ASP-DAC 2026
31st Asia and South Pacific Design Automation Conference

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

Jan. 21, 2026

Outline

Introduction

Motivation: Parameter Tuning is Everywhere

Parameter tuning is a daily bottleneck in implementation flows

- Hundreds of knobs across Synthesis $\rightarrow$ Floorplan $\rightarrow$ Placement $\rightarrow$ Routing
- QoR is multi-objective: performance, power, area
- No analytic model; each evaluation requires running the flow
- Search space is enormous: up to 10^{70} combinations

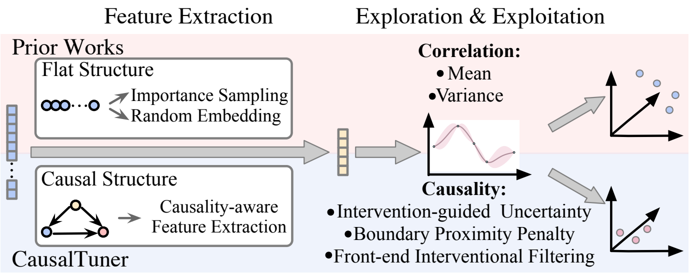
We need automation that is sample-efficient and robust.

What existing methods miss

Flat vector modeling ignores flow direction

- Many methods treat parameters as an unordered vector
- They mainly learn **correlations** from observed runs
- But EDA flows are **sequential pipelines**, not symmetric functions

The parameter space has a latent stage-wise causal structure.



Upper row: Prior works consider parameters as a flat structure and incorporate correlation only.
Lower row: CausalTuner leverages causal structure to guide feature extraction and sampling.

Problem formulation

High-dimensional tool parameter tuning:

Definitions:

- Parameters: $\boldsymbol{x} \in \mathcal{X}$ (high-dimensional; mixed discrete/continuous)
- QoR: $\boldsymbol{y} = f(\boldsymbol{x})$, expensive to evaluate via full EDA flow
- Goal: find Pareto-optimal configurations under limited budget

Pareto dominance: $\boldsymbol{x}_1 \succeq \boldsymbol{x}_2$ if $\forall m \in \{1, \ldots, M\}, f_m(\boldsymbol{x}_1) \leq f_m(\boldsymbol{x}_2)$, and $\exists k \in \{1, \ldots, M\}, f_k(\boldsymbol{x}_1) < f_k(\boldsymbol{x}_2)$

Core insight: EDA flows have built-in causal direction

Upstream decisions constrain downstream outcomes

- Synthesis choices shape netlist structure $\rightarrow$ affects placement/routing difficulty
- Downstream stages rarely cause upstream outcomes (no reverse influence)

In implementation flows, influence is **directional**.

CausalTuner Algorithm

Structural Causal Model (SCM) across Stages

Partition parameters into stage blocks:

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

SCM equations:

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

Upstream stages causally influence downstream stage parameter effects and final QoR.

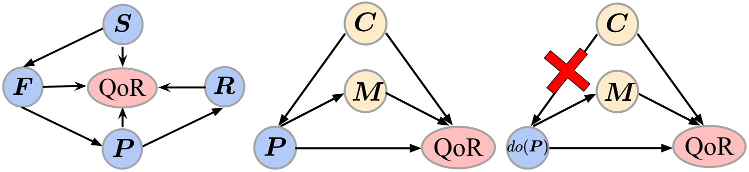
Why Interventions Matter in EDA Parameter Tuning

- Direct effect: $\boldsymbol{P} \rightarrow$ QoR
- Mediated effect: $\boldsymbol{P} \rightarrow \boldsymbol{M} \rightarrow$ QoR (robust, invariant mediator)
- Confounding: $\boldsymbol{P} \leftarrow \boldsymbol{C} \rightarrow$ QoR (unstable, exogenous confounder)

Interventions eliminate confounding influence:

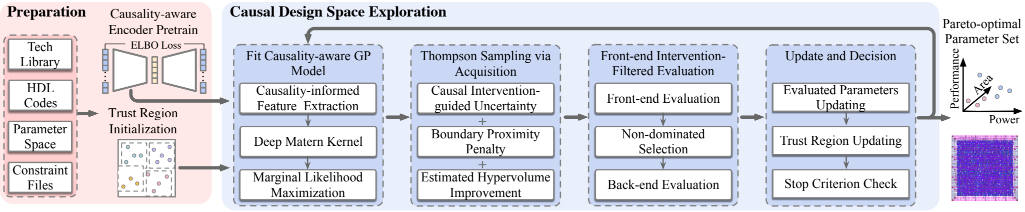
- Under $do(\boldsymbol{P} = \boldsymbol{p})$, incoming edges to $\boldsymbol{P}$ are removed $\rightarrow$ confounding is isolated

This motivates intervention-guided selection and filtering.



