CausalTuner: Will Causality Help High-Dimensional EDA Tool Parameter Tuning

Ziyang Yu*,[1], Peng Xu*,[1], Su Zheng[1], Siyuan Xu[2], Hao Geng[3], Bei Yu[1], Martin Wong[4]
[1]CUHK [2]Huawei Noah's Ark Lab [3]ShanghaiTech University [4]HKBU

***Abstract*—Electronic Design Automation (EDA) tools are central to Very Large Scale Integration (VLSI) design, where numerous parameters govern the Quality-of-Result (QoR) metrics, including performance, power, and area. The high dimensionality of the parameter space, coupled with complex interactions, makes manual tuning inefficient and hinders the scalability of automated methods. Existing methods typically treat parameters as flat vectors, neglecting the EDA flow's hierarchical causal structure, where early-stage decisions constrain later downstream stages. To address this, we propose CausalTuner, a causality-aware design space exploration framework for efficient parameter tuning. It employs a hybrid causal attention mechanism to capture stage-wise parameter interactions and embeds them into deep kernel Gaussian processes for accurate and generalizable surrogate modeling. The causal exploration strategies enhance sampling efficiency. Experiments show that CausalTuner outperforms state-of-the-art methods in both final QoR and efficiency.**

I. Introduction

Electronic Design Automation (EDA) tools are critical across the VLSI design flow, with front-end and back-end tools offering numerous configurable parameters that significantly impact quality-of-result (QoR) metrics such as timing, power, and area. These parameters affect the design sequentially, with synthesis settings influencing logic structure and layout constraints, while physical design parameters govern placement and routing quality. Despite their importance, parameter tuning remains challenging due to the absence of analytical QoR models and the exponential growth of the design space (up to 10^{70} combinations [1]). Manual tuning is labor-intensive and relies on domain expertise, while exhaustive search is computationally prohibitive. Consequently, automated tuning strategies are essential to enhance productivity, reduce time-to-market, and ensure consistent QoR across diverse designs.

Recent advances in EDA parameter tuning span heuristic search, surrogate modeling, and Bayesian optimization (BO). Heuristic methods such as evolutionary strategies [2], [3] offer population-based exploration but scale poorly in high-dimensional, mixed-type spaces. Recommender [4] adopts tensor decomposition techniques from recommender systems and employs a neural network to suggest effective parameter configurations. FIST [5] integrates feature importance sampling with XGBoost regression to improve sampling efficiency. In contrast, a growing number of recent approaches are built upon Bayesian optimization (BO) frameworks due to their sample efficiency and ability to balance exploration and exploitation.

The project is supported in part by Research Grants Council of Hong Kong SAR (No. RFS2425-4S02, CUHK14211824, and CUHK14210723).

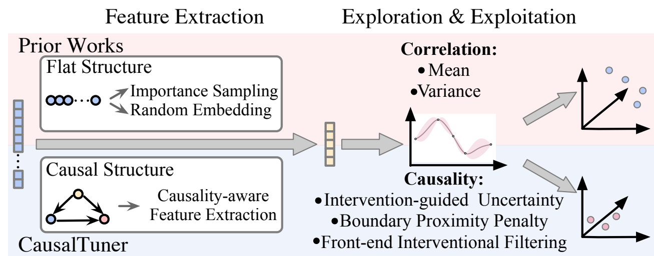


Fig. 1 Prior works consider parameters as a flat structure and incorporate correlation only (upper row). In contrast, CausalTuner leverages causal structure to guide feature extraction and sampling (lower row).

[1] introduces a Bayesian optimization framework with a Gaussian process surrogate to efficiently tune EDA tool parameters, while PTPT [6] extends it to a multi-objective setting via multi-task Gaussian processes. REMOTune [7] combines random embedding with trust-region BO to support scalable, parallel exploration. Explorer [8] addresses hybrid parameter spaces using diffusion-based kernels in BO. RankTuner [9] adopts a preference-based formulation with a pairwise Gaussian process, learning pairwise rankings to circumvent direct QoR regression.

However, existing methods generally treat tool parameters as flat, independent vectors and model the EDA flow as a black-box function, neglecting structural dependencies. While some approaches attempt to mitigate dimensionality via random embeddings or pairwise influence estimation [5], [7], [8], they fail to capture the causal structure intrinsic to sequential EDA flows. In practice, early-stage parameters (e.g., netlist synthesis) impose directional influence on downstream stages (e.g., placement, routing), constraining feasible configurations and shaping QoR outcomes [10]. Ignoring such asymmetries reduces surrogate model expressiveness and limits optimization efficacy in complex, high-dimensional settings.

