AutoDMP: Automated DREAMPlace-based Macro Placement

Anthony Agnesina, Puranjay Rajvanshi, Tian Yang, Geraldo Pradipta, Austin Jiao, Ben Keller, Brucek Khailany, Haoxing Ren

Anthony Agnesina, Research Scientist | ISPD March 26-29, 2023

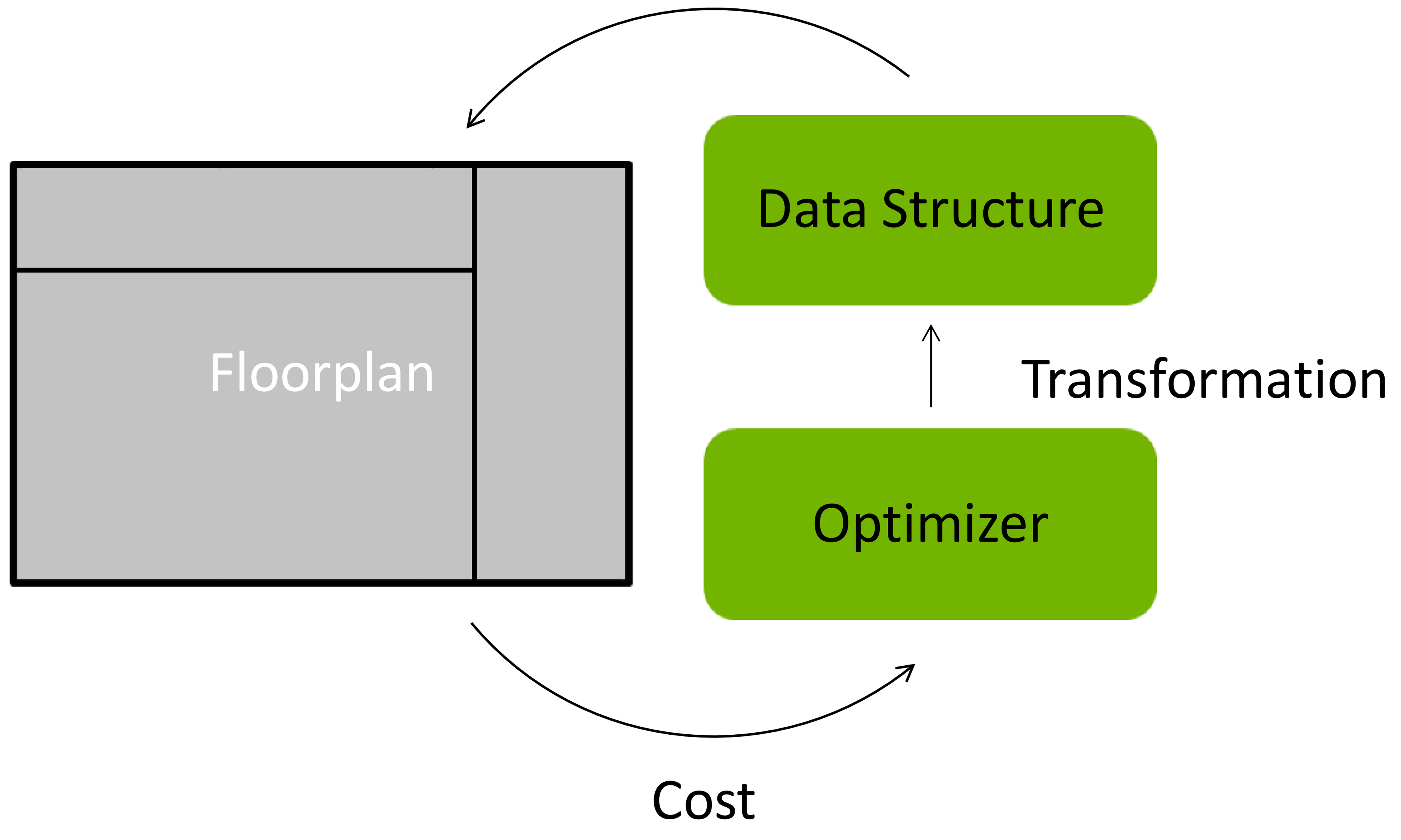
Agenda

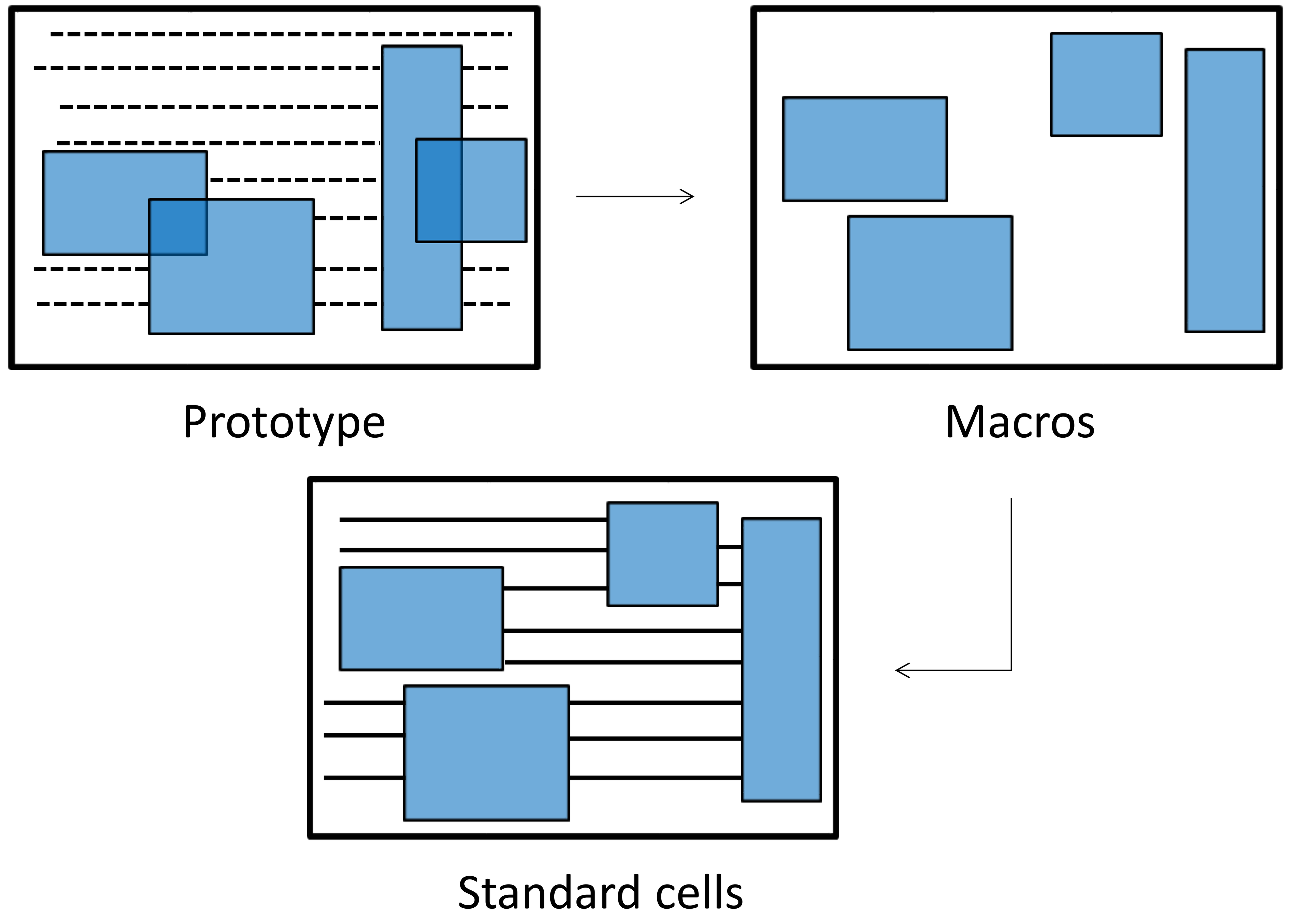
- Macro Placement / DREAMPlace
- Motivations
- DREAMPlace Extensions
- Multi-Objective Parameter Search
- Infrastructure
- Experiments
- Conclusion & Future Work

Macro Placement

Traditional methods

- Essential for Power-Performance-Area (PPA)
- **Floorplanning:** place large blocks **first**
  - Manual
  - Simulated Annealing
  - Top-down Partitioning
  - Reinforcement Learning
- **Mixed-Size:** place macros and standard cells **concurrently**
  - EDA practice
  - Simultaneous vs. Sequential flow





