OpenPARF 3.0: Robust Multi-Electrostatics Based FPGA Macro Placement Considering Cascaded Macros Groups and Fence Regions

Jing Mai, Jiarui Wang, Yifan Chen, Zizheng Guo, Xun Jiang, Yun Liang, Yibo Lin

Peking University

jingmai@pku.edu.cn

May 11, 2024

Outline

1. Introduction
2. The `OpenPARF 3.0` Framework
3. Experimental Results
4. Conclusion & Future Work

Introduction

Modern FPGA Macro Placement Challenges

The modern FPGAs come with advanced technologies along with more chanllenges.

Massive Design

- Millions of cells in a single FPGA design

Highly Heterogeneity

- Various cells types (macros & standard cells)
- Cascaded macro
- Cascaded macro group

Highly Discrete

- Clock region
- Fence region constraint

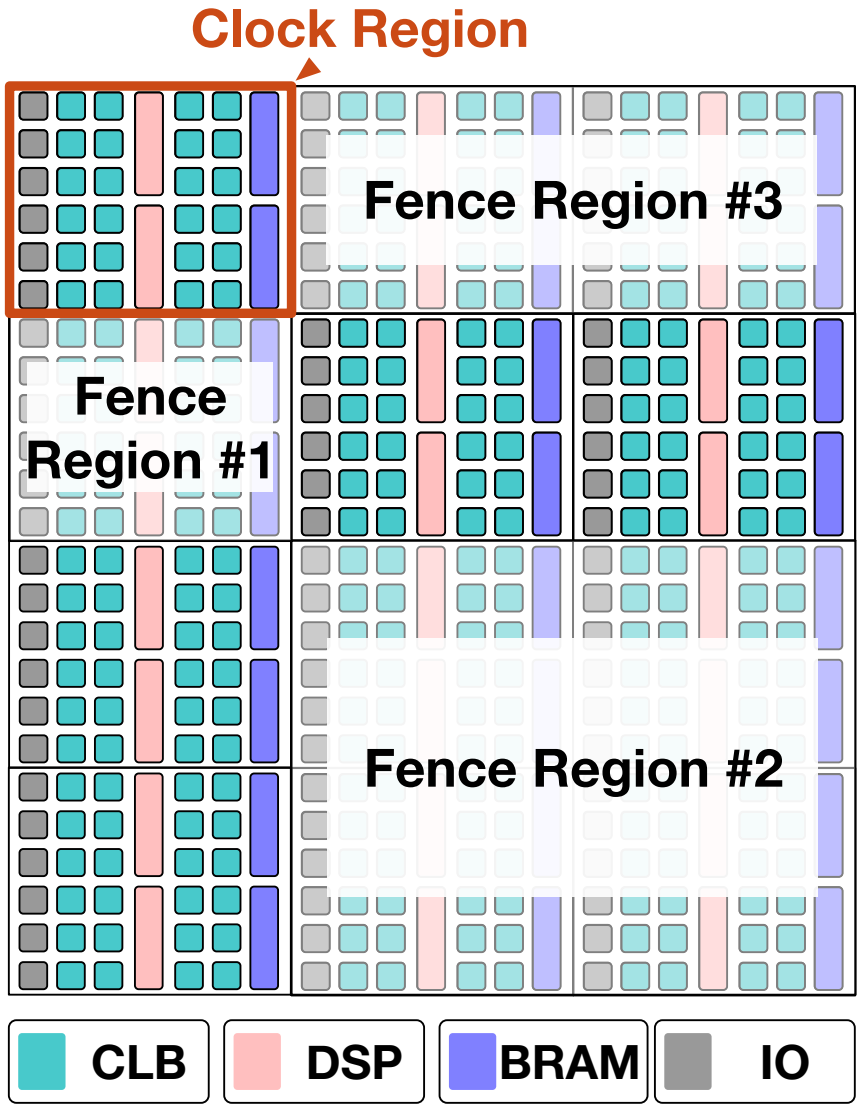
FPGA Architecture (I)

Various Instance Type

- Heterogenous resources & columnar distribution
- LUT & FF (standard cell, placed in CLB)
- DSP, BRAM (Macro)
- IO, etc.
- Clock Region

Fence Region Constraint

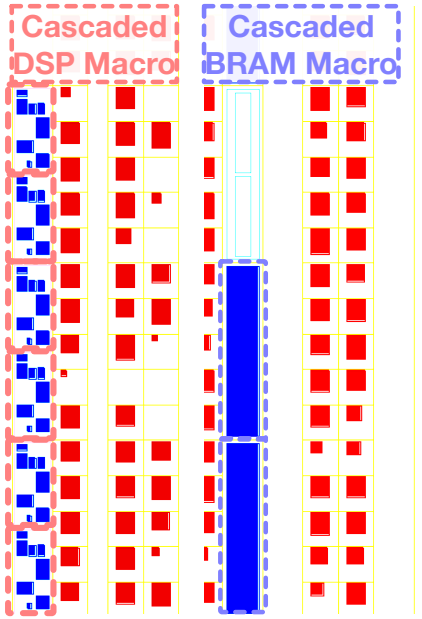
- Cells s.t. the constraint must be placed within the region
- Stem from clock regions or can be user-defined



FPGA Architecture (II)

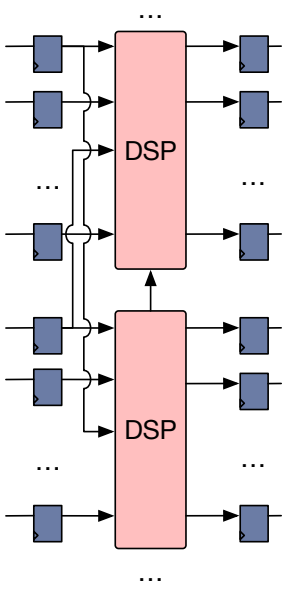
Cascaded Macro (CM)

- Cells must be placed in continuous columnated sites in a prescribed sequence
- Constraints on two macro types
- Cascaded DSP Macros
- Cascaded BRAM macros



Cascaded Macro Group

- *definition*: cascaded macros + the standard cells closely connected to them
- e.g., I/O signals of cascaded DSP macros are often connected to numerous FFs (~100)



FPGA macros significantly impact the overall routability of the design!