Recognizing this structural gap, we reformulate parameter tuning from a causal perspective. Bayesian optimization can be viewed as a sequential decision-making problem governed by these causal mechanisms [11], where each sampled configuration influences future search directions. This work seeks to address that gap by posing the following question: *Can causality help with high-dimensional EDA tool parameter tuning?* To leverage the causal structure, we propose *CausalTuner*, a novel framework that integrates structural causal reasoning into the BO loop for high-dimensional EDA tool parameter tuning, as shown in Fig. 1. At its core, CausalTuner proposes a hybrid causal attention mechanism to extract features capturing both intra-stage interactions and inter-stage causal ordering, feeding them into a deep kernel Gaussian process surrogate model. A trust-region strategy with boundary proximity penalty to stabilize parallel local search, and a front-end intervention filtering mechanism to prioritize upstream parameters with significant causal impact. Together, these components yield a unified and scalable framework that improves both the sample efficiency and interoperability. Our major contributions are summarized as follows:

- To the best of our knowledge, we are the first to uncover the latent causal structure in the EDA workflow, modeling hierarchical dependencies across different stages to guide parameter tuning more effectively.
- We design a causality-aware feature extraction method using a hybrid causal attention mechanism, which captures stage-wise parameter interactions and feeds them into a deep kernel Gaussian Process for accurate and expressive surrogate modeling.
- We propose a causality-guided Bayesian optimization framework, incorporating a trust-region method with boundary proximity penalties, and a front-end intervention filtering mechanism to improve sampling efficiency and stability.
- Experimental results show our framework substantially outperforms state-of-the-art EDA parameter tuning methods, achieving up to 45% improvement in hypervolume.

II. Preliminaries

A. Bayesian Optimization with Gaussian Process

Bayesian optimization (BO) provides a sample-efficient framework for global optimization of expensive black-box functions, making it well-suited for EDA tool parameter tuning where each design evaluation is costly. BO employs a probabilistic surrogate model to approximate the objective and iteratively selects query points via an acquisition strategy informed by the surrogate's predictive distribution. This balances exploration and exploitation to efficiently approach the global optimum.

Gaussian processes (GPs) are commonly used as surrogates in BO which provide closed-form predictions for both the mean and uncertainty of the objective value $\boldsymbol{y}$, $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\mu(\boldsymbol{x}), \sigma(\boldsymbol{x}))$which are essential for acquisition functions such as Expected Improvement (EI) or Hypervolume Improvement (HVI). However, in high-dimensional and structurally complex parameter spaces, standard GPs with simple kernels often fail to capture intricate dependencies, limiting their effectiveness in EDA contexts and motivating the development of enhanced kernels or structured feature extractors.

B. Causal Reasoning Model

To treat causality rigorously, we first need to formulate a mathematically well-defined causal reasoning model. Causal inference [12] provides a principled framework for identifying and quantifying cause-effect relationships among variables, distinguishing genuine causation from spurious correlation. Within this domain, causal reasoning focuses on explicitly modeling causal structures, typically through Structural Causal Models (SCMs).

An SCM formally captures the underlying causal mechanisms of a system [13]. It comprises a set of observed endogenous variables $\boldsymbol{X} = \{X_1, \dots, X_d\}$, unobserved exogenous variables $\boldsymbol{U} = \{U_1, \dots, U_d\}$, and structural equations $\boldsymbol{F} = \{f_1, \dots, f_d\}$ such that $X_i := f_i(Pa_i, U_i), \quad i \in \{1, \dots, d\}$, where each f_i maps from the domain of $U_i \cup Pa_i$ to the value of X_i. Here, $Pa_i \subseteq \boldsymbol{X} \setminus \{X_i\}$ denotes the set of direct causal parents of X_i, and $U_i \subseteq \boldsymbol{U}$ represents the associated exogenous variables. The exogenous variables represent unobserved influences and follow a joint probability distribution $P(\boldsymbol{U})$.

The causal relationships implied by an SCM are naturally represented by a Directed Acyclic Graph (DAG), where each node represents a variable in $\boldsymbol{X}$, and a directed edge $X_j \to X_i$ exists if and only if $X_j \in Pa_i$. The acyclic nature of the graph ensures a well-defined causal ordering among variables. This perspective will later help us describe the stage-wise causal structure of EDA parameters in Section III-B.

C. Problem Formulation

Definition 1 (Pareto Optimality)**.** *In a multi-objective minimization problem with M objectives, a solution $\boldsymbol{x}_1$ is said to dominate another solution $\boldsymbol{x}_2$ (denoted $\boldsymbol{x}_1 \succeq \boldsymbol{x}_2$) if $f_m(\boldsymbol{x}_1) \leq f_m(\boldsymbol{x}_2)$ for all $m \in \{1, \dots, M\}$, and there exists at least one objective k for which $f_k(\boldsymbol{x}_1) < f_k(\boldsymbol{x}_2)$.*

The set of all solutions not dominated by any other is called the Pareto-optimal set. These solutions define the Pareto front, representing the optimal trade-offs among competing objectives in the design space.