DREAMPlace

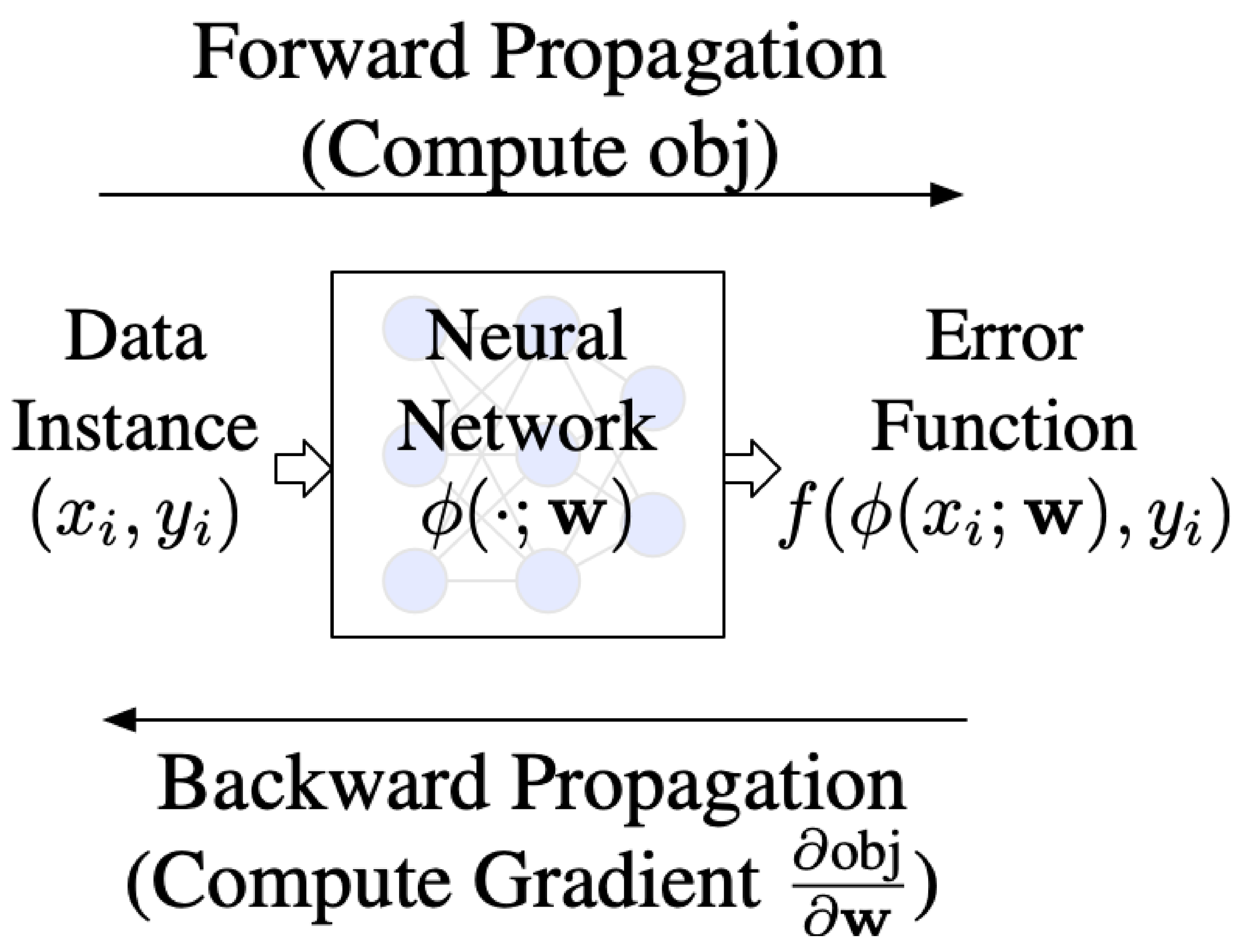
Analytical & nonlinear placer

- Constrained optimization with Lagrange

$$\min_{\mathbf{z}} \underbrace{\mathrm{WL}(\mathbf{z})}_{\text{wirelength}} \text{ s.t. } \underbrace{\mathrm{D}(\mathbf{z})}_{\text{density}} \leq t_d \longrightarrow \min_{\mathbf{z}} \mathrm{WL}(\mathbf{z}) + \lambda\, \mathrm{D}(\mathbf{z})$$

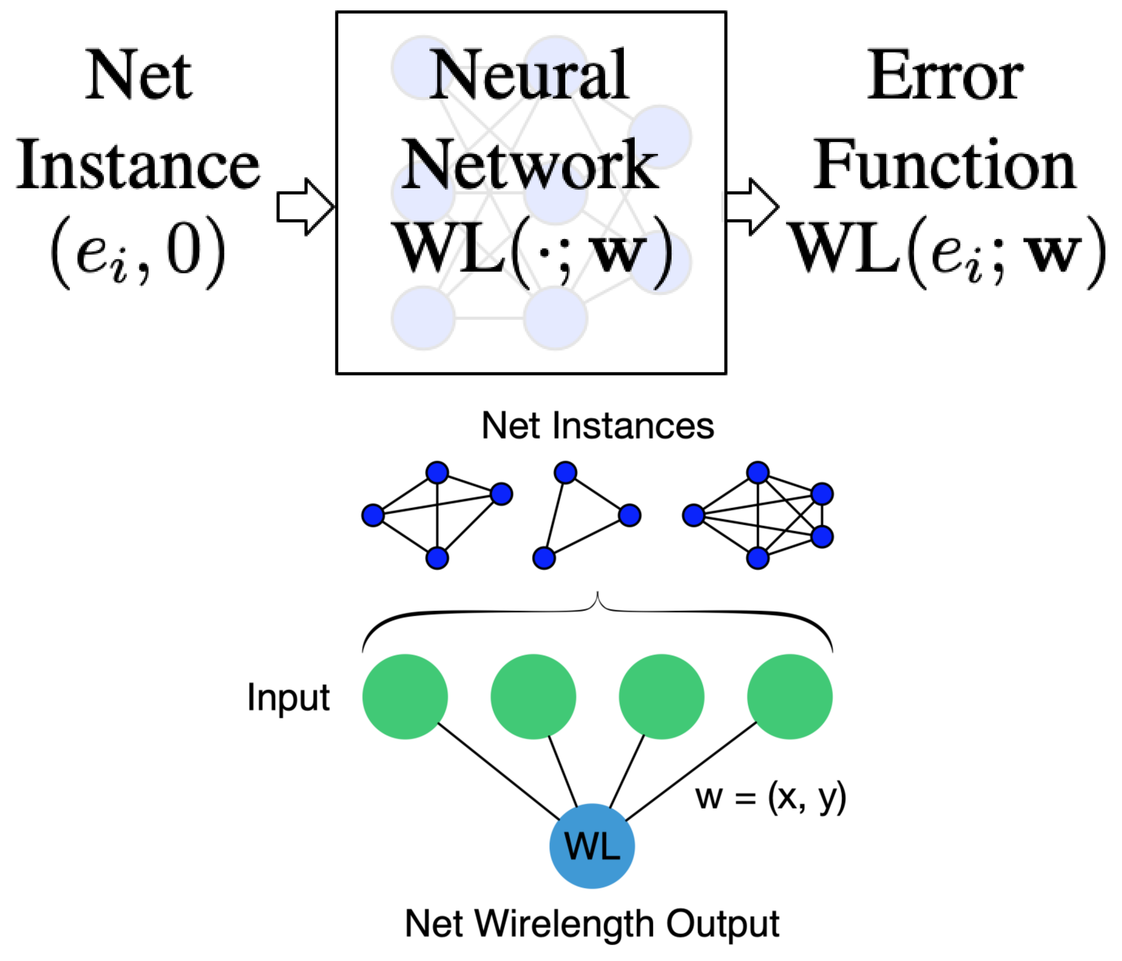
- Vast mathematical subject
  - Objective: **Nonlinear**, Linear, Quadratic, etc.
  - Solution: **Gradient Descent**, Simplex, Ellipsoid, etc.
- Neural Network Analogy
  - Joint work between UT Austin and NVIDIA
  - Accelerate ePlace/RePlace with GPUs/PyTorch

$$\min_{\mathbf{w}} \sum_{i}^{n} f(\phi(x_i; \mathbf{w}), y_i) + \lambda R(\mathbf{w})$$



Train a Neural Network

$$\min_{\mathbf{w}} \sum_{i}^{n} \mathrm{WL}(e_i; \mathbf{w}) + \lambda D(\mathbf{w})$$



Solve a Placement

AutoDMP

Motivations

- DREAMPlace is a **superfast** mixed-size placer
  - Treats macros & standard cells similarly → macro **legalization issues**
  - High influence of parameter settings on optimization quality

- **AutoDMP → generate quickly diverse high-quality macro placements**
  - Enhance DREAMPlace: fix legalization issues & expand the design space
  - Tune DREAMPlace settings with Multi-Objective Bayesian Optimization

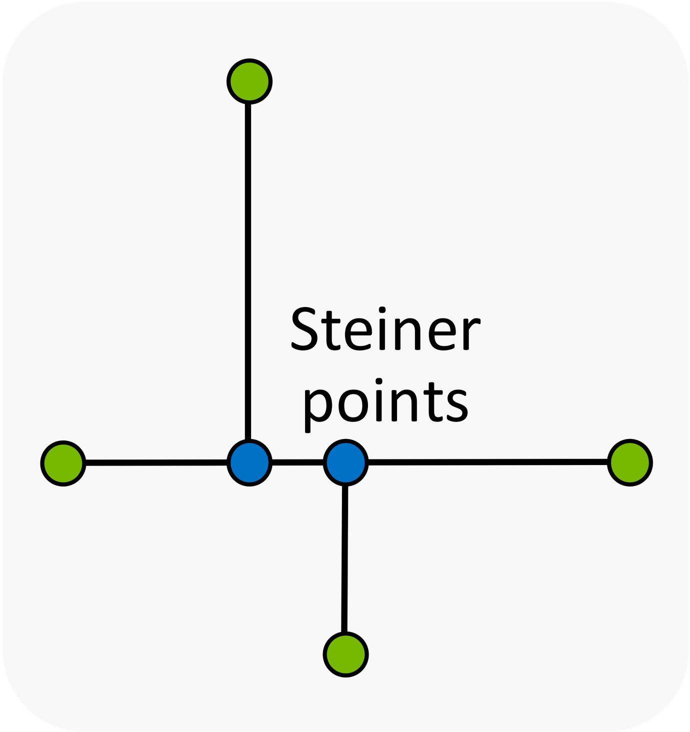
PPA Proxies

What is a good macro placement?

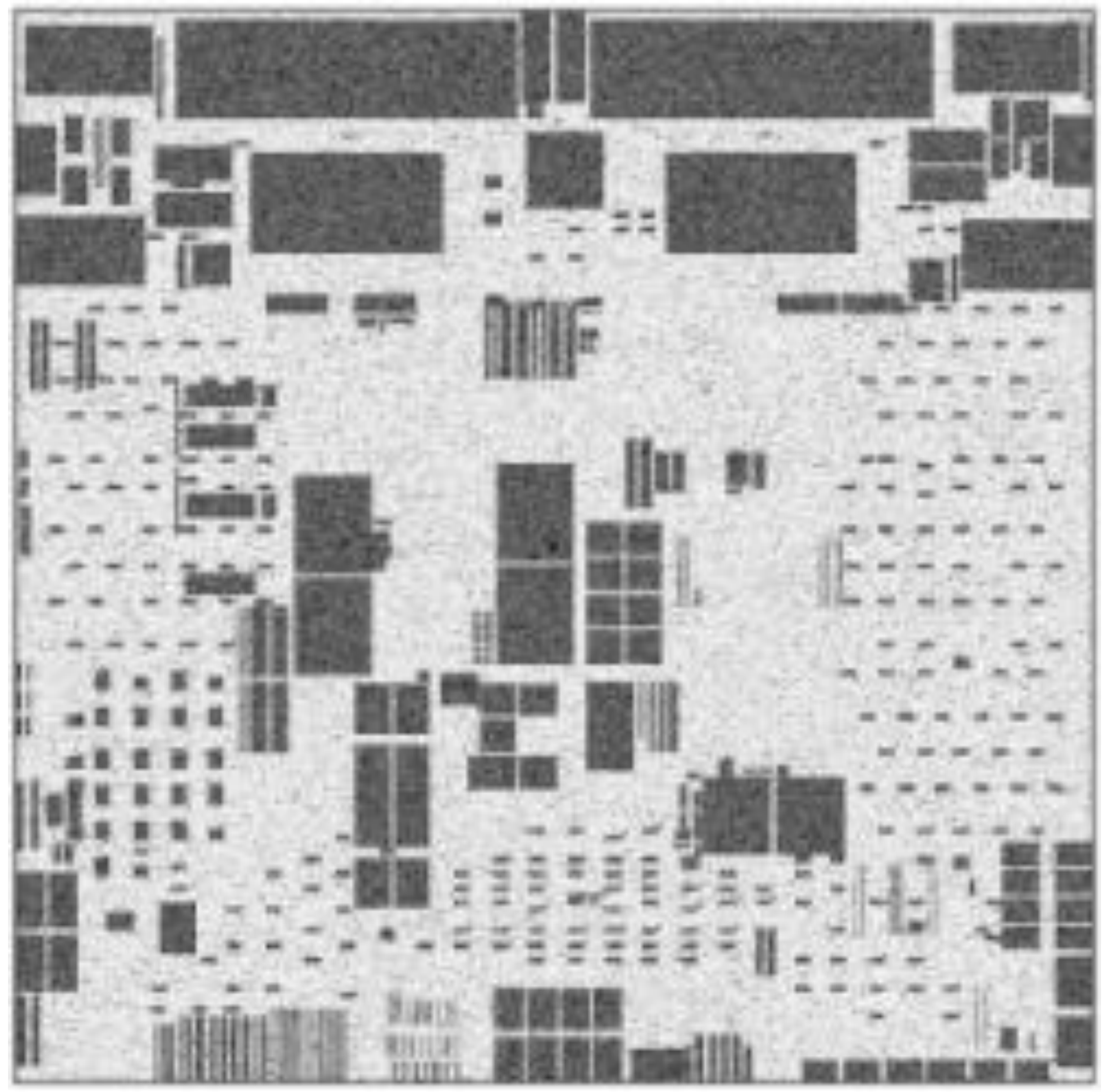
- Downstream steps chaos
- **AutoDMP → high-level proxies post-detailed placement (ABCDPlace)**
  - Wirelength: rectilinear Steiner minimum tree (RSMT)
  - Cell density: ePlace uniform density target

