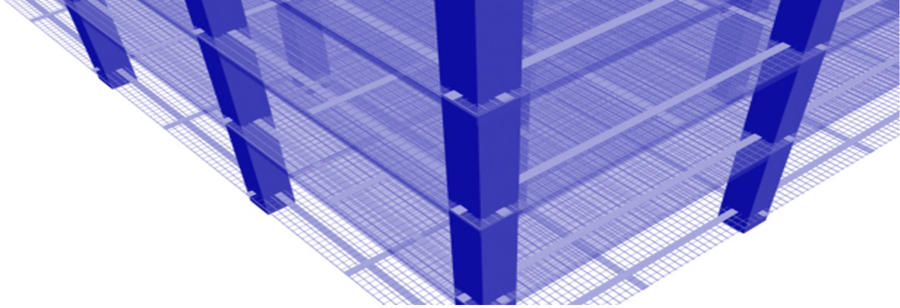
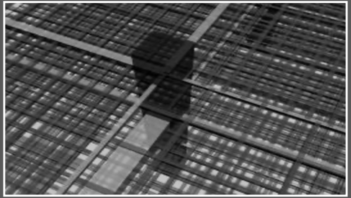
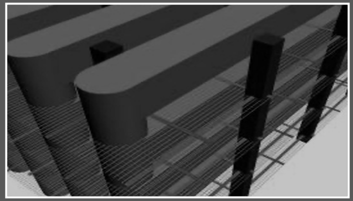
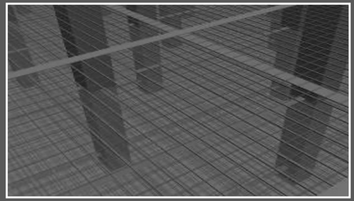
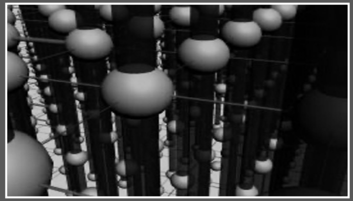


DREAM-GAN: Advancing DREAMPlace towards Commercial-Quality using Generative Adversarial Learning









Yi-Chen Lu[1], Haoxing Ren[2], Hao-Hsiang Hsiao[1], and Sung Kyu Lim[1]

Georgia Institute of Technology[1]

Nvidia[2]

Presentation Outline

- **Motivations**
  - Importance of DREAMPlace and its current limitations
- **Generative Adversarial Learning**
  - Introduction and why using GAN to improve DREAMPlace
- **DREAM-GAN Overview**
  - DREAMPlace as a generator to generate tool-alike placements
- **Detailed Architectures**
  - GNN-based and CNN-based discriminators
  - Soft-Bin Transformation
- **Experimental results**
  - Full-flow head-to-head comparisons using Synopsys ICC2
- **Discussion**
- **Conclusion and Future Work**

Motivation

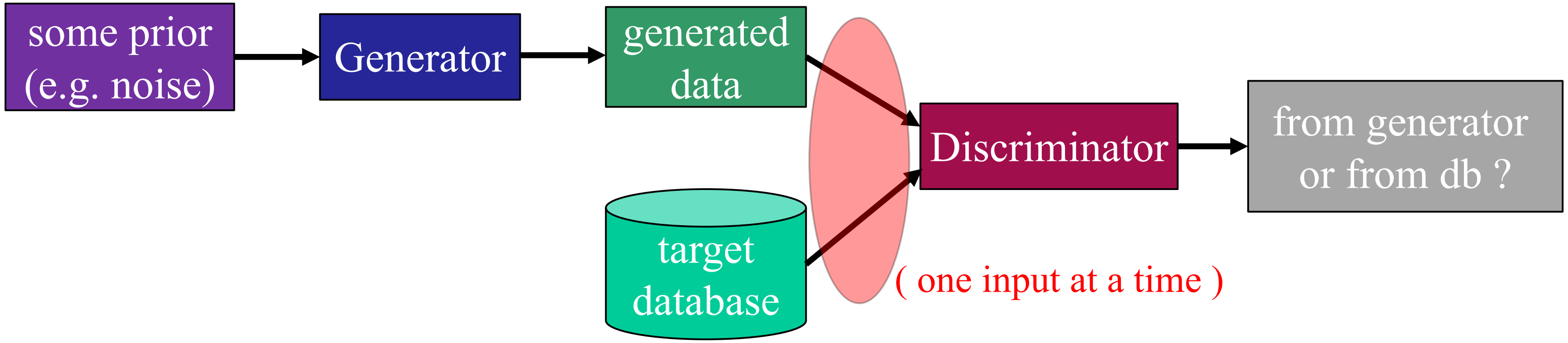
- DREAMPlace (DP) significantly boosts chip design productivity
  - perform placement from hours to minutes
  - originates from RePlace, the state-of-the-art academic placer
    - cpu-intensive objectives are implemented in CUDA and accelerated by PyTorch
  - but, solution quality is yet comparable to commercial tools'