Problem 1 (High-dimensional Tool Parameter Tuning)**.** *Given a high-dimensional parameter search space $\mathcal{X}$, each tool parameter inside $\mathcal{X}$ is regarded as a feature vector $\boldsymbol{x}$. For each $\boldsymbol{x}$, the corresponding quality-of-results (QoR) metrics $\boldsymbol{y}$ can be obtained through the VLSI implementation flow. The objective is to automatically identify Pareto-optimal parameters that optimize multiple QoR metrics—including performance, power consumption, and area—while minimizing runtime.*

III. CausalTuner Framework

A. Overview

The workflow of our high-dimensional parameter tuning framework, CausalTuner, which incorporates knowledge of latent causal relationships inherent in the EDA flow, is illustrated in Fig. 2. To enable layout generation, the EDA flow requires preliminary setup, including the technology library, HDL source files, constraint specifications, and defined parameter ranges. To extract accurate and informative causality-aware feature representations, the Transformer-based encoder with hybrid causal attention is pretrained in a self-supervised manner. To guide efficient and causality-informed exploration of the design space, CausalTuner integrates intervention-guided uncertainty and boundary proximity penalties into a

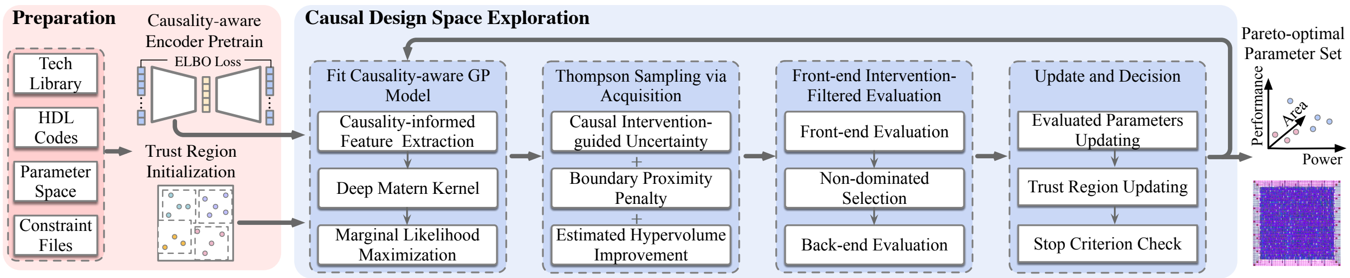


Fig. 2 The overall flow of our CausalTuner framework.

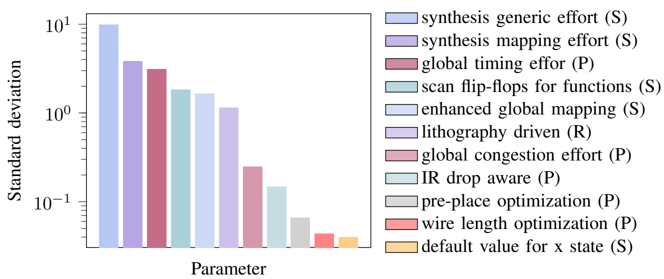


Fig. 3 Parameter importance on `RISCV32I`, measured by standard deviations via automatic relevance determination. S, P, and R denote synthesis, placement, and routing, respectively.

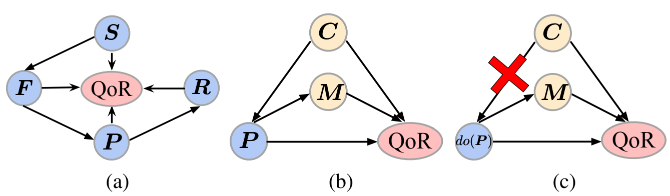


Fig. 4 (a) Causal graph of the VLSI design flow from synthesis ($\boldsymbol{S}$) to floorplanning ($\boldsymbol{F}$), placement ($\boldsymbol{P}$), and routing ($\boldsymbol{R}$), leading to final QoR. (b) Causal graph of placement parameters illustrating the roles of a confounder ($\boldsymbol{C}$) and a mediator ($\boldsymbol{M}$). (c) Intervention on placement parameters eliminates confounder influence.

trust region-based acquisition strategy and further applies an intervention-filtered evaluation mechanism to prioritize upstream parameters. The output consists of the explored parameter configurations from the iterative optimization process. from which the Pareto-optimal set is selected.

B. Causality-informed Feature Extractor

Prior work often models EDA tool parameters as flat, independent vectors, overlooking structural and stage-wise dependencies. To illustrate parameter heterogeneity, we apply automatic relevance determination (ARD) [14] on `RISCV32I`, as shown in Fig. 3. Standard deviations computed from 100 ARD models quantify each parameter's sensitivity; larger values indicate greater influence. This aligns with the inherent causal hierarchy of the EDA flow, where early-stage decisions (e.g., synthesis) constrain downstream stages (e.g., placement and routing). Ignoring such dependencies limits both optimization efficiency and interpretability.

Latent Causal Structure in Tool Parameters. To explicitly characterize the latent causal structure, we represent the EDA parameters using Structural Causal Models (SCMs). Consider a sequence of EDA tool parameters $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$, where parameters at earlier positions correspond to earlier stages of the EDA flow, and parameters at later positions correspond to subsequent stages. We partition the parameter sequence into K stage-specific blocks:

$$\boldsymbol{x} = [\hat{\boldsymbol{x}}^{(1)}, \ldots, \hat{\boldsymbol{x}}^{(K)}], \quad \hat{\boldsymbol{x}}^{(k)} = [x_1^{(k)}, \ldots, x_{n_k}^{(k)}], \tag{1}$$

where $\hat{\boldsymbol{x}}^{(k)}$ denotes parameters of the k-th stage, and n_k is the number of parameters at that stage. According to the structure analysis in Section II-B, the causal relationships among these parameters and the final QoR can be described through the following structural equations:

$$\begin{aligned} \hat{\boldsymbol{x}}^{(1)} &:= f_1(\boldsymbol{u}_1), \\ \hat{\boldsymbol{x}}^{(2)} &:= f_2(\hat{\boldsymbol{x}}^{(1)}, \boldsymbol{u}_2), \\ &\vdots \\ \hat{\boldsymbol{x}}^{(K)} &:= f_K(\hat{\boldsymbol{x}}^{(K-1)}, \boldsymbol{u}_K), \\ \boldsymbol{y} &:= f_Y(\hat{\boldsymbol{x}}^{(1)}, \ldots, \hat{\boldsymbol{x}}^{(K)}, \boldsymbol{u}_{\boldsymbol{y}}), \end{aligned} \tag{2}$$

where each f_i is a deterministic function representing the causal mechanism of stage parameters, $\boldsymbol{u}_i$ are exogenous latent factors encapsulating unobserved variations, and $\boldsymbol{y}$ collectively denotes QoR metrics.

The equations' causal structure forms a directed acyclic graph (DAG), as is shown in Fig. 4(a), where each node represents the set of parameters in one design stage and each direct link denotes a causal relationship between nodes. As an example, Fig. 4(b) illustrates the causal graph for placement parameters. The path $\boldsymbol{P} \rightarrow$ QoR captures the direct effect of placement parameters, whereas $\boldsymbol{P} \rightarrow \boldsymbol{M} \rightarrow$ QoR represents a beneficial mediated effect that improves robustness [15]. $\boldsymbol{M}$ denotes mediators invariant across tool-parameter distributions. By contrast, $\boldsymbol{P} \leftarrow \boldsymbol{C} \rightarrow$ QoR reflects unstable confounding introduced by earlier design choices and exogenous noise (e.g., tool randomness, unmodeled heuristics, discretization). In Fig. 4(c), an intervention $do(\boldsymbol{P})$ fixes $\boldsymbol{P}$ and removes its incoming edges, isolating its causal influence. This example illustrates how upstream decisions cascade through the EDA flow and ultimately shape chip-level QoR. To approximate the observational distribution $p(\boldsymbol{x})$ induced by this causal model, we aim to learn a mapping $\boldsymbol{x} \rightarrow \boldsymbol{z}$, treating $\boldsymbol{z}$

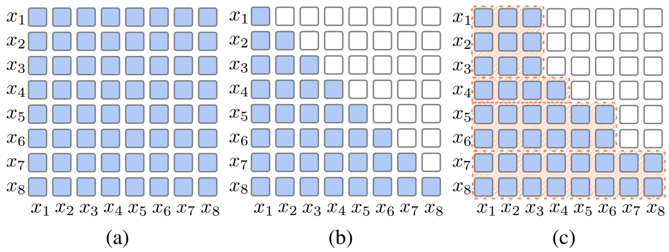


Fig. 5 Comparison of different attention patterns. A blue cell at row i, column j indicates that parameter x_i can attend to x_j while white cells indicate no attention. (a) Full attention matrix; (b) standard causal attention; (c) hybrid causal attention with a block-wise structure, grouping parameters from the same EDA stage within each orange block.

as latent causal representations.

Stage-based Hybrid Causal Attention. Rohekar *et al.* [16] demonstrated that the self-attention mechanism in Transformer architectures can estimate the SCMs defined in Equation (2). In EDA workflows, earlier stages causally influence later ones, but not the reverse, resulting in a directed, stage-wise structure that naturally aligns with a causal attention mask. Conventional Transformer-based causal attention [17] employs a lower-triangular mask, restricting each position to attend only to itself and preceding positions, as can be seen in Fig. 5(b). However, this rigid constraint is unsuitable for EDA parameter tuning, where parameters within the same stage should interact freely while preserving the hierarchical order across stages. Therefore, we propose a hybrid causal mask that permits unrestricted attention within each stage while blocking attention from later to earlier stages, as illustrated in Fig. 5(c).

Given the partitioned parameters $\boldsymbol{x} = [\hat{\boldsymbol{x}}^{(1)}, \ldots, \hat{\boldsymbol{x}}^{(K)}]$, the proposed hybrid causal attention is calculated as $\texttt{Att}_{HC}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \texttt{softmax}(\boldsymbol{M}_{ca} \odot (\boldsymbol{Q}\boldsymbol{K}^\top)/\sqrt{d})\boldsymbol{V}$, where $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are query, key, and value embeddings of parameter sequences, respectively. d is the embedding dimension. The causal mask $\boldsymbol{M}_{ca}$ is defined as:

$$\boldsymbol{M}_{ca} = \begin{bmatrix} \mathbf{1}_{n_1 \times n_1} & \mathbf{0}_{n_1 \times n_2} & \cdots & \mathbf{0}_{n_1 \times n_K} \\ \mathbf{1}_{n_2 \times n_1} & \mathbf{1}_{n_2 \times n_2} & \cdots & \mathbf{0}_{n_2 \times n_K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{1}_{n_K \times n_1} & \mathbf{1}_{n_K \times n_2} & \cdots & \mathbf{1}_{n_K \times n_K} \end{bmatrix}, \tag{3}$$

where $\mathbf{1}_{a \times b}$ and $\mathbf{0}_{a \times b}$ represent an $a \times b$ matrix of ones and a $a \times b$ matrix of zeros, respectively. $\odot$ is element-wise multiplication. In this framework, the K is set to 4 for synthesis, floorplan, placement, and routing stages, respectively. This hybrid causal mask also conceptually corresponds to a truncated factorization of the interventional distribution:

$$p(\hat{\boldsymbol{x}}^{(K)} | \hat{\boldsymbol{x}}^{(1)}) = \prod_{k=2}^{K} p(\hat{\boldsymbol{x}}^{(k)} | \hat{\boldsymbol{x}}^{(k-1)}). \tag{4}$$