$$\mathrm{D}(\mathbf{z}) = \sum_{b \in B} \rho_b \psi_b \sim t_d\,|B|$$

  - Congestion: rectangular uniform wire density (RUDY) map



RSMT

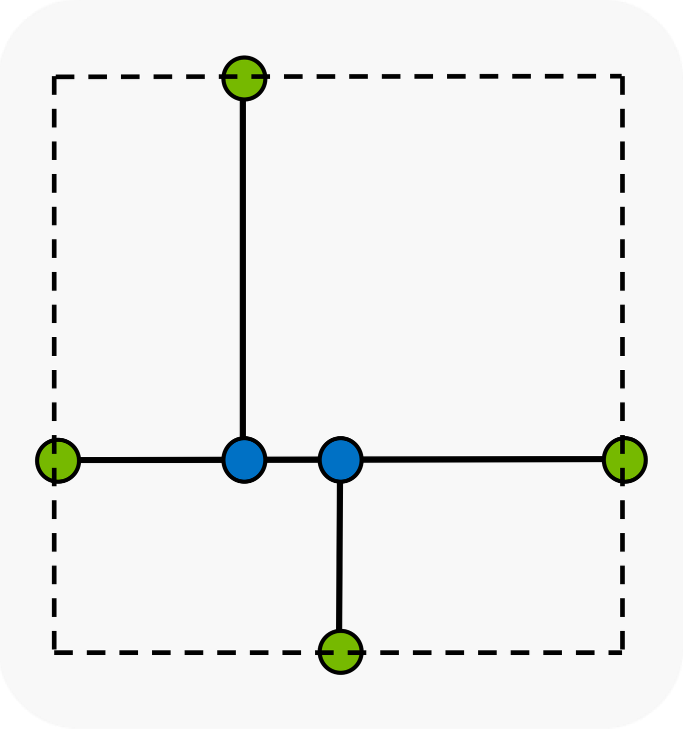


eDensity

DREAMPlace Extensions

- Weigh the smooth half-perimeter wirelength (HPWL) of nets
  - Improve correlation with RSMT during global placement
  - Based on pin count (RISA)

$$\mathrm{WL}(\mathbf{z}) = \sum_{e\in E} w(|e|)\mathrm{WL}(e;\mathbf{z})$$



HPWL

- Greedy macro-orientation refinement during detailed placement

- RUDY + Macros + Gaussian filter
  - Congestion score = average of top-10% most congested bins

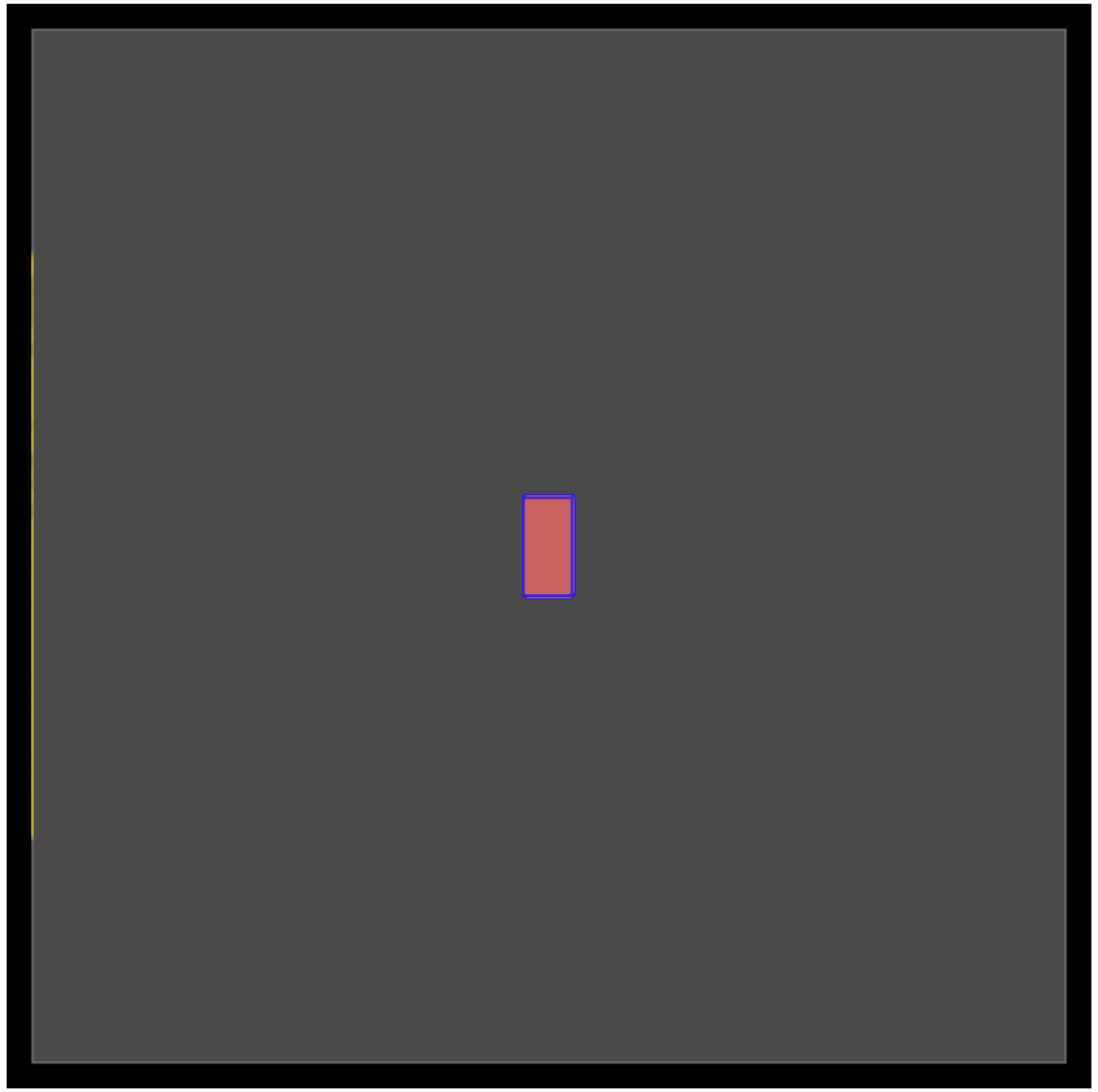
$$\mathrm{RUDY}(g) = \frac{\overbrace{\boxed{\sum_{e\in E}\frac{\mathrm{OA}(e,g)}{\mathrm{BBOX}(e)}}}^{\text{Net Demand}}}{\underbrace{\boxed{s(g)}}_{\text{Routing Supply}} - \underbrace{\boxed{\sum_{m\in M}\alpha(m)\mathrm{OA}(m,g)}}_{\text{Macro Demand}}}$$