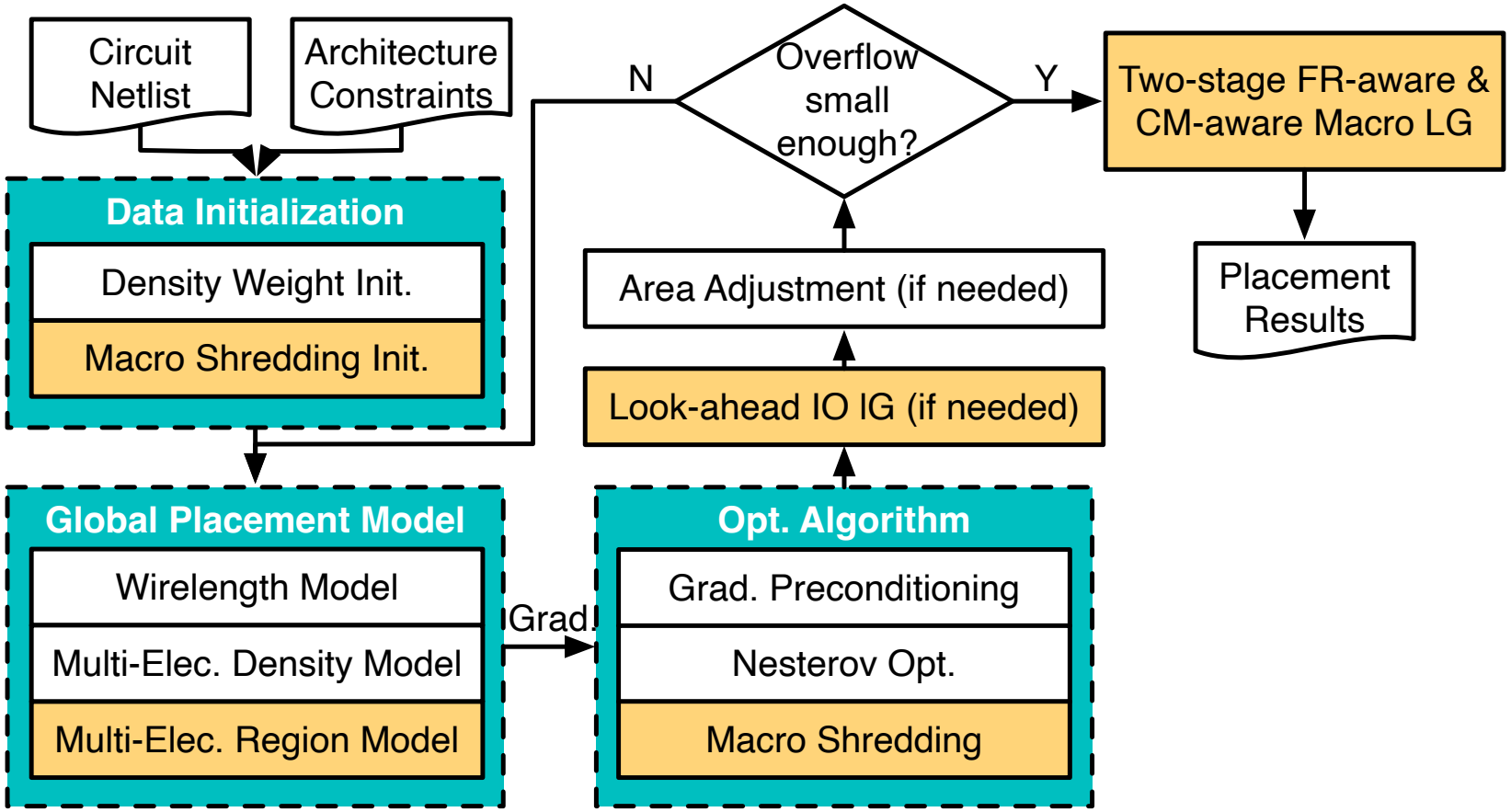
Contribution

We propose **`OpenPARF 3.0`**, a robust FPGA macro placer considering both fence region constraints and cascaded macro constraints.

- We propose a novel multi-electrostatics region model accompanied by a footprint compression technique to effectively handle the discontinuity in the fence region.
- We propose a divergence-aware density weight scheduling scheme which can effectively addresses robustness issues.
- We propose a cascaded macro shredding technique to address the imbalance issue of cascaded macro sizes.
- Experiments demonstrate that `OpenPARF 3.0` can achieve 13.3-49.1% overall score improvement as well as 1.81-3.18× speedup compared with Xilinx `Vivado 2021.1` and other SOTA academic FPGA macro placers.

The OpenPARF 3.0 Framework

Overall Flow of OpenPARF 3.0



Global Placement Model

Resource Tpye Set $\mathcal{T}$

$$\mathcal{T} = \{LUT, FF, DSP, BRAM, IO\}.$$

Multi-Electrostatics-based Global Placment Model

considering fence region constraints and cascaded macro constraints,

$$\min_{\boldsymbol{x},\boldsymbol{y}} L(\boldsymbol{x},\boldsymbol{y}) = \widetilde{W}(\boldsymbol{x},\boldsymbol{y}) + \sum_{s \in S} \lambda_s \mathcal{D}_s,$$
$$s.t.\ \mathcal{D}_s = \Phi_s + \frac{\mu}{2}\mathcal{P}_s\Phi_s^2, \forall s \in S^D \cup S^R,$$
Cascaded macro Constraints,

where S^D denotes the multi-electrostatics density model as `OpenPARF` [Mai+, ASICON'23],

$$S_D = \{S_{LUT}^D, S_{FF}^D, S_{DSP}^D, S_{BRAM}^D, S_{IO}^D\}.$$

Multi-Electrostatics-based Region Model

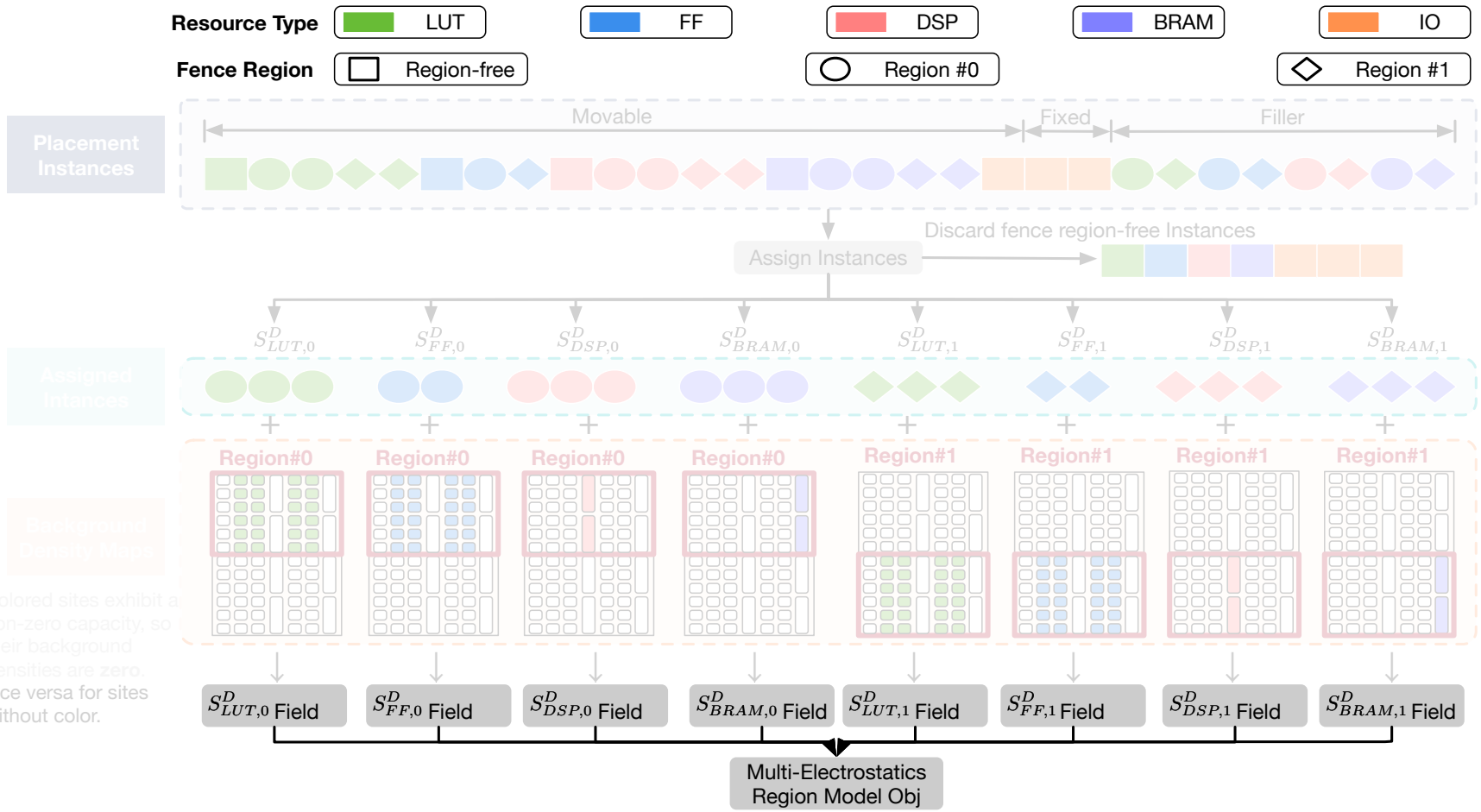
S_R denotes our proposed multi-electrostatics region model to resolve fence region constraints.