(a) (b) (c)

(a) Causal graph of the VLSI design flow from synthesis ($\boldsymbol{S}$) to final *QoR*.
(b) Causal graph of placement parameters illustrating the roles of confounder ($\boldsymbol{C}$) and mediator ($\boldsymbol{M}$).
(c) Intervention on placement parameters eliminates confounder influence.



The overall flow of our CausalTuner framework.

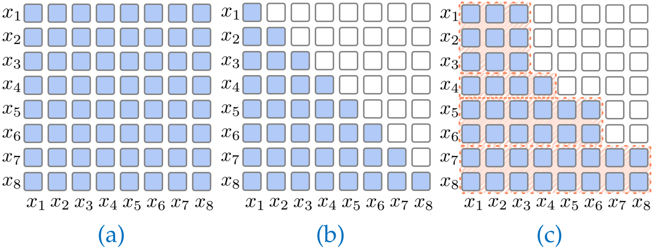
1. **Causality-aware feature extractor:** hybrid causal attention
2. **Deep kernel GP:** surrogate on learned embeddings
3. **Causal exploration:** trust-region BO, boundary proximity penalty, front-end intervention-filtered evaluation

Challenge:

- Full attention: too unconstrained; ignores stage ordering
- Standard causal mask (triangular): too strict; blocks useful within-stage interactions

Solution: Hybrid causal attention

- Free attention within a stage
- Causal ordering across stages (later stages cannot attend to previous stages; preserves hierarchy)



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.

Hybrid Causal Attention

Attention formula:

$$\texttt{Att}_{HC}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \texttt{softmax}(\boldsymbol{M}_{ca} \odot (\boldsymbol{Q}\boldsymbol{K}^\top)/\sqrt{d})\boldsymbol{V} \tag{3}$$

Mask matrix:

$$\boldsymbol{M}_{ca} = \begin{bmatrix} \mathbf{1}_{n_1 \times n_1} & \mathbf{0}_{n_1 \times n_2} & \cdots & \mathbf{0}_{n_1 \times n_K} \\ \mathbf{1}_{n_2 \times n_1} & \mathbf{1}_{n_2 \times n_2} & \cdots & \mathbf{0}_{n_2 \times n_K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{1}_{n_K \times n_1} & \mathbf{1}_{n_K \times n_2} & \cdots & \mathbf{1}_{n_K \times n_K} \end{bmatrix}, \tag{4}$$

stage k cannot attend to stages $\leq k$, and fully within its own stage.

Goal: learn the causal representation: $\boldsymbol{x} \rightarrow \boldsymbol{z}$
Learning Objective:

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

- Standard Gaussian prior $p(\boldsymbol{z}|\boldsymbol{u}) = \mathcal{N}(0, \boldsymbol{I})$ at initialization
- Pretraining done once on sampled valid design points; reused across benchmarks in same space

Deep Kernel Gaussian Process (DKGP)

Deep kernel definition:

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

- $h_{\phi}(\cdot)$: pretrained hybrid-causal encoder
- $\kappa_{5/2}(r)$: Matérn-5/2 kernel
- Benefit: improved expressiveness + calibrated uncertainty in high-dim space

Acquisition Objective

We score candidates by predicted hypervolume improvement:
Hypervolume (HV) definition:

$$HV_{\boldsymbol{y}_{ref}}(\mathcal{P}(\boldsymbol{y})) = \lambda_M(\cup_{\boldsymbol{y} \in \mathcal{P}(\boldsymbol{y})}[\boldsymbol{y}, \boldsymbol{y}_{ref}]). \tag{7}$$

Hypervolume Improvement (HVI) definition:

$$HVI(\hat{\boldsymbol{y}}_* | \mathcal{P}(\boldsymbol{y}), \boldsymbol{y}_{ref}) = HV_{\boldsymbol{y}_{ref}}(\mathcal{P}(\boldsymbol{y} \cup \hat{\boldsymbol{y}}_*)) - HV_{\boldsymbol{y}_{ref}}(\mathcal{P}(\boldsymbol{y})) \tag{8}$$