Parameter Space

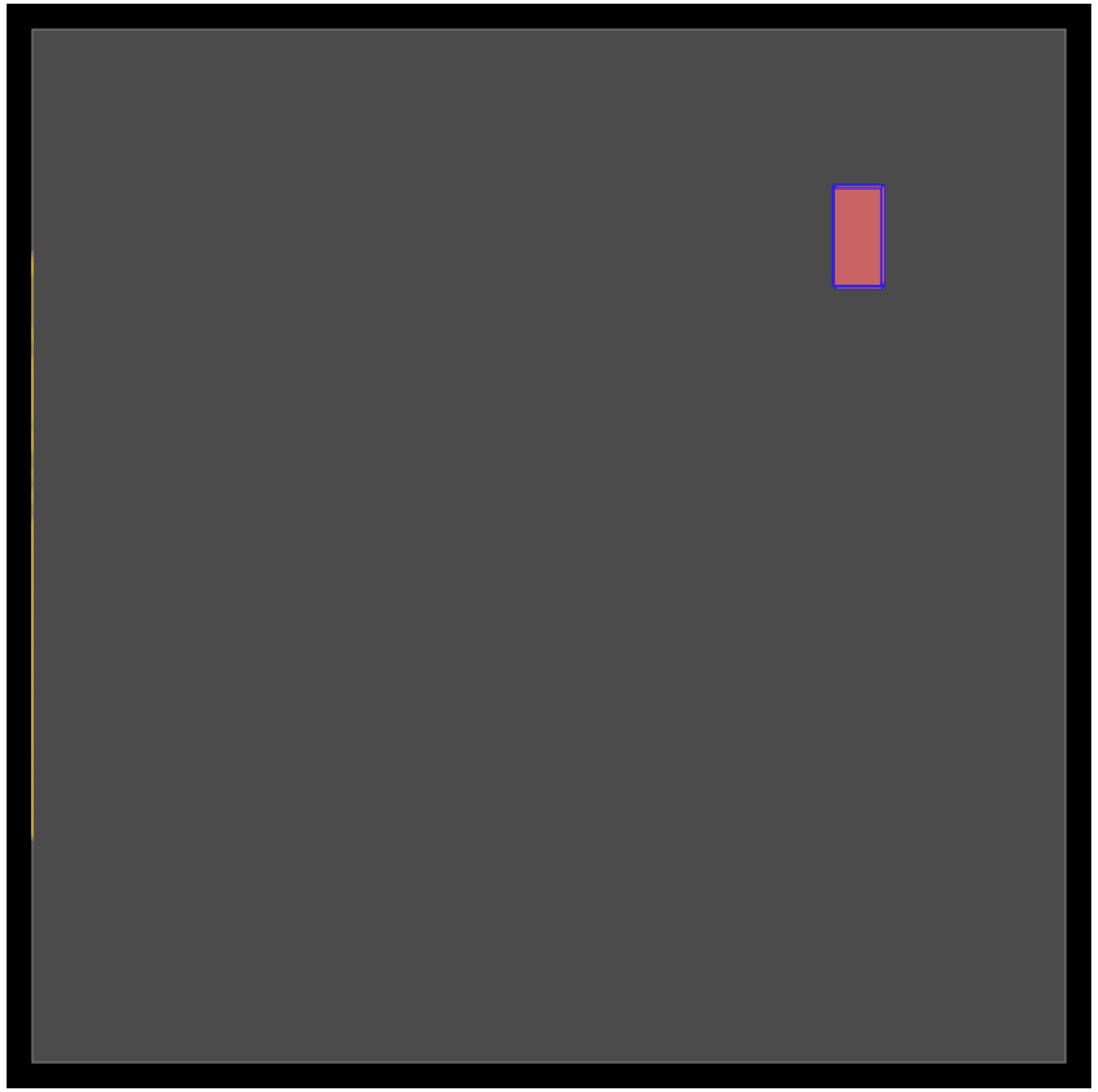
Parameter	Search Range	$\widehat{c_V}$ (%) RSMT	$\widehat{c_V}$ (%) Cong.	Divg. Rate
*horiz. initial position	[0.2, 0.8] (%)	2.2	0.9	0.0
*vert. initial position	[0.2, 0.8] (%)	2.0	1.1	0.0
*horiz. macro halo	technology dep.	1.8	1.3	0.0
*vert. macro halo	technology dep.	1.7	1.2	0.0
target density d_{target}	$[a_{\text{util}} - 0.2, a_{\text{util}}]$ (%)	-	-	-
density weight	$[1e^{-6}, 1.0]$	3.1	1.7	0.0
smooth HPWL model	{LSE, WA}	0.7	1.1	0.0
smooth HPWL initial γ_0	[0.10, 0.50]	5.1	1.9	0.0
GD initial LR lr_0	$[1e^{-4}, 1e^{-2}]$	1.4	1.0	0.0
GD LR decay	[0.99, 1.0]	6.7	2.3	53.2
GD optimizer	[Adam, Nesterov]	1.2	0.8	54.2
# horiz. global bins	{256, 512, 1024, 2048}	1.3	0.9	0.0
# vert. global bins	{256, 512, 1024, 2048}	3.1	1.3	21.1
λ update lower coeff. L	[0.90, 0.99]	4.2	1.9	0.0
λ update upper coeff. U	[1.01, 1.15]	27.0	7.5	1.8
λ update $\Delta\,\text{HPWL}_{REF}$	$[1.5e^5, 5.5e^5]$	2.3	1.2	0.0

- Base DREAMPlace parameters
  - Optimization-related
  - Density target
- Initial macro/cell locations → placement **diversity**
- Macro halos → **fix legalization** issues
- Large parameters effects