Multi-Electrostatics Region Model (I)

Multi-Electrostatics-based Region Model

For each resource type $t \in \mathcal{T}$ within each fence region $f \in \mathcal{F}$, we construct an electrostatics system $S^R_{t,f}$.

In other words, we construct $|S^R| = |\mathcal{T}||\mathcal{F}|$ more potential energy terms as the density objective.

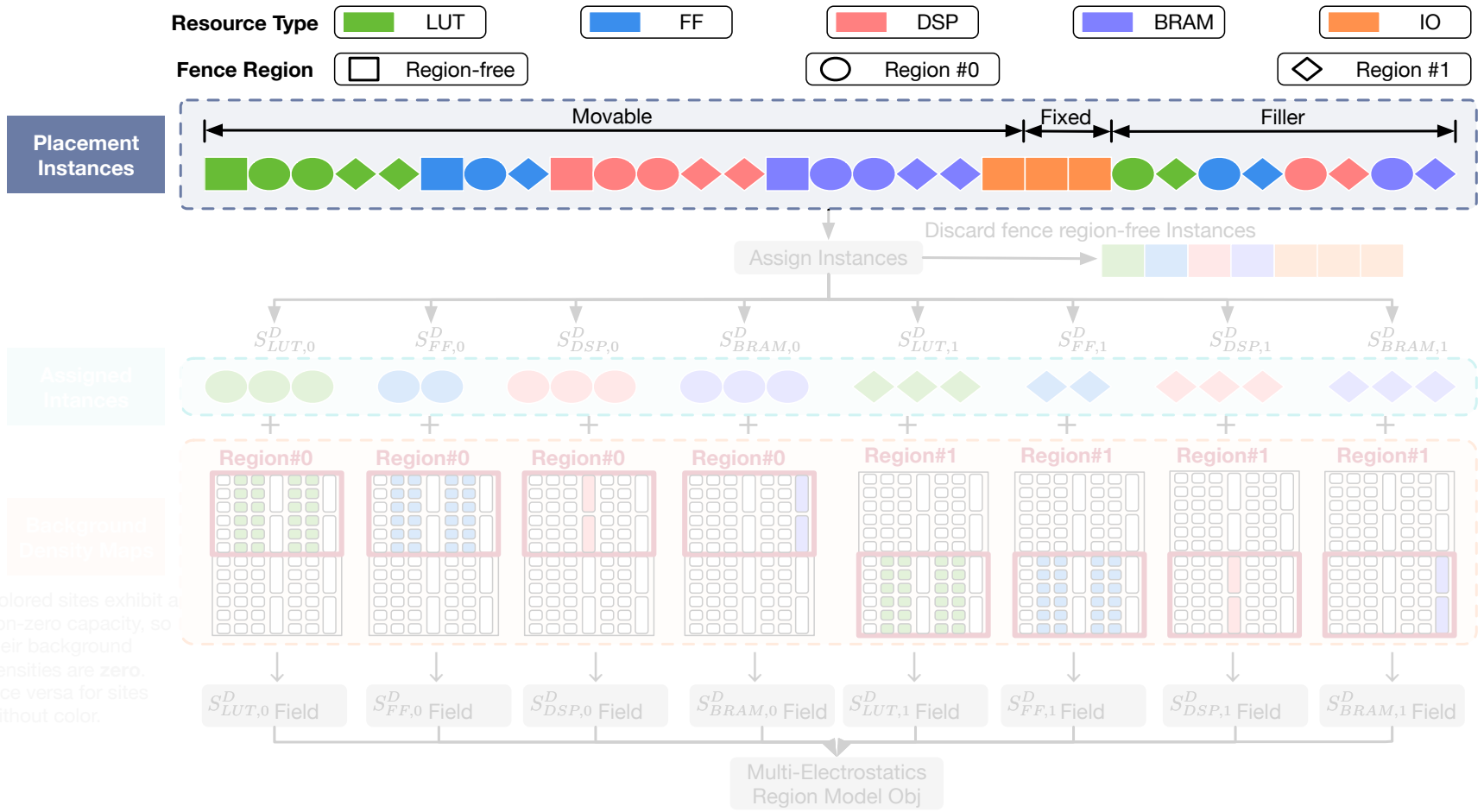


Multi-Electrostatics Region Model (II)

1. Placement Instances

All placement instances are firstly categorized as movable instances, fixed instances, and fillers.

Fillers are artificially created for the purpose of target density control for each electrostatics system.