The causal mask enforces stage-wise conditional independence, ensuring that each stage attends only to its predecessors. This explicit encoding of stage-wise causal dependencies improves feature extraction accuracy and interpretability, ultimately enhancing downstream parameter optimization and QoR prediction.

Causality-aware Embedding Generation. To enhance generalization and provide effective initialization for downstream Bayesian optimization in data-scarce regimes, we pretrain the hybrid causality-aware feature extractor via self-supervised variational inference. Specifically, we optimize a variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{u})$ to approximate the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{u})$, where $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are parameters of the feature extraction model and a simple MLP-based decoder, respectively. Training minimizes the negative evidence lower bound (ELBO) over tool parameters $\boldsymbol{x}$ $\mathcal{X}$:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\phi}; \boldsymbol{x}) =& \mathbb{E}_{z \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{u})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{u})] \\ &+ \mathcal{D}_{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{u}) || p(\boldsymbol{z}|\boldsymbol{u})), \end{aligned} \tag{5}$$

where $\mathcal{D}_{KL}(q(\cdot)||p(\cdot))$ is Kullback-Leibler divergence between two distributions $q(\cdot)$ and $p(\cdot)$. We adopt a standard Gaussian prior $p(\boldsymbol{z}|\boldsymbol{u}) = \mathcal{N}(0, \boldsymbol{I})$ in initialization phase.

C. Causal Design Space Exploration

Deep Kernel Gaussian Process Surrogate Model. We propose a causality-aware Bayesian optimization (BO) framework for efficient exploration in high-dimensional, multi-objective EDA settings. The surrogate model adopts a deep kernel Gaussian process (DKGP), integrating a pretrained causality-aware feature extractor $h_{\boldsymbol{\phi}}(\cdot)$ with a Matérn-5/2 kernel $\kappa_{5/2}(r)$:

$$\tilde{k}_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{x}') = \kappa_{5/2}\left(\|h_{\boldsymbol{\phi}}(\boldsymbol{x}) - h_{\boldsymbol{\phi}}(\boldsymbol{x}')\|_2\right), \tag{6}$$

yielding expressive embeddings and calibrated uncertainty.

Trust Region Strategy with Boundary Proximity Penalty. In high-dimensional bounded design spaces, BO tends to oversample near parameter boundaries, where surrogate uncertainty is artificially inflated due to the absence of extrapolative data. Such regions are often infeasible or unstable in VLSI design. To address this, we introduce a boundary proximity penalty: For each normalized parameter vector $\boldsymbol{x} \in [0, 1]^d$, the distance to the nearest boundary in each dimension is computed as $d_i = \min(x_i, 1 - x_i)$, and a Gaussian-decaying penalty is applied:

$$\text{BP}(\boldsymbol{x}) = \sum_{i=1}^{d} \exp\left(-\frac{d_i^2}{\sigma^2}\right), \tag{7}$$

where $\sigma = 0.1$ is decaying parameter. Subtracting BP from acquisition scores discourages boundary-adjacent candidates. This soft penalty improves robustness without hard constraints and complements trust region sampling strategy [7].

Sampling with Front-end Intervention Filtering. Due to the high cost of full-flow evaluations, we adopt an early-stage filtering strategy to improve sampling efficiency. Specifically, $2b$ candidates are drawn via Thompson sampling and scored using a composite acquisition function:

$$\text{Score}(\boldsymbol{x}_*) = HVI(\hat{\boldsymbol{y}}_* | \mathcal{P}(\boldsymbol{y}), \boldsymbol{y}_{ref}) - \alpha \cdot \text{BP}(\boldsymbol{x}_*). \tag{8}$$

TABLE I Examples of the flow parameters.

Stage (#Params)	Parameter Name	Range or Options

Synthesis (105)	`auto partition`
	`synthesis general effort`
	`synthesis map effort`
		false/true
		medium/low/high/express/none
		high/low/medium/express/none

Floorplan (7)	`aspect ratio` `core margin`	0.5-2.0 1.0-5.0

Placement (13)	`global timing effort`
	`global congestion effort`
	`pre-placement optimization`
	`wireclength optimization`
		medium/high
		low/medium/high
		false/true
		none/medium/high