Initial Positions



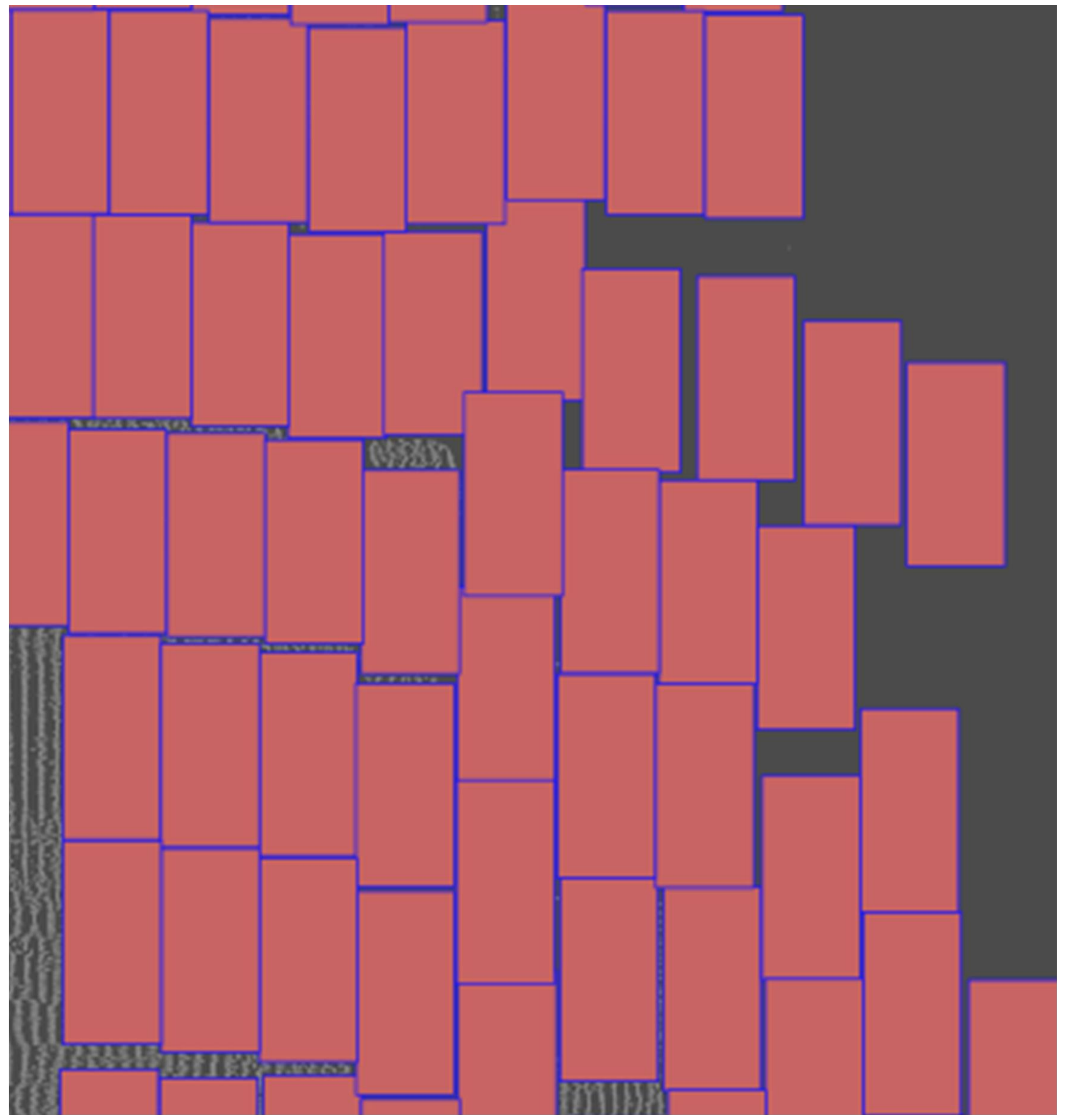
Center Position



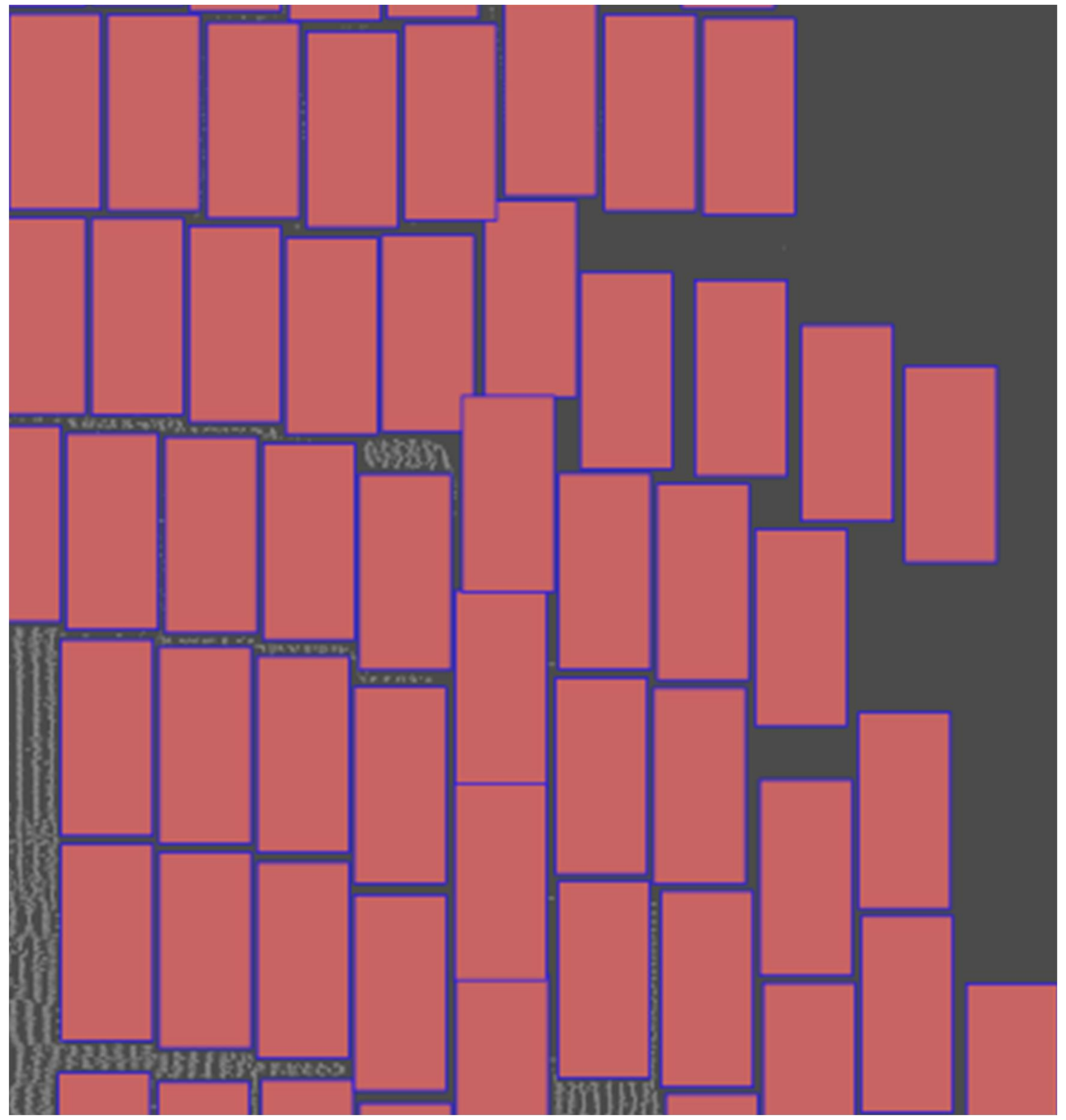
Upper Right Position

Macro Halos

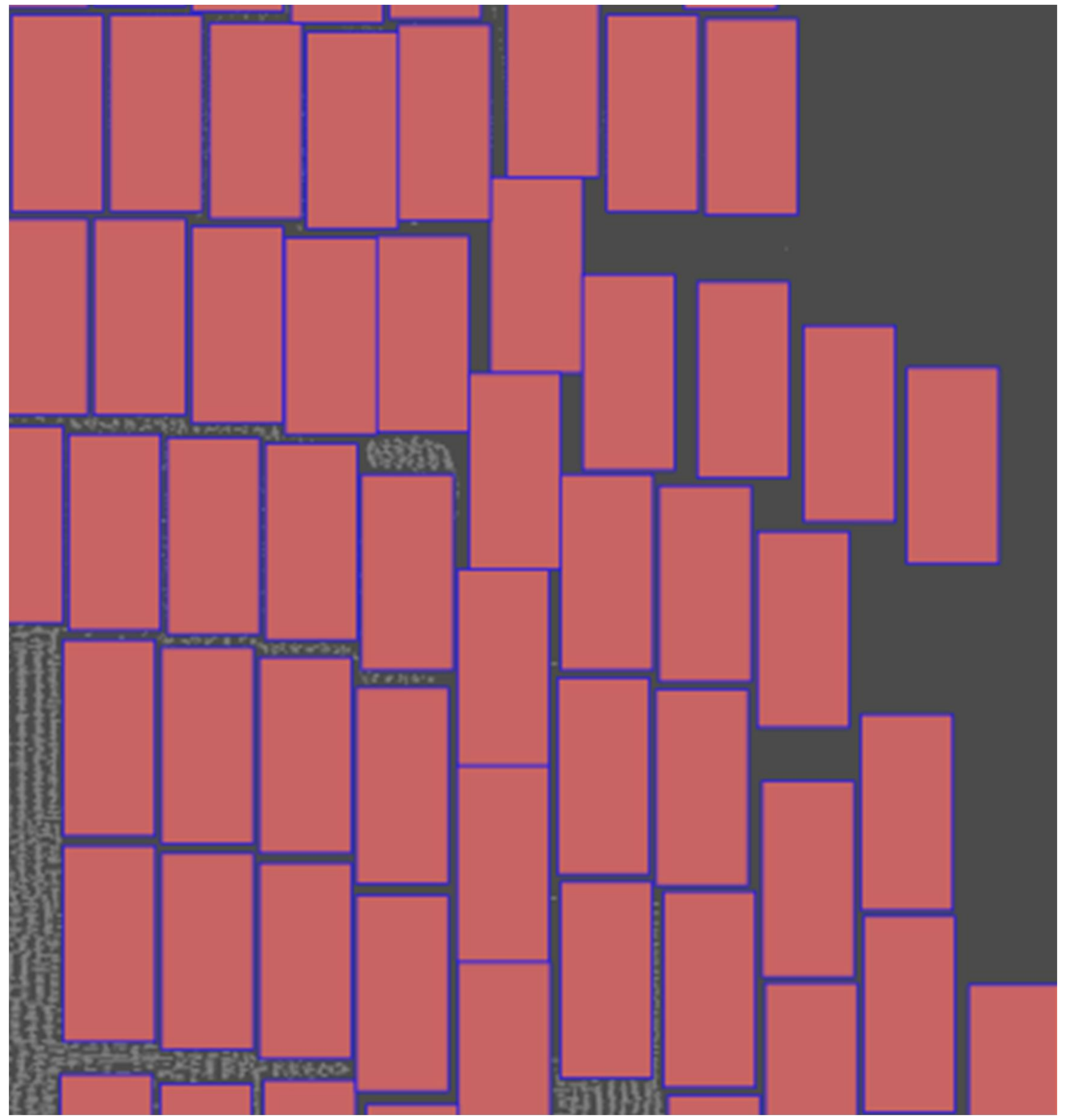
- Macros overlap at the end of global placement → large macro displacements & standard cells disruption
- “Buffer” zones removed before detailed placement



Macros with Halos



Removing Halos



Legalized Macros

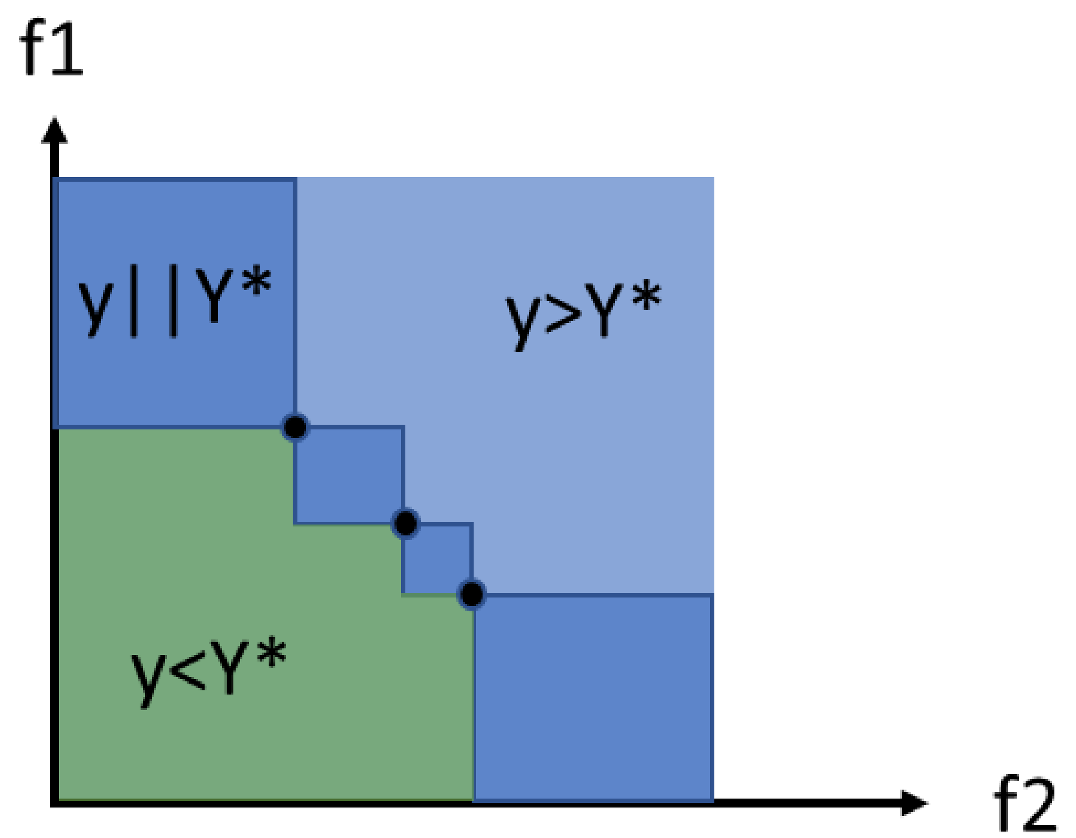
Multi-Objective Bayesian Optimization

DREAMPlace parameter tuning

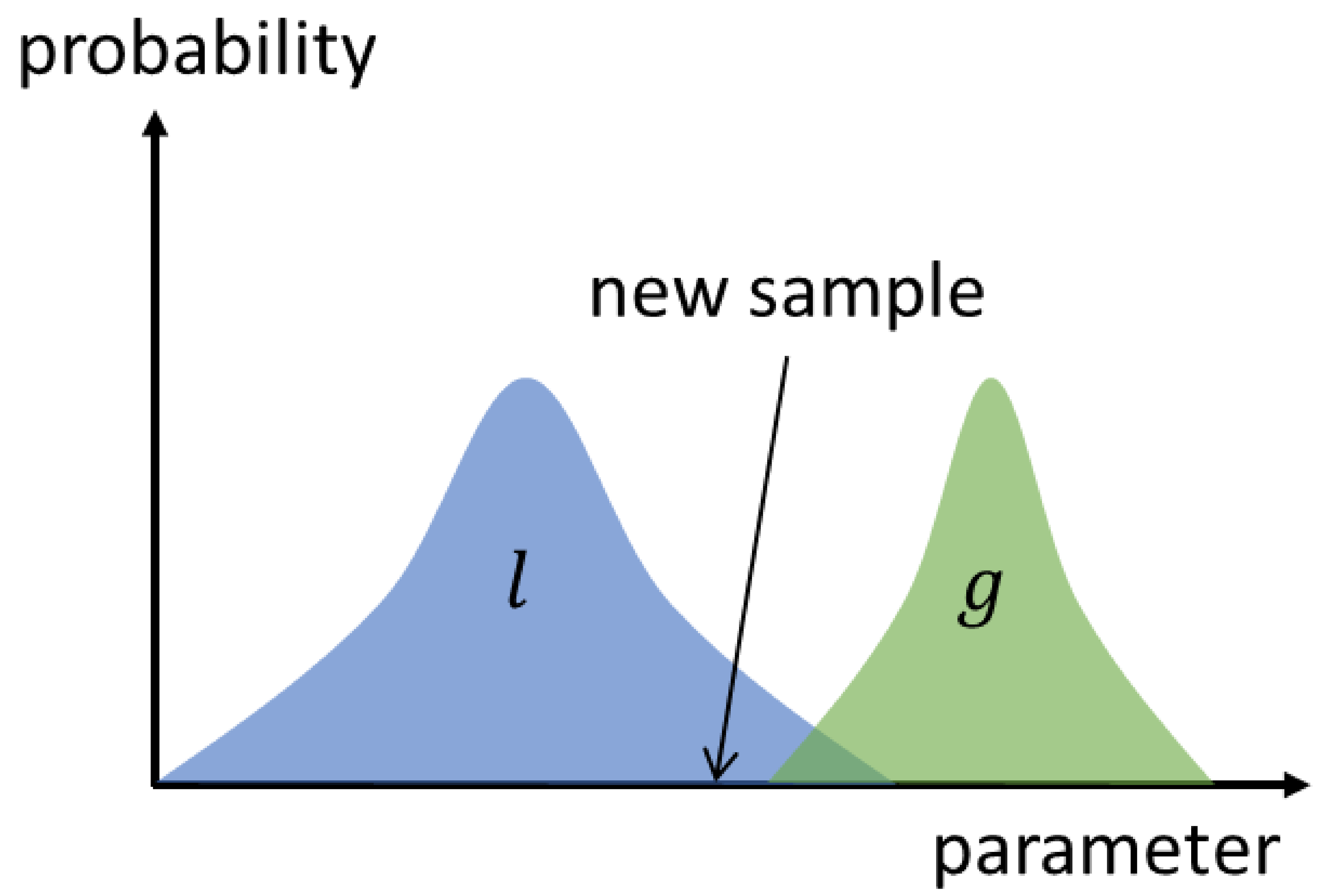
- Search for **multiple competing objectives**: RSMT, density, congestion
  - Limited scalarized method

$$\text{Cost}(\boldsymbol{p}) = \text{RSMT} + \underset{?}{\beta}\ \text{Density} + \underset{?}{\mu}\ \text{Congestion}$$