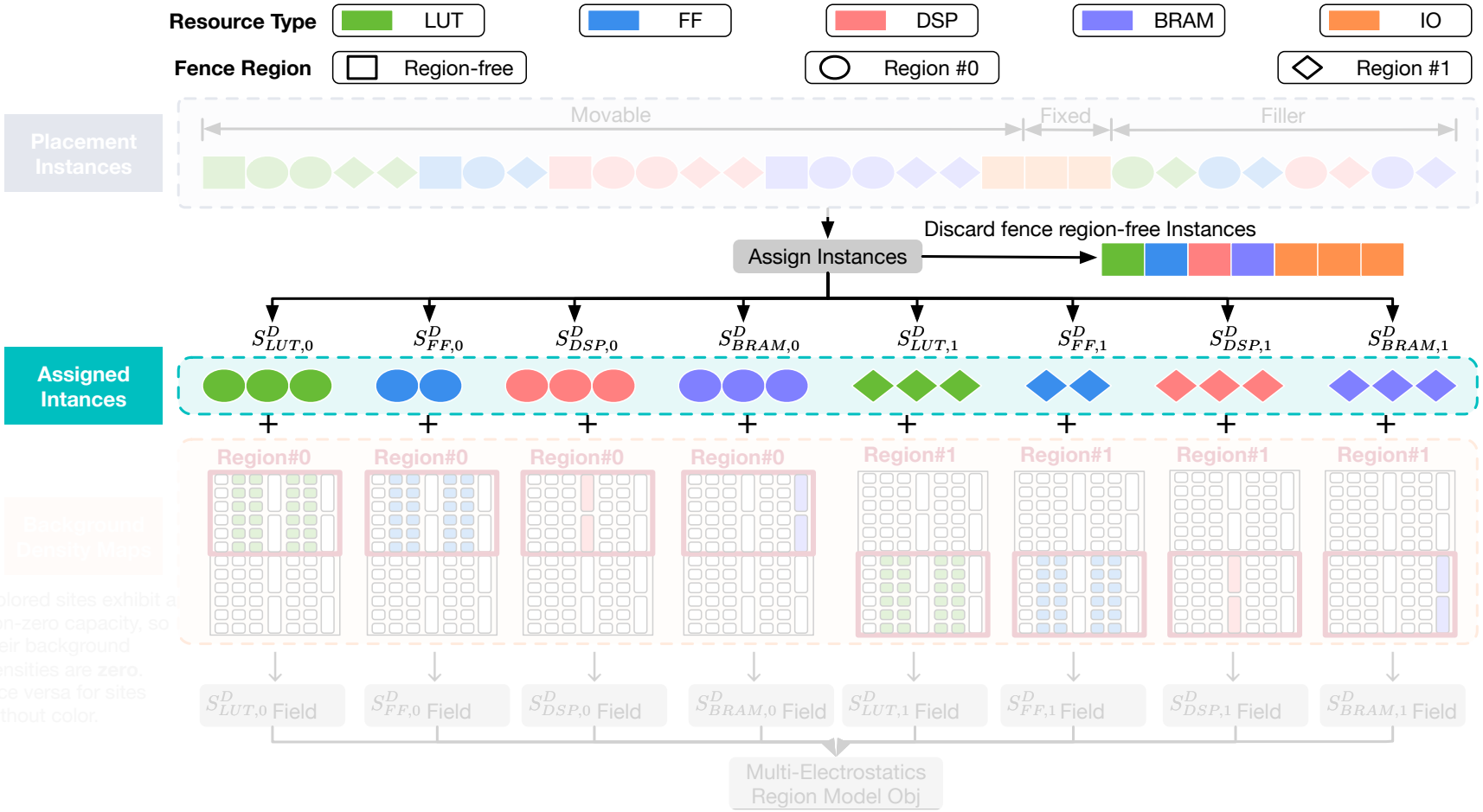


Multi-Electrostatics Region Model (III)

2. Assigned Instances

The placement instances are then assigned to electrostatics systems based on 1) their resource type and 2) the constraints imposed by the fence region they subject to.

Region-free Instances will be discarded.

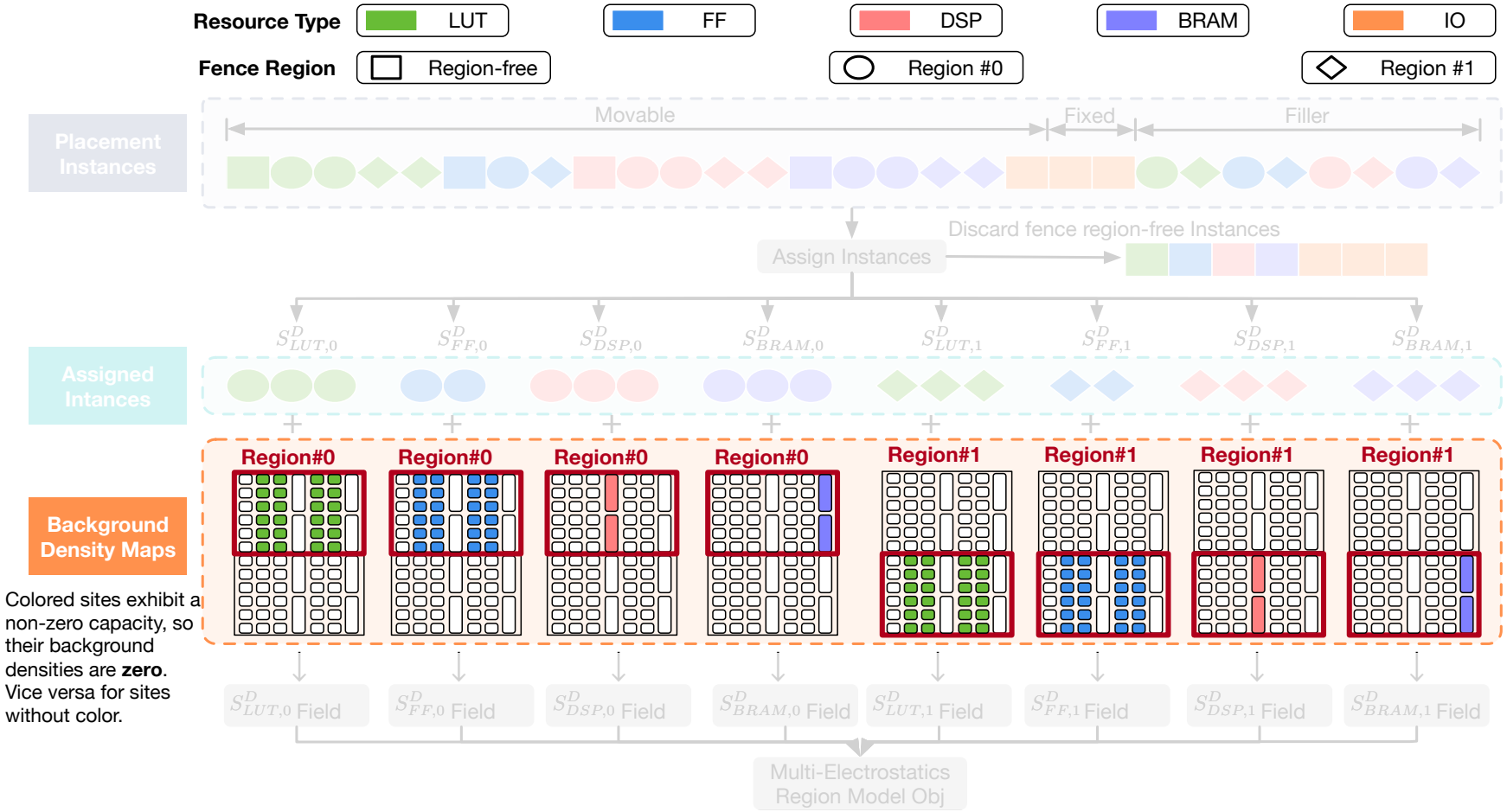


Multi-Electrostatics Region Model (IV)

3. Background Density Maps

For sites with capacity, the background density is set to zero, indicating the permissibility of placing instances on those sites.

For sites without capacity, the background density is set to the non-zero target density.



Footprint Compression Technique

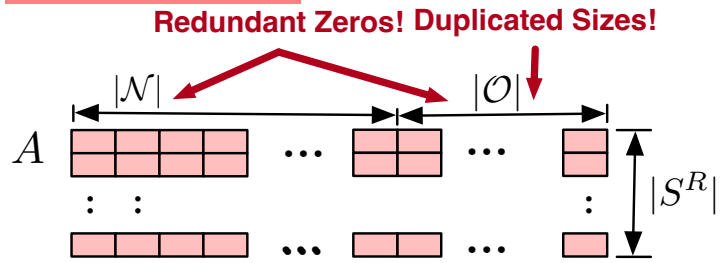
Memory allocation is required to store the sizes of placement instances under different electrostatics systems.