Routing (9)	`timing driven` `via driven`	false/true false/true

In above equation α is weight coefficient and HVI is expressed as $HVI(\hat{\boldsymbol{y}}_*|\mathcal{P}(\boldsymbol{y}), \boldsymbol{y}_{ref}) = V_{\boldsymbol{y}_{ref}}(\mathcal{P}(\boldsymbol{y} \cup \hat{\boldsymbol{y}}_*)) - V_{\boldsymbol{y}_{ref}}(\mathcal{P}(\boldsymbol{y}))$, where $\hat{\boldsymbol{y}}_*$ is the GP-predicted QoR for sampled $\boldsymbol{x}_*$, and $\mathcal{P}(\boldsymbol{y})$ is the current Pareto set. The hypervolume (HV) corresponds to the M-dimensional *Lebesgue measure* λ_M of the region dominated by an approximate Pareto frontier $\mathcal{P}(\boldsymbol{y})$ and bounded from below by a reference point $\boldsymbol{y}_{ref} \in \mathbb{R}^M$. It can be expressed as:

$$HV_{\boldsymbol{y}_{ref}}(\mathcal{P}(\boldsymbol{y})) = \lambda_M(\cup_{\boldsymbol{y} \in \mathcal{P}(\boldsymbol{y})}[\boldsymbol{y}, \boldsymbol{y}_{ref}]). \tag{9}$$

All candidates undergo lightweight front-end evaluation, and the top b are selected based on objective-space diversity. These intermediate QoRs serve as soft interventions, filtering out low-quality configurations and allowing the optimizer to allocate full-flow resources to promising candidates.

IV. Experimental Results

A. Experimental Setup and Benchmarks

We conduct experiments using a VLSI design flow based on Cadence Genus and Innovus (version 17.1), optimizing 134 parameters across four EDA stages—synthesis, floorplan, placement, and routing—which is consistent with recent state-of-the-art works [7], [9]. Some examples of tunable parameters are shown in TABLE I. Different parameters control various stages' functional options and impact subsequent stages. For example: `auto partition` enables/disables automatic partitioning of the design into sub-blocks for hierarchical synthesis, `global timing effort` controls timing-driven placement aggressiveness, `via driven` optimizes via numbers reduction.

We evaluate CausalTuner on a suite of RISC-V processor designs synthesized using TSMC 65nm technology. The primary benchmarks include `RISCV32I` [18] and `Rocket` [19], comprising approximately $7.6k$ and $14.2k$ standard cells, respectively. `RISCV32I` is hand-written in Verilog, whereas `Rocket` is generated using Chisel, leading to differences in design structure and the optimal parameters for synthesis.

In the setting of CausalTuner, the causality-aware feature extractor generates a 32-dimensional embedding, followed by 2 transformer layers with 2 attention heads. To avoid the overhead of full VLSI implementation, the model is pretrained via self-supervised learning on 10,000 valid design points sampled from the parameter space. Pretraining is performed once on a GeForce RTX 3090 for 100 epochs (Ĩ0 minutes) and reused across all benchmarks with the same design space.

The baseline methods compared are considered state-of-the-art (SOTA) works in EDA tool parameter tuning as in [9]: 1) **FIST** [5]: Uses XGBoost ensemble trees and importance sampling to adjust EDA design parameters. 2) **DAC'19** [4]: Employs tensor decomposition and regression to build a collaborative prediction model, reducing parameter-tuning effort. 3) **MLCAD'19** [1]: Applies Bayesian optimization to explore EDA tool parameter space. 4) **ICCAD'21** [2]: An open-source platform integrating optimization algorithms like evolutionary algorithms and tree-structured Parzen estimators. 5) **TCAD'22** [6]: Uses multi-objective Bayesian optimization to find Pareto-optimal design parameters and incorporates multi-task Gaussian processes to model objective correlations. 6) **TODEAS'23** [7]: A leading method for guided parameter tuning using random embedding and multi-objective trust-region Bayesian optimization. 7) **DATE'24** [8]: An attention-based EDA tool parameter explorer that uses hybrid Gaussian processes to model interactions between continuous and discrete parameters. 8) **ICCAD'24** [9]: A ranking-based parameter tuning framework that achieves state-of-the-art results in EDA tool parameter tuning task.

To assess the quality of searched parameter configurations, we adopt a set of metrics focused on improving Quality of Results (QoR) as in [7], [9]. The primary metric is hypervolume (HV), measured with respect to a reference point $[150.0, 150.0, 150.0]$. The definition is given in Equation (9). We further report pairwise hypervolumes for comprehensive evaluation: $\text{HV}_{0,1}$ (performace-power), $\text{HV}_{0,2}$ (performance-area), and $\text{HV}_{1,2}$ (power-area). In addition, we use several maximum improvement metrics: MPI1 (performance), MPI2 (power), MAI (area), MPPI (product of clock period and power), and MPAI (product of clock period and area). All methods are evaluated under the same environment and implementation settings as in [7], [9] to ensure fair comparison.

