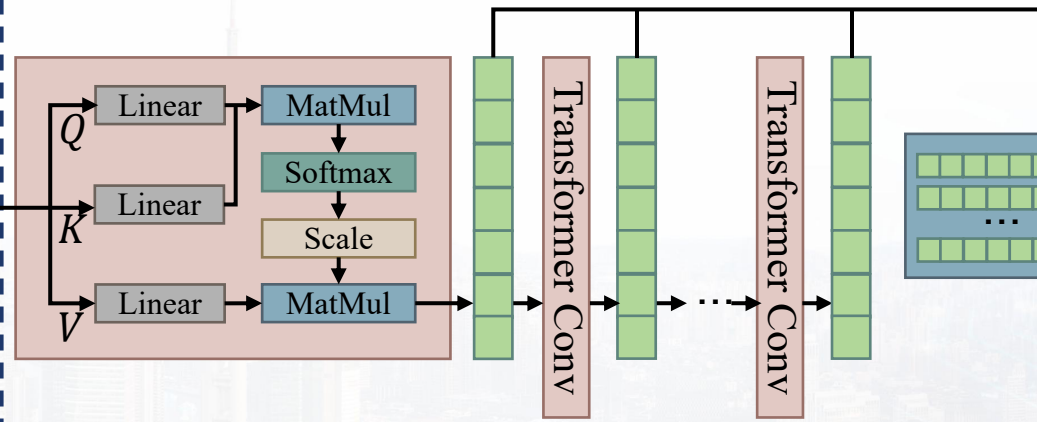
## 1 Graph Construction

Convert C/C++ into an augmented program graph.



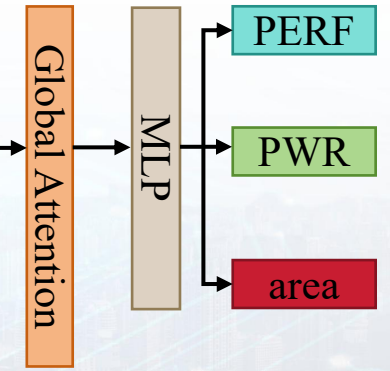
## 2 Feature Extraction

Learn rich representation with TransformerConv layers.

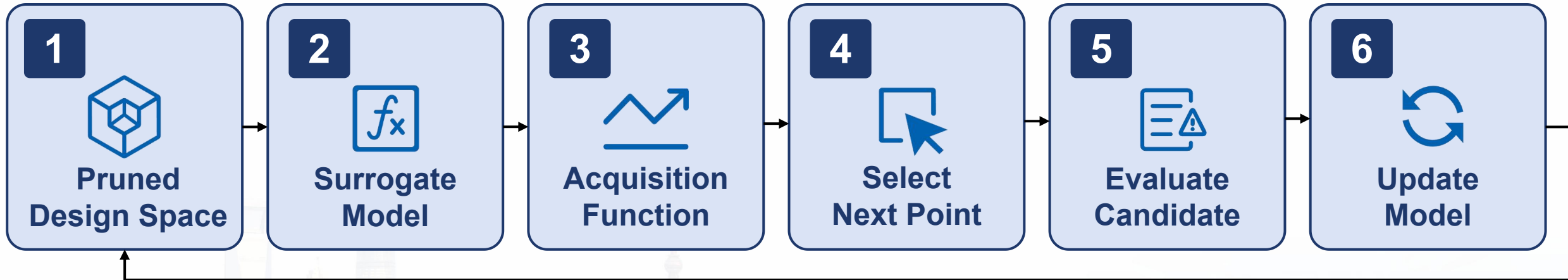


## 3 PPA Prediction

Predict key Qor metrics for fast evaluation.



# BO-Based Design Space Exploration Engine



- Bayesian optimization (BO) is a **probabilistic framework** for multi-objective optimization.
- It constructs a **surrogate model** to probabilistically characterize the objective function.
- The acquisition function (AF) **balances exploitation and exploration**.
- We adopt **Expected Improvement (EI)** as the AF.

Expected Improvement (EI)

$$EI(x) = \int_{-\infty}^{+\infty} \max(y^* - y, 0) p(y|x) dy$$

$y^*$  denotes the best objective value among explored points.