Space Complexity of previous approach: $\boldsymbol{O(\#nodes \times \#eSystems)}$

Three Observations

- Each movable and fixed instance subject to at most one fence region constraint.
- Each filler is exclusively associated with one electrostatics system.
- Filler sizes within the same electrostatics system are consistent.

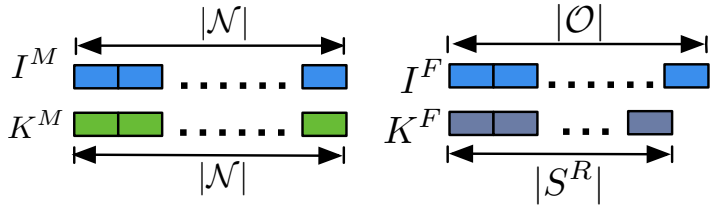
Previous Approach



Footprint Compression Technique

- For movable and fixed instances
  - I_i^M : the electrostatics system to which instance i is assigned
  - K_i^M : the size of the instance
- For fillers
  - I_i^F : the electrostatics system to which instance i is assigned
  - K_j^F : the size of the filler within electrostatics system

Our Approach



Space Complexity of our approach: $\boldsymbol{O(\#nodes + \#eSystems)}$

Reduce the GPU memory usage from 21.4GB to 3.9GB on case `Design_142` with 22 fence regions!

Divergence-aware Density Weight Scheduling

$$\min_{\boldsymbol{x},\boldsymbol{y}} L(\boldsymbol{x},\boldsymbol{y}) = \widetilde{W}(\boldsymbol{x},\boldsymbol{y}) + \sum_{s\in S} \lambda_s \mathcal{D}_s,$$
$$s.t.\ \mathcal{D}_s = \Phi_s + \frac{\mu}{2}\mathcal{P}_s\Phi_s^2, \forall s\ \in S_D \cup S_R$$

The update strategy for density weight $\boldsymbol{\lambda}$ also plays a crucial role in the result quality and optimization stability.

Two additional auxiliary variables

$$w_s^{(t+1)} = \sum_{i\in\mathcal{N}_s} |\partial\widetilde{W}^{(t)}/\partial x_i|_1, \quad \forall s \in S$$
$$d_s^{(t+1)} = |\nabla\mathcal{D}_s^{(t)}|_1, \quad \forall s \in S$$

where $\mathcal{N}_s$ represents all the nodes within the electrostatics system s.

Divergence-aware weight $\boldsymbol{\theta}$

$$\boldsymbol{\theta}^{(t+1)} = \max\left(1, \boldsymbol{d}^{(t+1)}/\boldsymbol{w}^{(t+1)}\right)/\boldsymbol{\lambda}^{(t)}$$
$$\boldsymbol{\lambda}^{(t+1)} = \min(\boldsymbol{u}^{(t+1)} \odot \boldsymbol{v}^{(t+1)}, \gamma/\boldsymbol{\theta}^{(t+1)})$$

When the optimization comes to the final stage, the large density gradient ($|\boldsymbol{d}| \gg |\boldsymbol{w}|$) is likely to cause the optimization to diverge.

$\boldsymbol{\theta}$ can effectively govern the growth rate of λ_s not to surpass a upper bound $\frac{\gamma}{d_s^{(t+1)}/w_s^{(t+1)}}$.

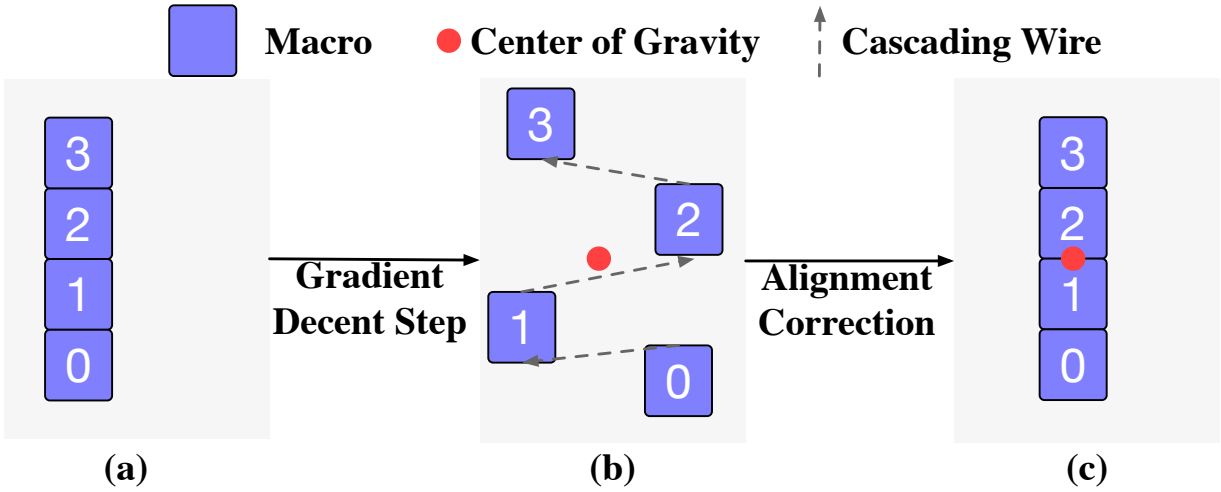
Cascaded Macro Shredding Technique

Pesudo Net Weighting within Cascaded Macro Group

Enlarge the weight of nets within the cascaded macro groups by a factor of two.

Macro Shredding

1. substitute large cascaded structures with multiple individual macros.
2. shred the cascaded macros into individual macros and update the placement
3. At the end of each iteration, arrange the shredded macros in a columnar shape based on the horizontal coordinates of their center of gravity.



Experimental Results

Experimental Setup (I)

Implementation

- C++ & Python
- Build upon `OpenPARF` [Mai+, ASICON'23] for agile development with GPU acceleration

Machine

- Intel(R) Xeon(R) Silver 4210R CPU (2.40 GHz, 10 cores)
- 512GB RAM
- One NVIDIA RTX 2080Ti GPU