B. Comparisons with SOTA Methods

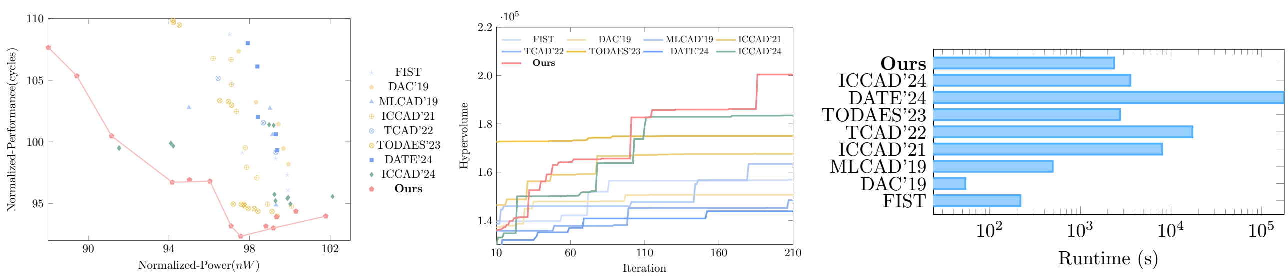
TABLE II and TABLE III present detailed comparisons on `RISCV32I` and `Rocket` benchmarks, respectively. CausalTuner consistently achieves the best tuning results. Compared to the leading baseline ICCAD'24 [9], CausalTuner improves hypervolume by 11.4% on `RISCV32I`, with 9.9%, 2.9% and 16.7% gains in pairwise hypervolumes $\text{HV}_{0,1}$, $\text{HV}_{0,2}$ and $\text{HV}_{1,2}$, respectively. Fig. 6(a) visualizes the Pareto-optimal sets selected by baselines and our method. CausalTuner's Pareto frontier dominates in the normalized performance and power metrics space, as shown in the red line. Our method also achieves the best maximum improvements across most objectives, reflecting the benefit of its sophisticatedly designed exploration strategies. On `Rocket` benchmark, CausalTuner also achieves the best Pareto sets with superior parameter qualities. It exceeds ICCAD'24 [9] by 3.0% hypervolume improvement, with pairwise improvements of 1.7%, 8.2%, and 2.2%. While TODAES'23 [7] reaches higher maximum improvements in the product of clock period and

TABLE II Comparison of parameter tuning methods on `RISCV32I` Benchmark.

Metric \ Method	FIST [5]	DAC'19 [4]	MLCAD'19 [1]	ICCAD'21 [2]	TCAD'22 [6]	TODAES'23 [7]	DATE'24 [8]	ICCAD'24 [9]	Ours
HV (10^5)	1.57	1.55	1.63	1.68	1.48	1.75	1.44	1.84	**2.05**
$\text{HV}_{0,1}$ (10^3)	2.85	2.72	3.00	2.95	2.70	3.05	2.63	3.44	**3.78**
$\text{HV}_{0,2}$ (10^3)	2.94	2.99	3.00	3.07	2.95	3.12	2.84	3.43	**3.53**
$\text{HV}_{1,2}$ (10^3)	2.97	2.97	3.00	3.14	2.79	3.23	2.77	3.00	**3.50**
MPI1 (%)	3.16	2.54	5.00	3.81	3.56	4.38	2.08	13.64	**13.65**
MPI2 (%)	3.90	2.12	5.12	5.23	0.85	6.27	0.68	5.04	**12.20**
MAI (%)	5.47	7.18	4.64	7.10	5.15	7.45	4.74	5.12	**7.47**
MPPI (%)	6.94	4.51	9.88	8.83	4.37	10.38	1.30	13.73	**13.91**
MPAI (%)	8.46	9.53	9.41	10.63	8.52	11.53	5.43	**12.26**	10.36

TABLE III Comparison of parameter tuning methods on `Rocket` benchmark

Metric \ Method	FIST [5]	DAC'19 [4]	MLCAD'19 [1]	ICCAD'21 [2]	TCAD'22 [6]	TODAES'23 [7]	DATE'24 [8]	ICCAD'24 [9]	Ours
HV (10^5)	1.47	1.19	1.35	1.50	1.31	1.61	1.36	1.67	**1.72**
$\text{HV}_{0,1}$ (10^3)	3.03	2.79	2.93	3.16	2.85	3.35	2.97	3.45	**3.51**
$\text{HV}_{0,2}$ (10^3)	3.02	2.75	2.94	3.16	2.84	3.18	2.63	3.06	**3.31**
$\text{HV}_{1,2}$ (10^3)	2.42	1.85	2.19	2.23	2.20	2.51	2.40	2.69	**2.75**
MPI1 (%)	12.38	14.50	13.44	**16.72**	11.97	16.11	7.34	13.00	13.91
MPI2 (%)	-0.51	-6.70	-2.99	-2.42	-2.83	1.57	2.16	5.09	**5.34**
MAI (%)	-1.01	-7.25	-3.32	-2.55	-3.39	-1.31	-3.79	-0.96	**-0.28**
MPPI (%)	11.93	8.77	10.85	14.70	9.48	**17.43**	5.46	13.11	15.26
MPAI (%)	11.50	8.30	10.67	14.60	8.99	**15.01**	1.02	6.68	8.43



(a) (b) (c)

Fig. 6 Visualization of experimental results. (a) Pareto frontier and comparison of selected Pareto-optimal sets in normalized performance-power metrics space on `RISCV32I` benchmark. (b) Comparison of hypervolume progression over iterations on `RISCV32I` benchmark. (c) Comparison of hypervolume on `BlackParrot` benchmark.

power or area, its hypervolume is 6.8% lower than ours, with 4.8%, 4.1% and 9.6% worse pairwise hypervolumes, revealing its limited robustness. In contrast, our method maintains consistently superior overall performance across objectives, indicating better stability and reduced sensitivity to outliers.