- Commercial tools adopt multi-objective placement optimization
  - e.g., timing, power, routability …
  - vanilla DP solely focuses on wirelength and density
  - same netlist, but very "different" placement
    - in terms of "cell locations" → can be visualized using "density maps"

- How can we use existing tool placements to improve DP?
  - generative adversarial learning to close the "difference gap"

Generative Adversarial Learning

- Introduction of Generative Adversarial Networks (GANs)
  - Generator goal:
    - To generate *meaningful* distributions from non-meaningful inputs
  - Discriminator goal:
    - To find out the true origin of its inputs



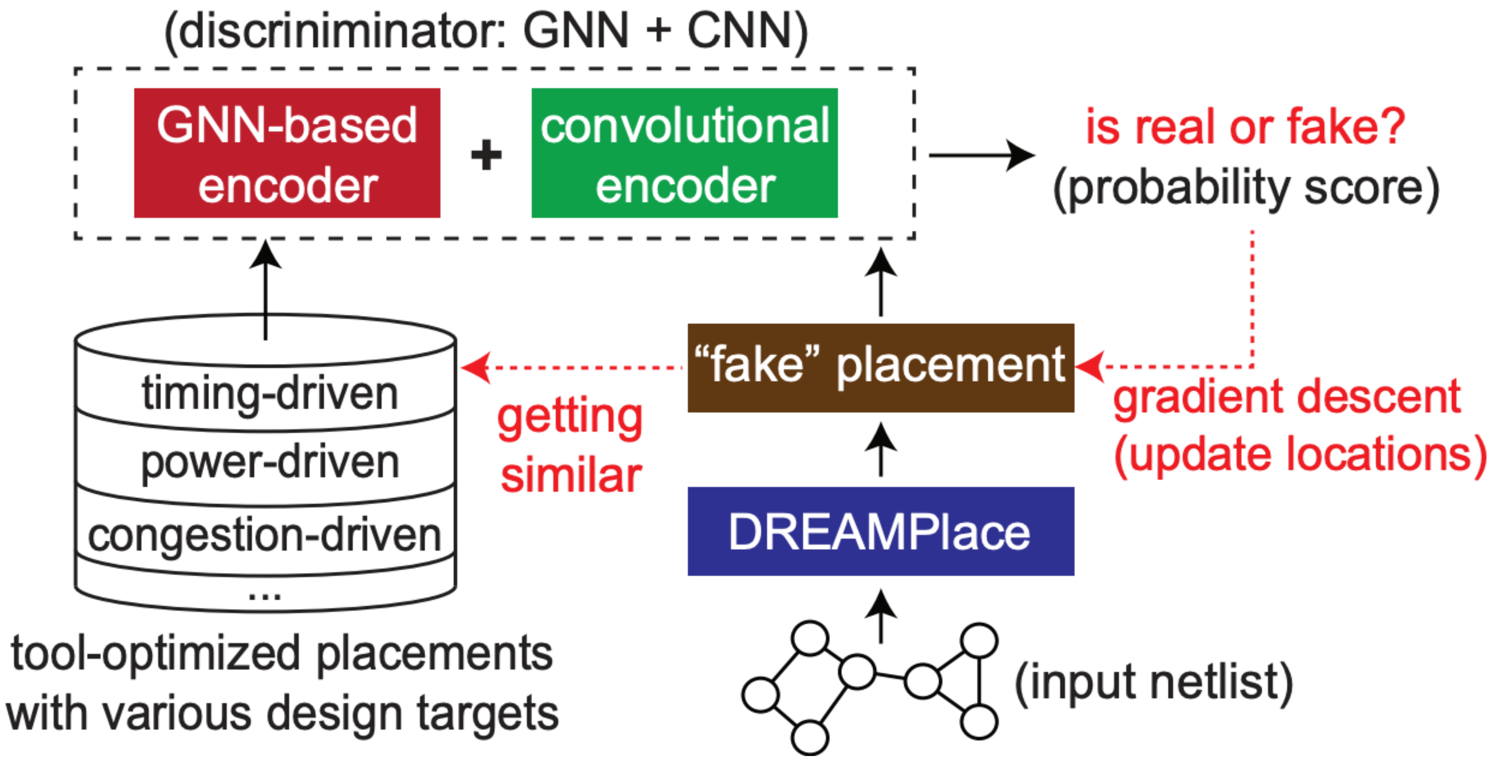
- DREAMPlace can be naturally considered as a "generator"
- we build "discriminator" to differentiate placement origins