Benchmark Suite

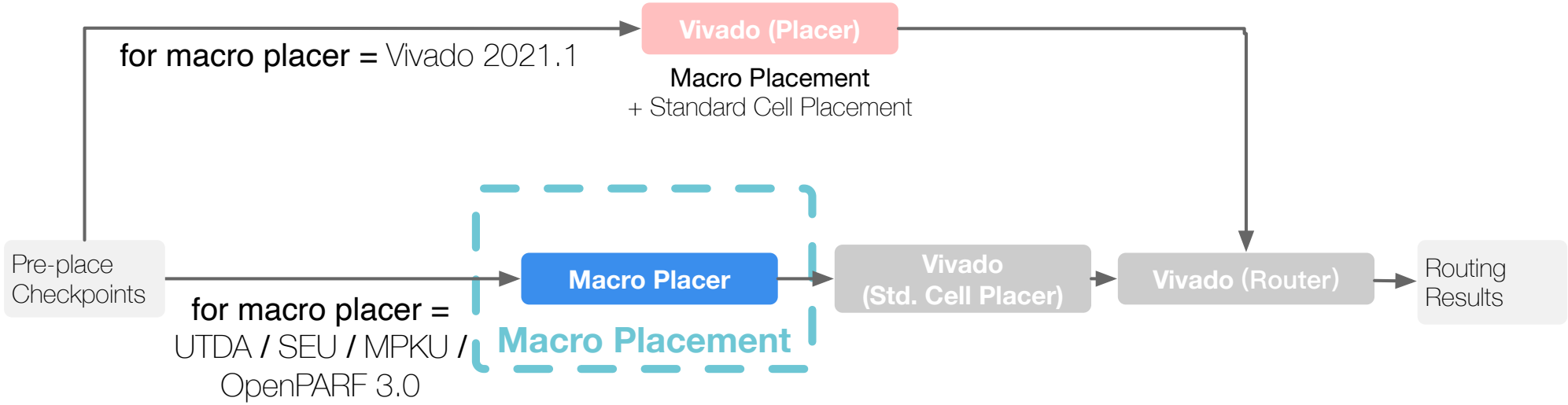
- MLCAD 2023 FPGA Macro Placement Contest [Bustany+, MLCAD'23]

Macro Placers for Comparison

- `Vivado 2021.1:` a representative FPGA placement commercial tool
- `UTDA`, `SEU`, and `MPKU:` top three winners in MLCAD 2023 FPGA Macro Placement Contest [Bustany+, MLCAD'23]

Experimental Setup (II)

Evaluation Flow



Evaluation Metrics

Metrics	Indication
Score	Overall evaluation of placement runtime, routing runtime, and routability [Bustany+, MLCAD'23]
Routability	Short, long, and global congestion
#ripup	Detailed routing overflow
RT_{mpl}	Macro placement runtime

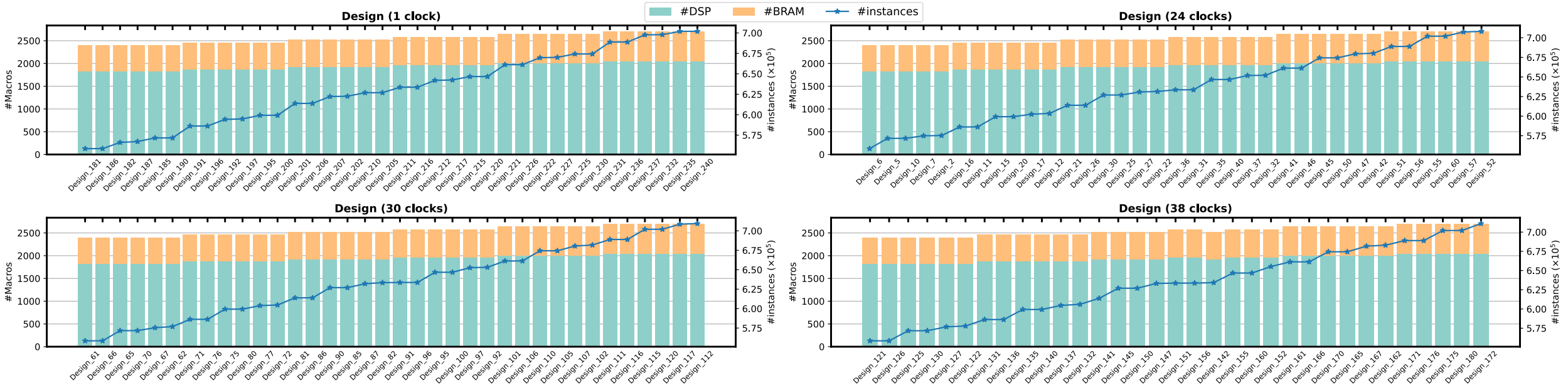
Experimental Setup (III)

Benchmark Statistics

- 16nm single-die UltraScale+ xcvu3p
- 140 synthetically generated designs
- Varying levels of difficulty
  - #clocks (1, 24, 30, 38)
  - Rent's Exponent (0.65, 0.67, 0.7, 0.72)

Designs	Statistics
Number	140
#Instances	558K-711K
#DSP+#BRAM	2.4K-2.7K
LUT (%)	70%-84%
FF (%)	38%-47%
BRAM (%)	80%-90%
DSP (%)	80%-90%
Rent[5]	0.65-0.72
#Regions	0-22
#Instances within Regions	0-285K
Instances within Regions (%)	0%-44.29%
Cascaded DSP Macros Size	{2, 5, 7, 10, 60} ×
Cascaded BRAM Macros Size	{2, 5, 7, 10, 30} ×