- $\hat{\boldsymbol{y}}_*$: GP predicted QoR for sampled $\boldsymbol{x}_*$
- $\mathcal{P}(\boldsymbol{y})$: current Pareto set
- $\boldsymbol{y}_{ref}$: reference point used for HV

Boundary Proximity Penalty

BO may oversample near bounds where uncertainty is high but solutions are unstable/infeasible.
For normalized $\boldsymbol{x} \in [0,1]^d$, define $d_i = \min(x_i, 1 - x_i)$
Boundary proximity penalty:

$$\mathrm{BP}(\boldsymbol{x}) = \sum_{i=1}^{d} \exp\left(-\frac{d_i^2}{\sigma^2}\right), \tag{9}$$

Composite score function:

$$\mathrm{Score}(\boldsymbol{x}_*) = HVI(\hat{\boldsymbol{y}}_* | \mathcal{P}(\boldsymbol{y}), \boldsymbol{y}_{ref}) - \alpha \cdot \mathrm{BP}(\boldsymbol{x}_*). \tag{10}$$

Soft discouragement improves stability without hard constraints.

Front-end Intervention-filtered Evaluation

Key Insight: Treat front-end results as soft interventions to filter poor configurations early.
Process:

1. Sample 2b candidates via Thompson sampling
2. Do lightweight front-end evaluation for all 2b samples
3. Select top b samples based on score and objective-space diversity
4. Only top b proceed to expensive back-end / full-flow evaluation

Experimental Results

Experimental Setup

Flows and tools:

- VLSI design flow based on Cadence Genus and Innovus (version 17.1)
- 134 parameters across four EDA stages—synthesis, floorplan, placement, and routing
- `RISCV32I` and `Rocket` benchmarks

Table: Examples of the flow parameters.

Stage (#Params)	Parameter Name	Range or Options
Synthesis (105)	`auto partition`	false/true
	`synthesis general effort`	medium/low/high/express/none
	`synthesis map effort`	high/low/medium/express/none
Floorplan (7)	`aspect ratio`	0.5-2.0
	`core margin`	1.0-5.0
Placement (13)	`global timing effort`	medium/high
	`global congestion effort`	low/medium/high
	`pre-placement optimization`	false/true
	`wireclength optimization`	none/medium/high
Routing (9)	`timing driven`	false/true
	`via driven`	false/true

Experimental Results

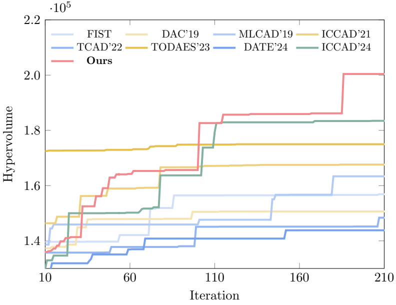
TABLE II Comparison of parameter tuning methods on `RISCV32I` Benchmark.

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**
$HV_{0,1}$ (10^3)	2.85	2.72	3.00	2.95	2.70	3.05	2.63	3.44	**3.78**
$HV_{0,2}$ (10^3)	2.94	2.99	3.00	3.07	2.95	3.12	2.84	3.43	**3.53**
$HV_{1,2}$ (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

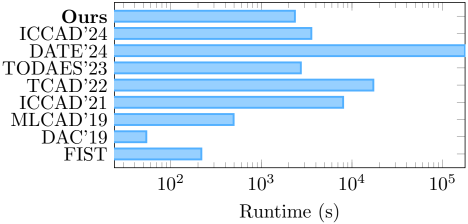
TABLE III Comparison of parameter tuning methods on `Rocket` benchmark

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**
$HV_{0,1}$ (10^3)	3.03	2.79	2.93	3.16	2.85	3.35	2.97	3.45	**3.51**
$HV_{0,2}$ (10^3)	3.02	2.75	2.94	3.16	2.84	3.18	2.63	3.06	**3.31**
$HV_{1,2}$ (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

Convergence and Runtime



(a)



(b)

(a) Comparison of hypervolume progression over iterations on `RISCV32I` benchmark. (b) The runtime comparison on `RISCV32I` benchmark.

- Starts weaker but overtakes baselines around iteration 100
- Continues improving towards best final hypervolume

Conclusion

- CausalTuner injects stage-wise causal structure into representation and BO Process
- Hybrid causal attention + DKGP improves modeling and uncertainty
- Penalty and intervention-filtered evaluation improves stability and efficiency

Causality is not just interpretability—it measurably improves high-dimensional EDA parameter tuning.

THANK YOU!