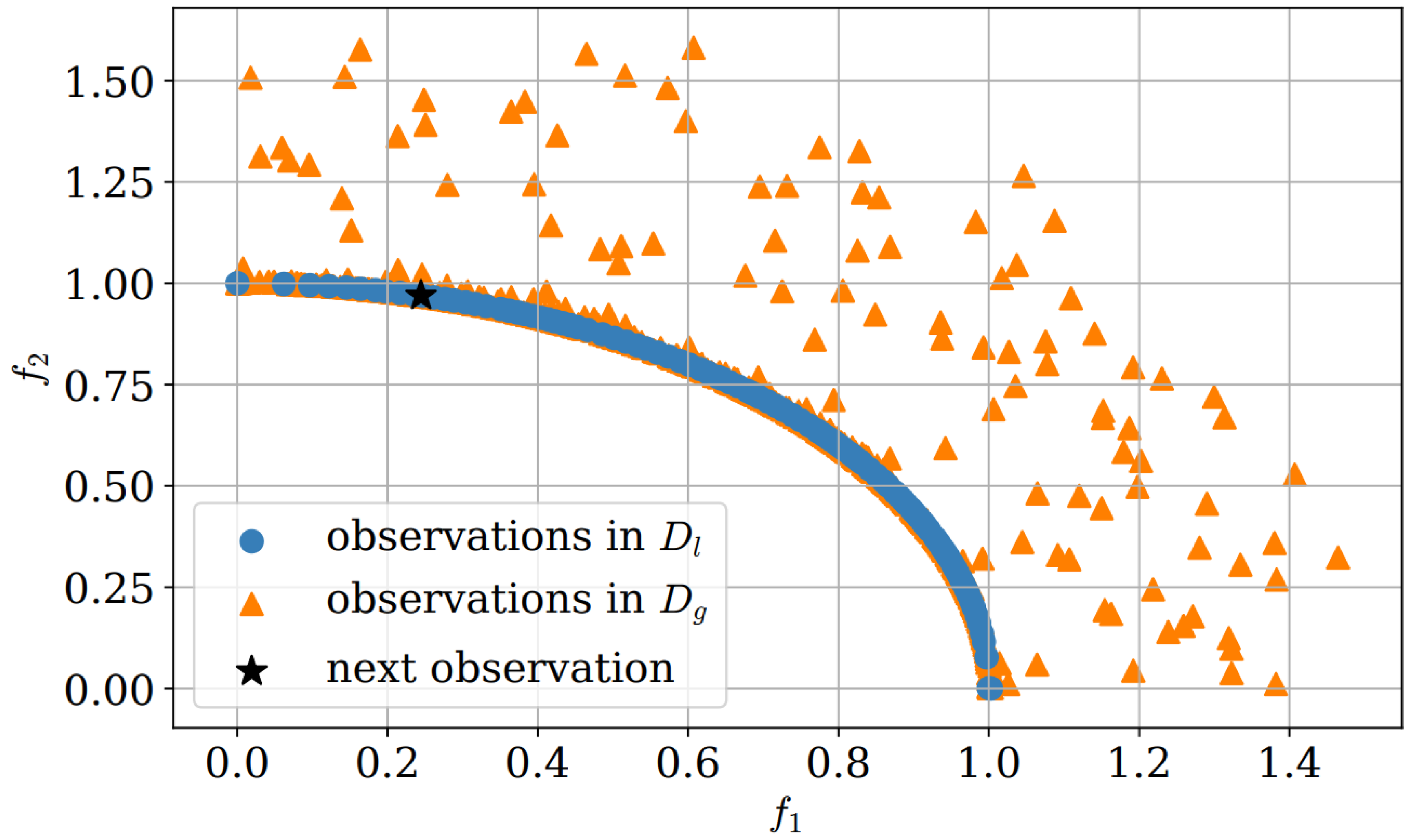
  - True multi-objective: **Pareto**-front modeling of PPA trade-offs
    - Tree-structured Parzen Estimator **(MOTPE)** for kernel density estimation of good/bad samples



(a)



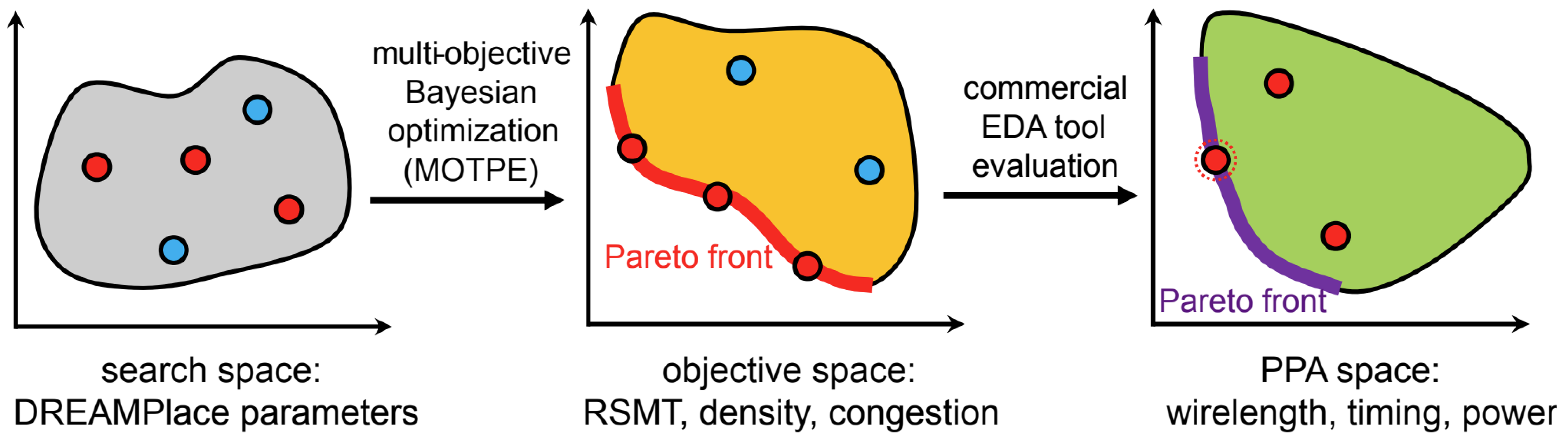
(b)



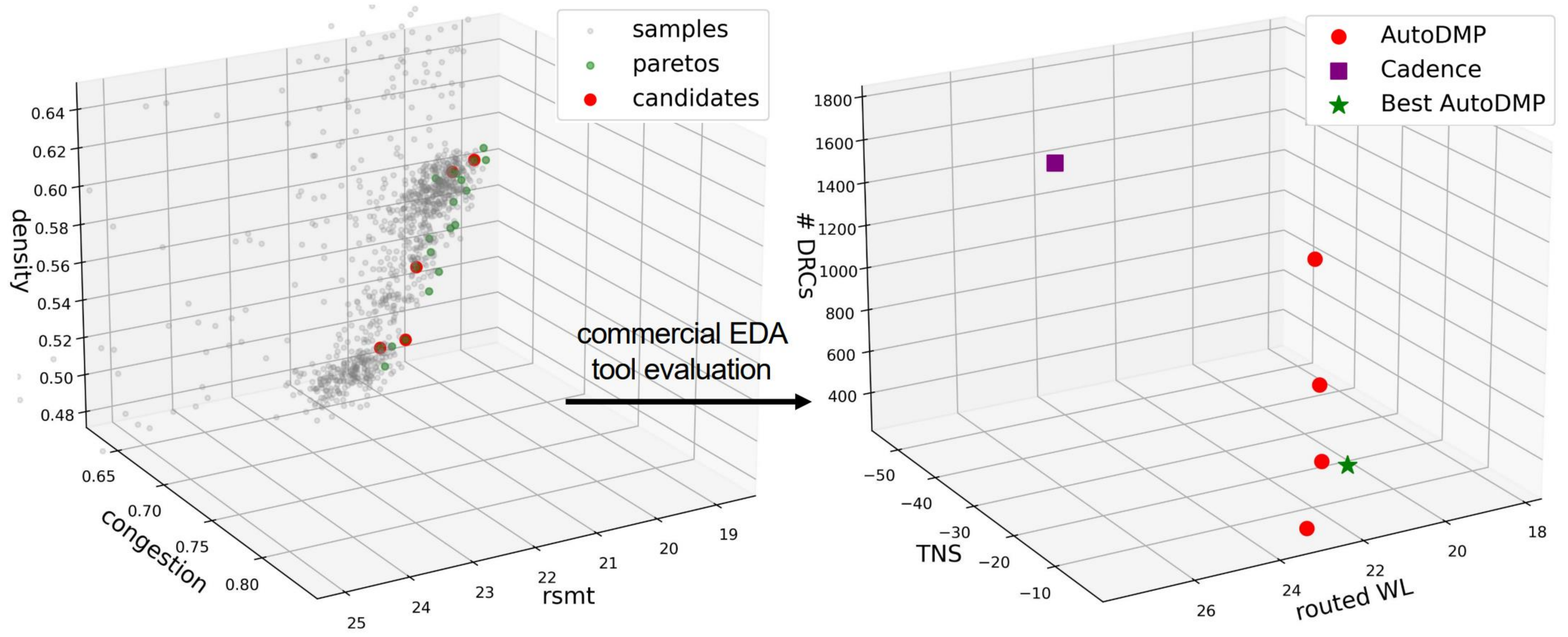
Y. Ozaki, Y. Tanigaki, S. Watanabe, and M. Onishi. Multiobjective Tree-Structured Parzen Estimator for Computationally Expensive Optimization Problems