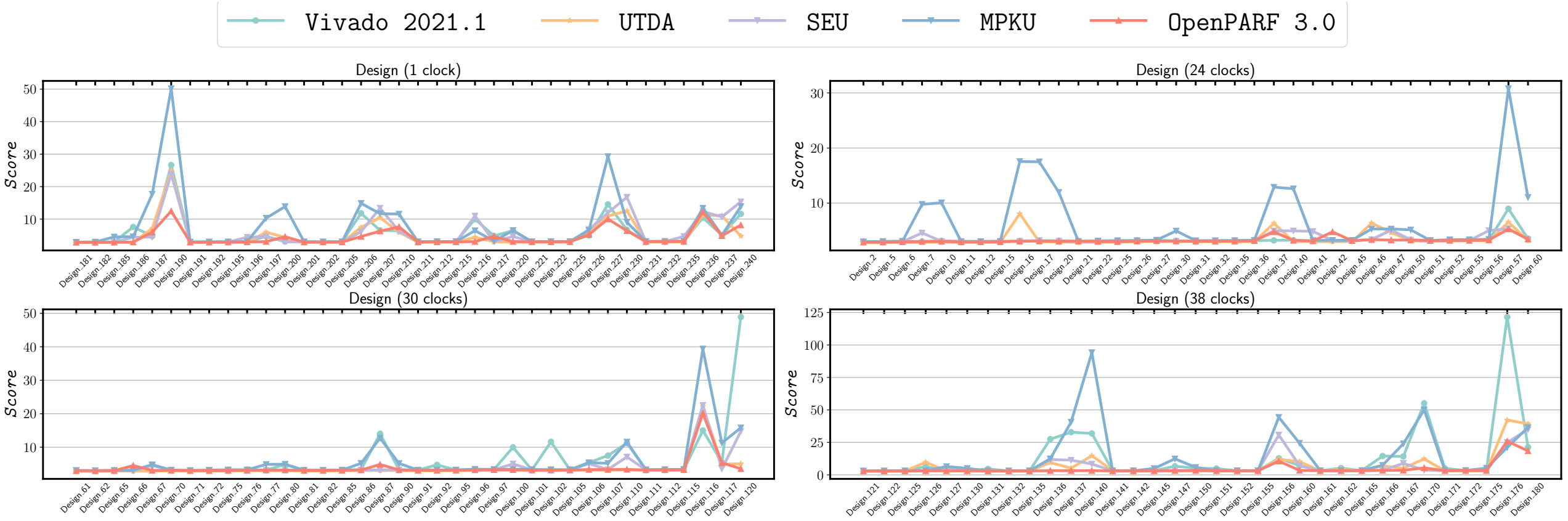
Benchmark Overview



Comparison with State-of-the-Art Placers

Overall *Score* Comparison on MLCAD 2023

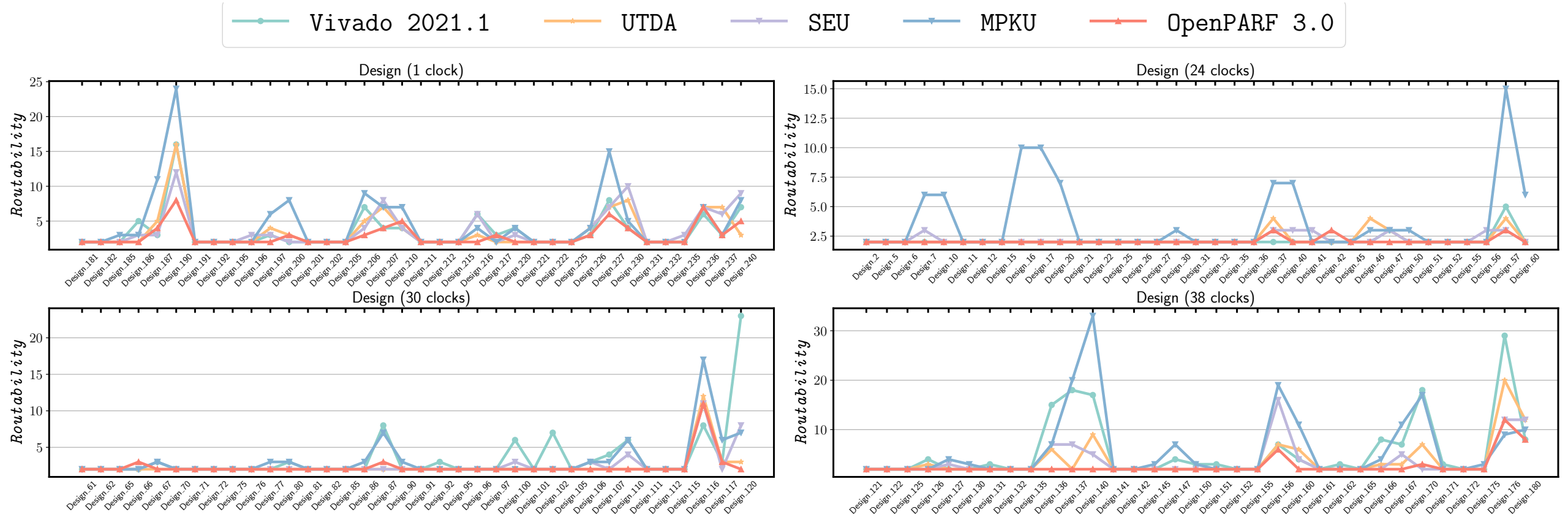
- 27.8% better than Vivado 2021.1
- 13.3% better than SEU
- 6.9% better than UTDA
- 49.1% better than MPKU



Comparison with State-of-the-Art Placers

Routability Comparison on MLCAD 2023

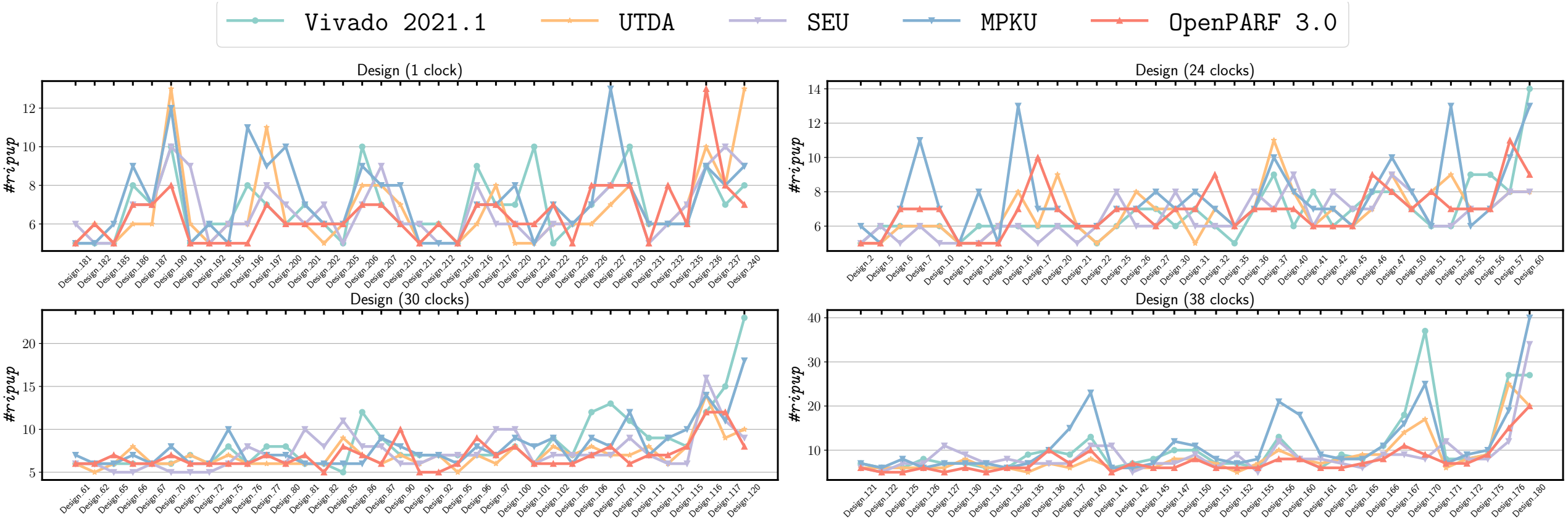
- 22.8% better than `Vivado 2021.1`
- 12.1% better than `SEU`
- 8.7% better than `UTDA`
- 39.1% better than `MPKU`