# Experiment

# Experimental Setup

## ● Benchmarks

- 10 kernels selected from **Machsuite Benchmark** for experiments
- 5 additional kernels selected from **PloyBench** for training
- 500 random pragma combinations generated per kernel

## ● Predictor Training

- 6 TransformerConv layers
- 500 epochs | batch size 64
- Adam optimizer | initial learning rate 0.001

## ● Platform

- Ubuntu 24.04.1 LTS
- Intel(R) Xeon(R) Platinum 8383C CPU
- 4 NVIDIA GeForce RTX 3090 GPUs



## Evaluation



**Performance:** CP achieved post-implementation



**Power:** total on-chip power (W)



**Area:** sum of normalized utilization rates of LUTs, FFs, DSPs, and BRAMs

$$ADRS(\Gamma, \Omega) = \frac{1}{|\Gamma|} \sum_{\gamma \in \Gamma} \min_{\omega \in \Omega} f(\gamma, \omega)$$

The reference set contains pareto-optimal solutions aggregated from all compared methods.

# ADRS Comparison of the Pareto Fronts

## ● Tool Flow

- Vitis HLS 2025.1 + Vivado 2025.1
- targeting Xilinx Virtex-7 VC707 part xc7vx485tffg1761-2 at 100 MHz

## ● Baseline

- Simulated Annealing (SA)<sup>1</sup>
- Non-dominated Sorting Genetic Algorithm (NSGA-II)<sup>2</sup>
- **HGBO** (MOTPE-FL)<sup>3</sup>

<sup>1</sup>A. Mahapatra and B. C. Schafer, "Machine-learning based simulated annealer method for high level synthesis design space exploration," Proceedings of the 2014 Electronic System Level Synthesis Conference (ESLsyn), pp. 1–6, 2014.

<sup>2</sup>B. C. Schafer, "Parallel high-level synthesis design space exploration for behavioral ips of exact latencies," ACM Trans. Des. Autom. Electron. Syst., vol. 22, no. 4, 2017.

<sup>3</sup>H. Kuang, X. Cao, J. Li, and L. Wang, "Hgbo-dse: Hierarchical gnn and bayesian optimization based hls design space exploration," 2023, pp. 106–114.

Benchmark	SA	NSGA-II	HGBO	Ours
aes	<b>0.0753</b>	0.6531	0.1546	0.1272
bfs_bulk	0.5173	0.6385	0.3768	<b>0.1334</b>
fft_strided	0.4931	0.4999	0.3011	<b>0.2630</b>
gemm_ncubed	0.7612	0.7079	0.5517	<b>0.0529</b>
md_knn	0.6455	0.4073	0.2306	<b>0.1986</b>
nw	0.3627	0.4103	0.1457	<b>0.1106</b>
sort_radix	0.0001	0.0173	<b>0.0001</b>	0.0001
spmv_ellpack	0.6309	0.2895	0.1249	<b>0.0999</b>
stencil3d	0.5348	0.5659	0.3495	<b>0.3192</b>
viterbi	0.6643	0.6535	0.3366	<b>0.2821</b>
<b>Average</b>	0.4685	0.4843	0.2572	<b>0.1587</b>
<b>Improv. over SA</b>				<b>66.13%</b>
<b>Improv. over NSGA-II</b>				<b>67.23%</b>
<b>Improv. over HGBO</b>				<b>38.30%</b>

# Conclusion

# Conclusion

- Our work can be summarized **as follows**:
  - Replace traditional loop optimization pragma combinations with the **PERFORMANCE pragma**, deduce the algorithmic logic underlying the PERFORMANCE pragma, further prune the design space, achieving a significant reduction in its scale.
  - Propose an HLS DSE framework integrating an accurate **GNN-based** PPA predictor and an efficient **BO-based** design space exploration engine.
  - Experimental results demonstrate that our framework obtains superior pareto-optimal sets compared to SOTA methods under the same time constraint.
- **Future work** will focus on:
  - Expand the current design space by incorporating pragmas of **different optimization granularities**.
  - Explore better QoR prediction models and DSE methodologies to enhance the performance of our framework.
  - Combining **RAG with LLM** to further prune the design space and perform early candidate selection during the exploration phase.

**THANK YOU!**