Two-Level PPA Evaluation

- MOTPE maps **placement space** to **proxy space**

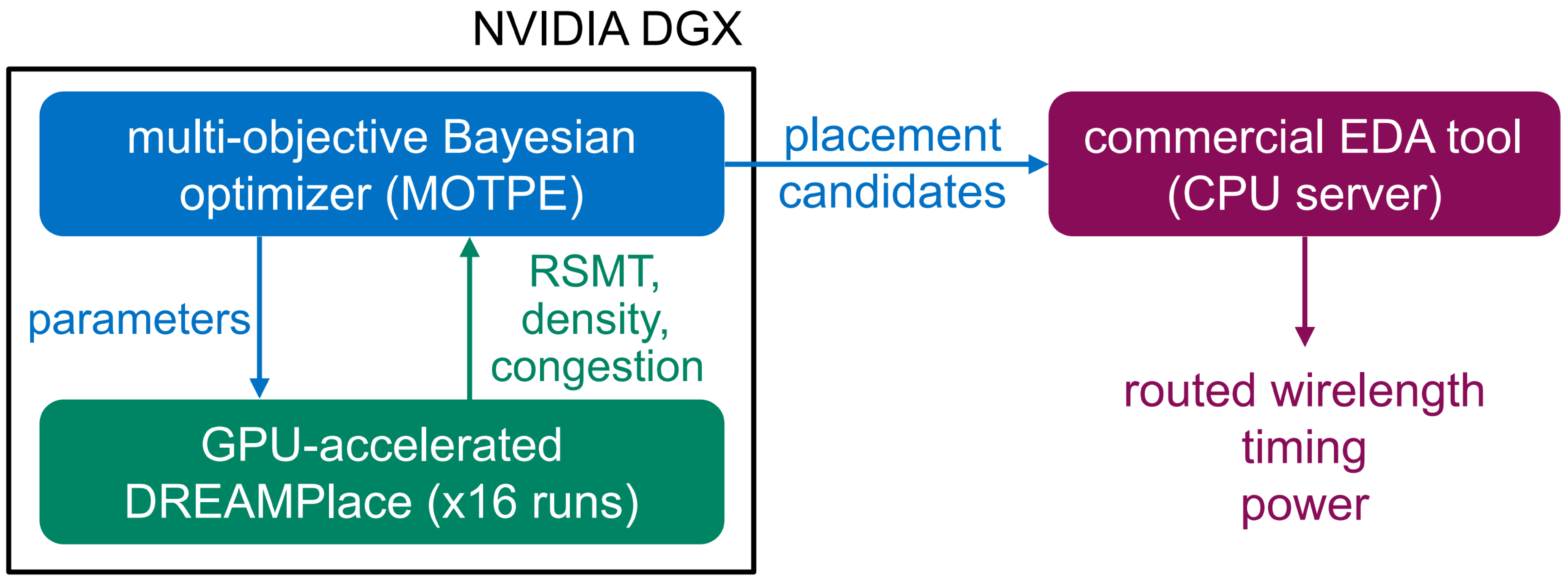


- EDA tool maps **good proxy trade-offs** to **PPA space**
  - **K-means** clustering of Pareto points to reduce EDA tool usage



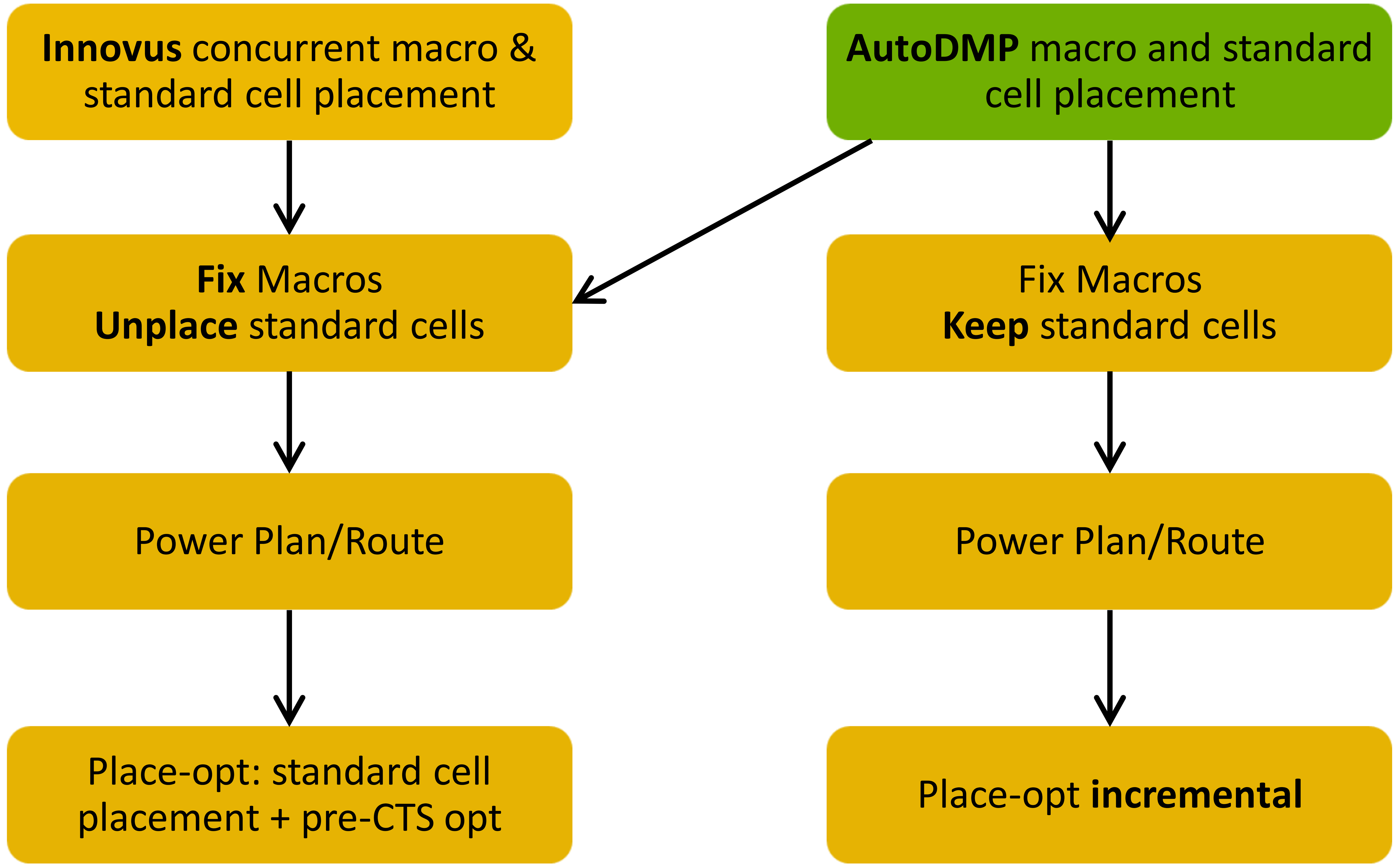
AutoDMP Infrastructure

- Multiple DREAMPlace on an **NVIDIA DGX A100 Station**
- Highly **scalable**: 1,000 samples ~ **3h** for a design with 2.7M cells & 320 macros



Experimental Settings

- Open-source **TILOS** EDA flow and macro placement benchmark
  - Reference Cadence Innovus is a mixed-size sequential flow
  - AutoDMP placement used in two ways in the EDA flow