Comparison with State-of-the-Art Placers

Overall *#ripup* (detailed routing congestion) Comparison on MLCAD 2023

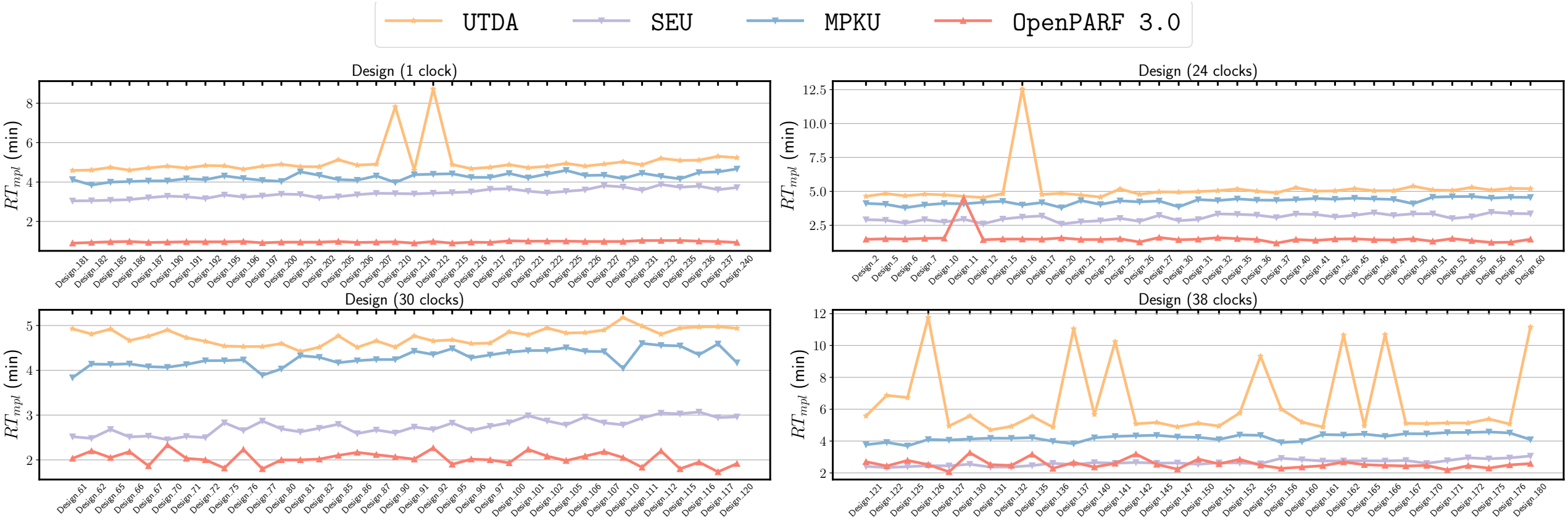
- 10.8% better than `Vivado 2021.1`
- 4.0% better than `SEU`
- 2.7% better than `UTDA`
- 16.8% better than `MPKU`



Comparison with State-of-the-Art Placers

Macro Placement Runtime RT_{mpl} Comparison on MLCAD 2023

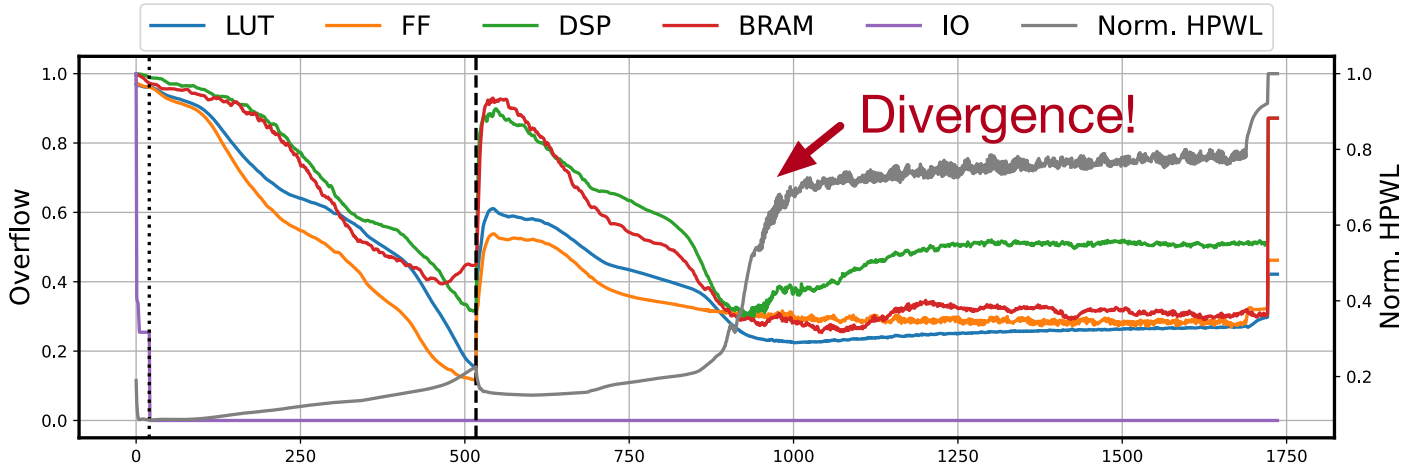
- 3.180X faster UTDA
- 2.599X faster than MPKU
- 1.808X faster than SEU



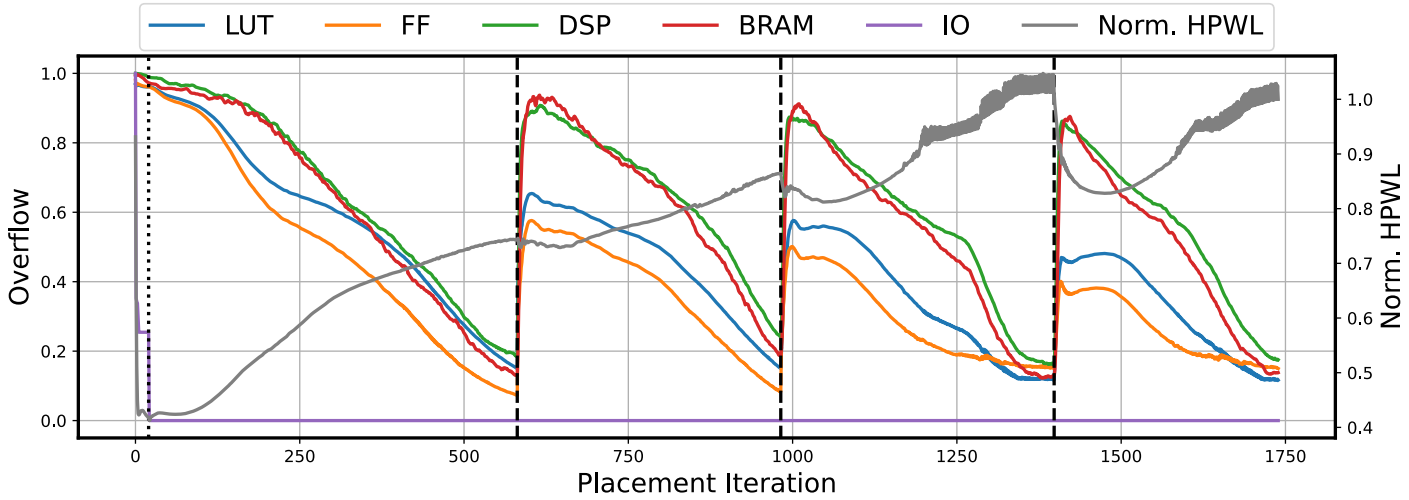
Robustness Evaluation

Overflows and HPWL Comparison on the `Design_156`

- The density weight updating method in [Gu+, ICCAD'20]



- Our proposed divergence-aware density weight scheduling



Conclusion & Future Work

Conclusion & Future Work

Conclusion

- **`OpenPARF 3.0`**: a robust FPGA macro placer considering cascaded macro groups and fence regions
- We resolve the fence region heterogeneity by a novel multi-electrostatics region model to handle the discontinuity of the solution space
- We propose a macro shredding technique to mitigate the size imbalance of cascaded macros
- We propose a dynamic density weight scheduling scheme to address robustness issues of divergence

Future Work

- Timing-driven / Routability-driven FPGA Macro Placement
- Reinforcement Learning for FPGA Macro Placement

THANK YOU!