We compare convergence across methods via hypervolume progression in Fig. 6(b). CausalTuner begins with relatively weak initial configurations but rapidly outpaces all baselines as the search progresses. This demonstrates its ability to accumulate and exploit causal knowledge over time, enabling more informed exploration. Notably, CausalTuner surpasses competing methods around iteration 100 and continues to improve, ultimately achieving the highest hypervolume by a significant margin. This reflects its strong sample efficiency and effective navigation of high-dimensional design spac.

The runtime comparison of optimization procedures is shown in Fig. 6(c). DAC'19 [4] relies on a simple sampling strategy with a neural network, thus achieving the shortest runtime. FIST [5] and MLCAD'19 [1] also exhibit low runtime overhead, primarily due to the use of simple surrogate models and sampling acquisition. Compared to other high-performing parameter tuning baselines, our method achieves the fastest overall runtime. CausalTuner is 16.3% faster than TODAES'23 [7], which also employs a trust-region strategy, and 51.9% faster than ICCAD'24 [9], whose pairwise Gaussian process incurs significant computational overhead.

V. Conclusion

In this paper, we introduce CausalTuner, the first framework to exploit latent causal structure for high-dimensional EDA parameter tuning. It combines causality-aware feature extraction with a hybrid stage-wise causal attention mechanism to model both inter-stage ordering and intra-stage dependencies. To improve sampling efficiency, we introduce a boundary proximity penalty and intervention-based candidate filtering. Empirical results show that CausalTuner outperforms state-of-the-art methods in both search quality and runtime, achieving up to a 45% gain in hypervolume.

References

[1] Y. Ma, Z. Yu, and B. Yu, "CAD tool design space exploration via bayesian optimization," in *Proc. MLCAD*, 2019.

[2] J. Jung, A. B. Kahng, S. Kim, and R. Varadarajan, "METRICS2.1 and flow tuning in the IEEE CEDA robust design flow and OpenROAD," in *Proc. ICCAD*, 2021.

[3] M. M. Ziegler, H.-Y. Liu, G. Gristede, B. Owens, R. Nigaglioni, and L. P. Carloni, "A synthesis-parameter tuning system for autonomous design-space exploration," in *Proc. DATE*, 2016, pp. 1148–1151.

[4] J. Kwon, M. M. Ziegler, and L. P. Carloni, "A learning-based recommender system for autotuning design flows of industrial high-performance processors," in *Proc. DAC*, 2019.

[5] Z. Xie, G.-Q. Fang, Y.-H. Huang, H. Ren, Y. Zhang, B. Khailany, S.-Y. Fang, J. Hu, Y. Chen, and E. C. Barboza, "FIST: A feature-importance sampling and tree-based method for automatic design flow parameter tuning," in *Proc. ASPDAC*, 2020.

[6] H. Geng, T. Chen, Y. Ma, B. Zhu, and B. Yu, "Ptpt: physical design tool parameter tuning via multi-objective bayesian optimization," *IEEE TCAD*, vol. 42, no. 1, pp. 178–189, 2022.

[7] S. Zheng, H. Geng, C. Bai, B. Yu, and M. D. Wong, "Boosting vlsi design flow parameter tuning with random embedding and multi-objective trust-region bayesian optimization," vol. 28, no. 5, pp. 1–23, 2023.

[8] L. Donger, S. Qi, X. Qi, C. Tinghuan, and G. Hao, "Attention-based eda tool parameter explorer: From hybrid parameters to multi-qor metrics," 2024.

[9] P. Xu, S. Zheng, Y. Ye, C. Bai, S. Xu, H. Geng, T.-Y. Ho, and B. Yu, "Ranktuner: When design tool parameter tuning meets preference bayesian optimization," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD). IEEE*, 2024, pp. 1–7.

[10] T. Ajayi and D. Blaauw, "Openroad: Toward a self-driving, open-source digital layout implementation tool chain," in *Proceedings of Government Microcircuit Applications and Critical Technology Conference*, 2019.

[11] V. Aglietti, T. Damoulas, M. Álvarez, and J. González, "Multi-task causal learning with gaussian processes," *Advances in neural information processing systems*, vol. 33, pp. 6293–6304, 2020.

[12] J. Pearl, *Causality*. Cambridge university press, 2009.

[13] ——, *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009.

[14] D. Wipf and S. Nagarajan, "A new view of automatic relevance determination," *Advances in neural information processing systems*, vol. 20, 2007.

[15] X. Yang, H. Zhang, and J. Cai, "Deconfounded image captioning: A causal retrospect," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 12 996–13 010, 2021.

[16] R. Y. Rohekar, Y. Gurwicz, and S. Nisimov, "Causal interpretation of self-attention in pre-trained transformers," *Advances in Neural Information Processing Systems*, vol. 36, pp. 31 450–31 465, 2023.

[17] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[18] J. E. Stine, R. Ridley, and T.-D. Ene. (2021) Osu datapath/control rv32 single-cycle and pipelined architecture in sv. [Online]. Available: https://github.com/stineje/osu-riscv

[19] M. W. et al. (2017) Ibex risc-v core. [Online]. Available: https://github.com/lowRISC/ibex