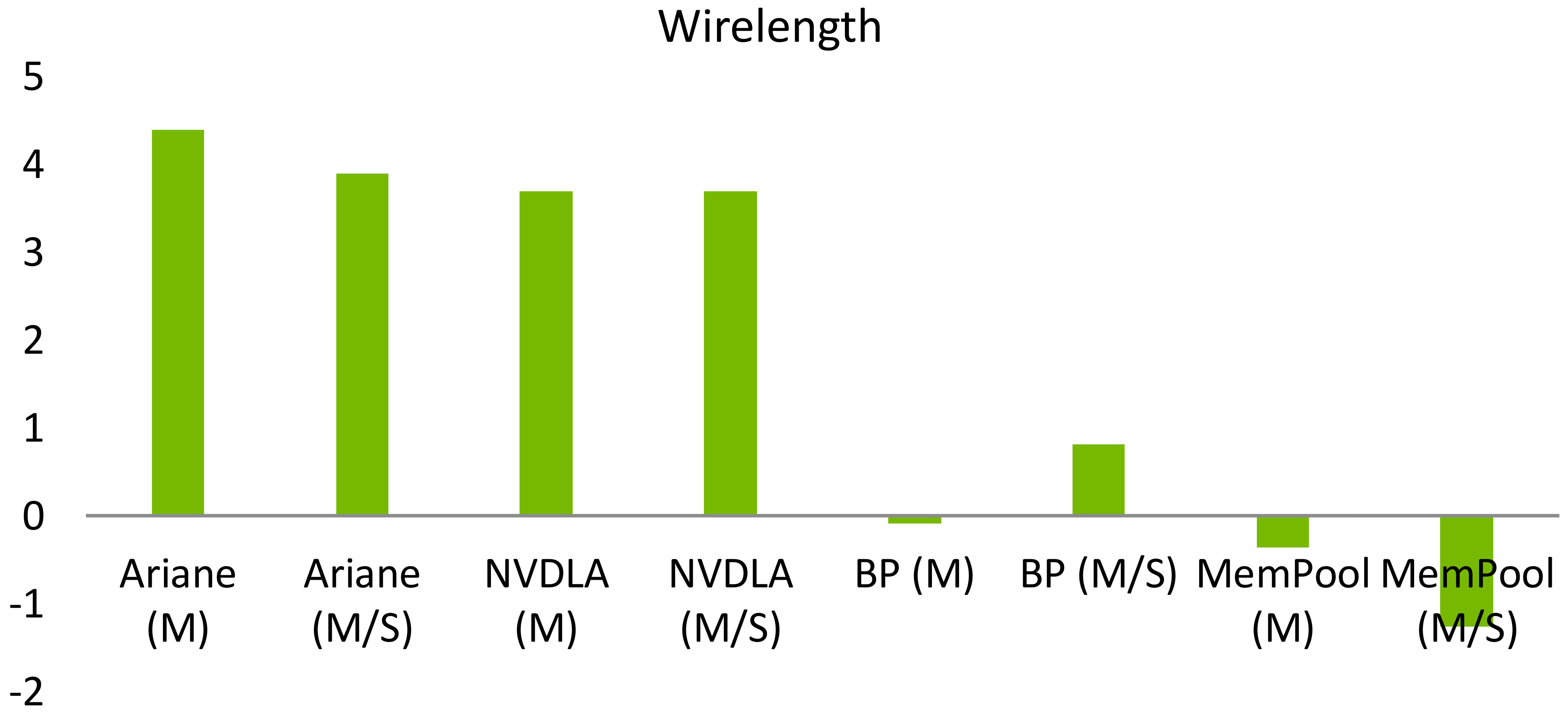


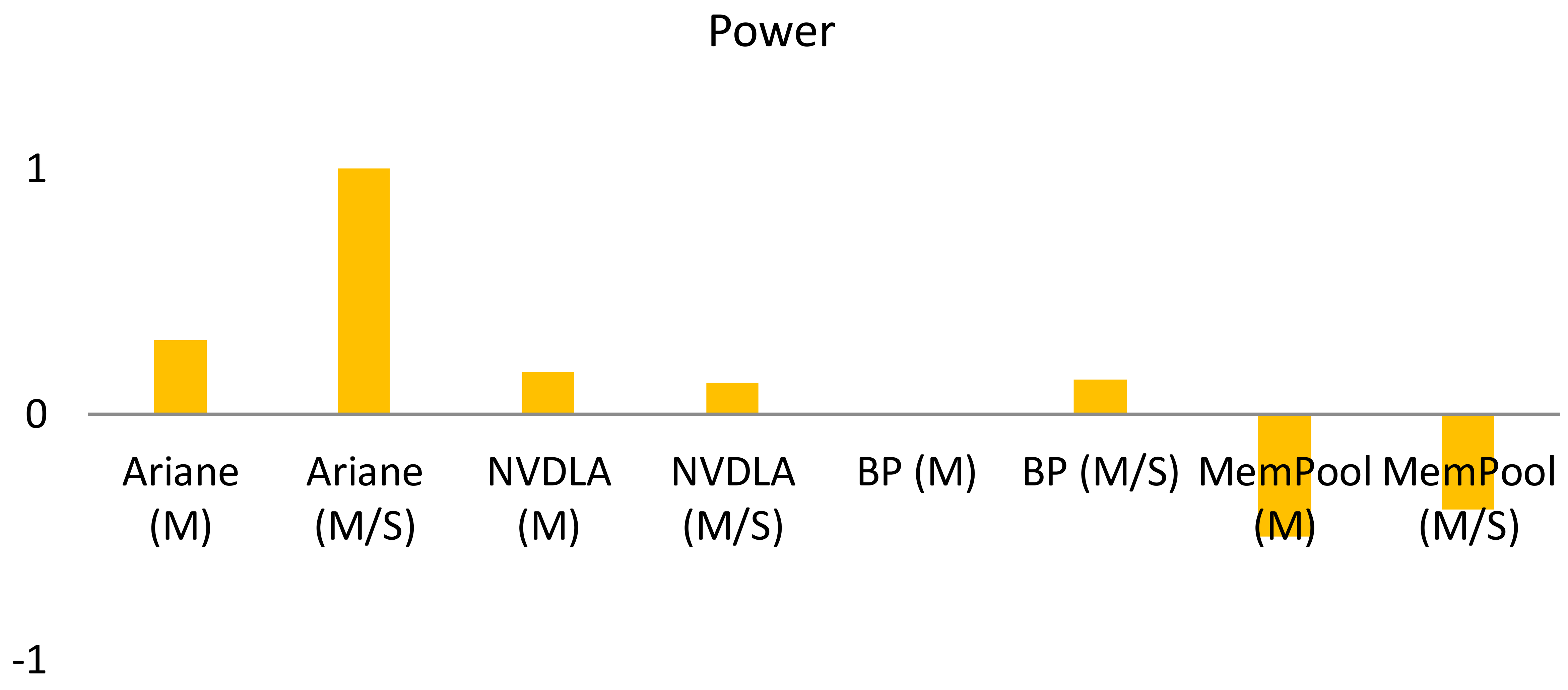
TILOS Reference Flow

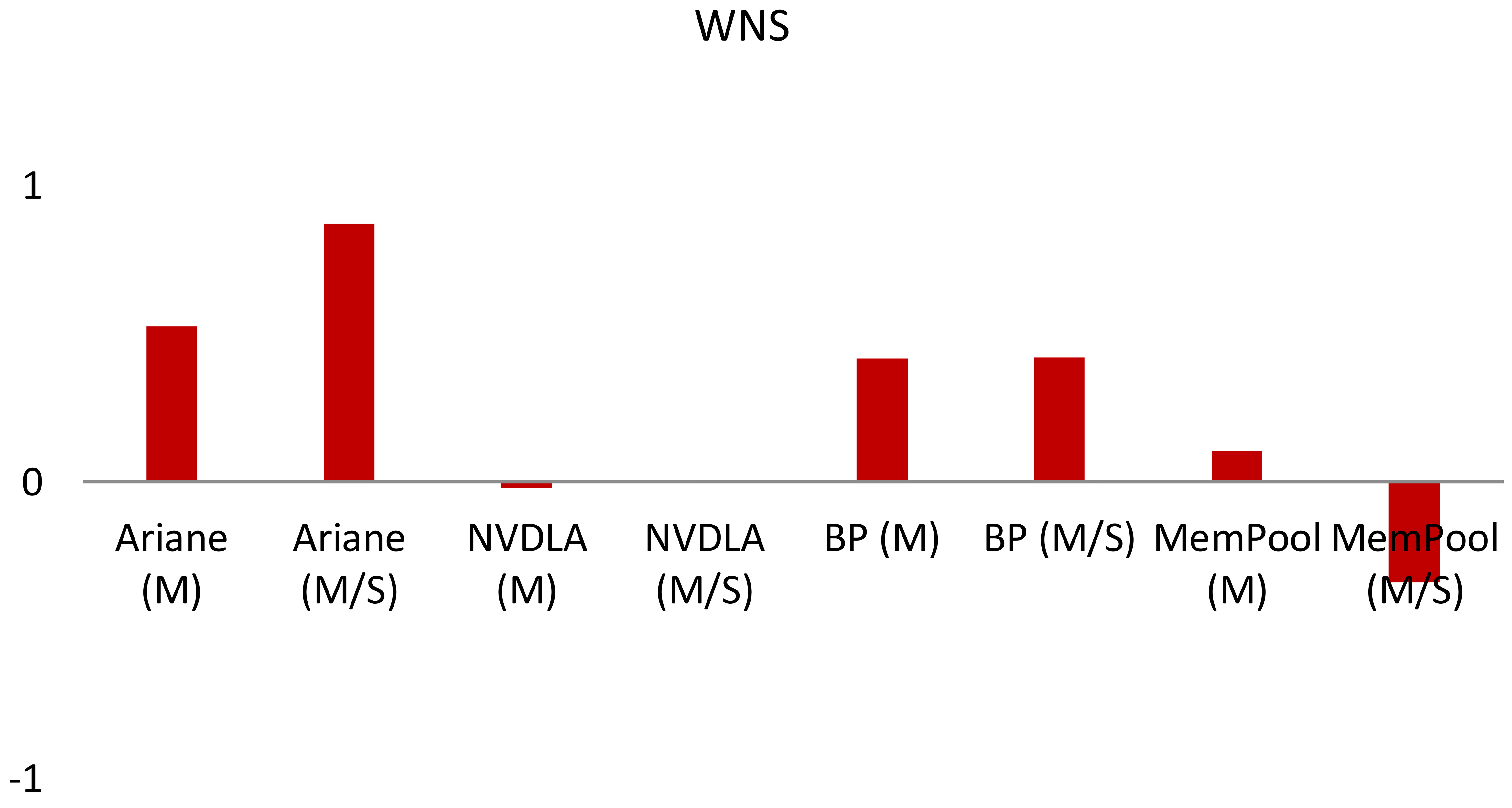
AutoDMP Integration

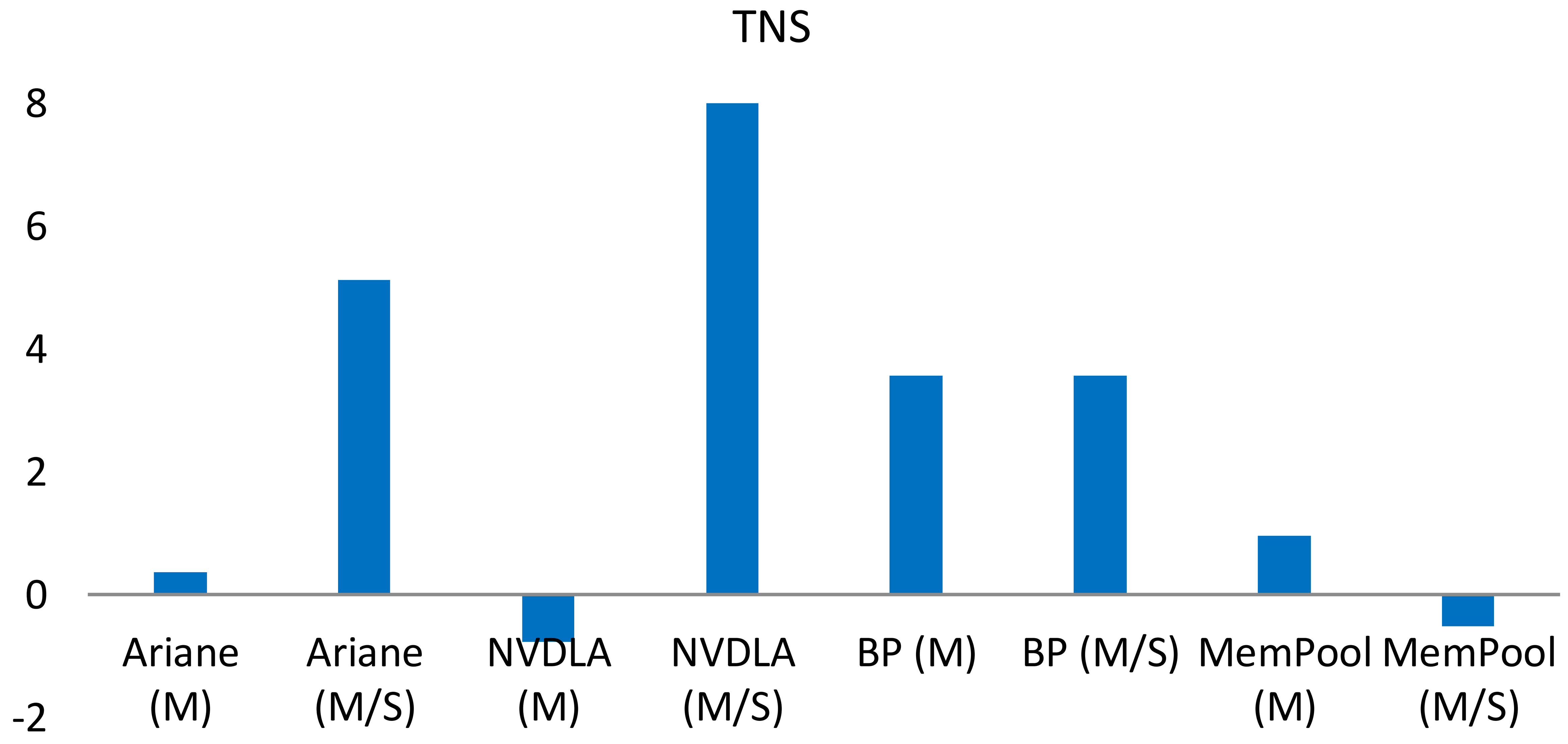
Post-Route Results

Design (M = macros, M/S = macros & standard cells)









Conclusion & Future Work

- Open-source AutoDMP **enhances GPU-accelerated DREAMPlace** for mixed-size placement
  - Fix macro legalization issues
  - Expand design space with additional parameters

- AutoDMP **explores placement space efficiently**
  - High-level PPA proxies
  - Navigate the Pareto front with multi-objective Bayesian optimization
  - Two-level PPA evaluation

- AutoDMP obtains **competitive PPA** results on TILOS benchmark

- Future work
  - More complex industrial-size designs
  - Better PPA proxies
  - Beyond PPA (e.g., chiplet floorplan exploration)