Why using GAN to Improve DP?

- Generalizability
  - GAN focuses on parameterizing target distributions
    - Not memorizing!→ no net matching, cell alignment… etc.
  - DREAM-GAN does not require designs to be “exact” in the database
    - Number of cells/nets can be different between different placements
    - Ideally, can work between different designs (i.e., transfer learning)
      - under investigation (future work)

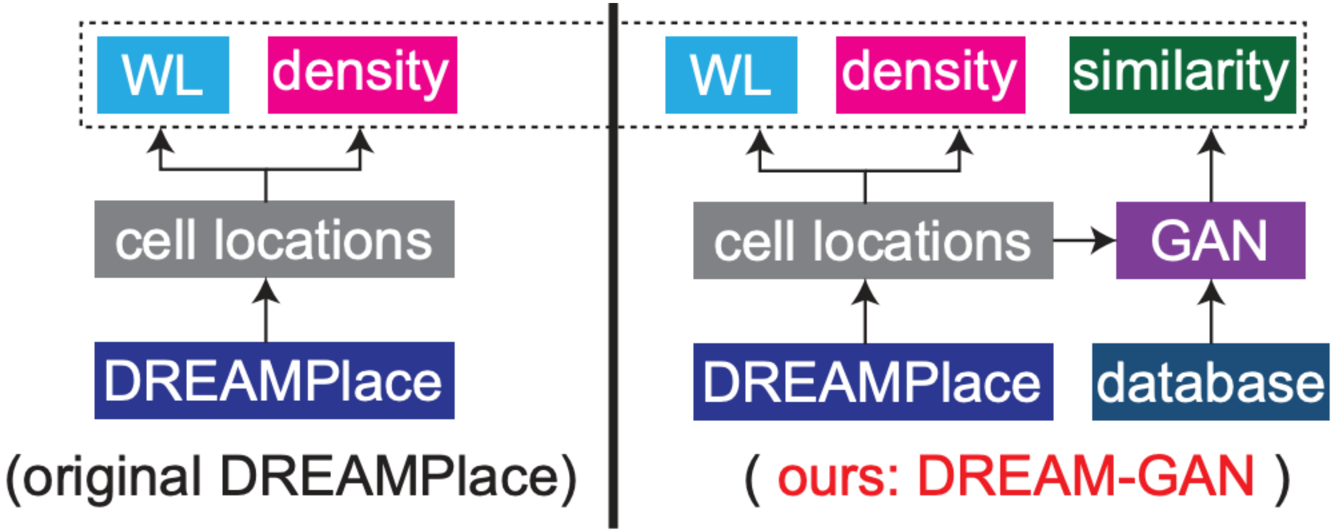
- Flexibility in Objectives
  - GAN can consume multiple “signals” at the same time
    - E.g., DREAM-GAN uses “netlist connectivity” and “bin-density map”
  - Optimization without needs of exact objective formulations
    - Exact, differentiable PPA objectives are hard to define
    - and often require huge effort to be GPU-acceleratable

DREAM-GAN Overview



- **Quantify placement similarity by:**
  - **Graph connectivity → using Graph Neural Networks (GNNs)**
  - **cell-density map → using Convolutional Neural Networks (CNNs)**

Objective Difference: DP vs. DREAM-GAN

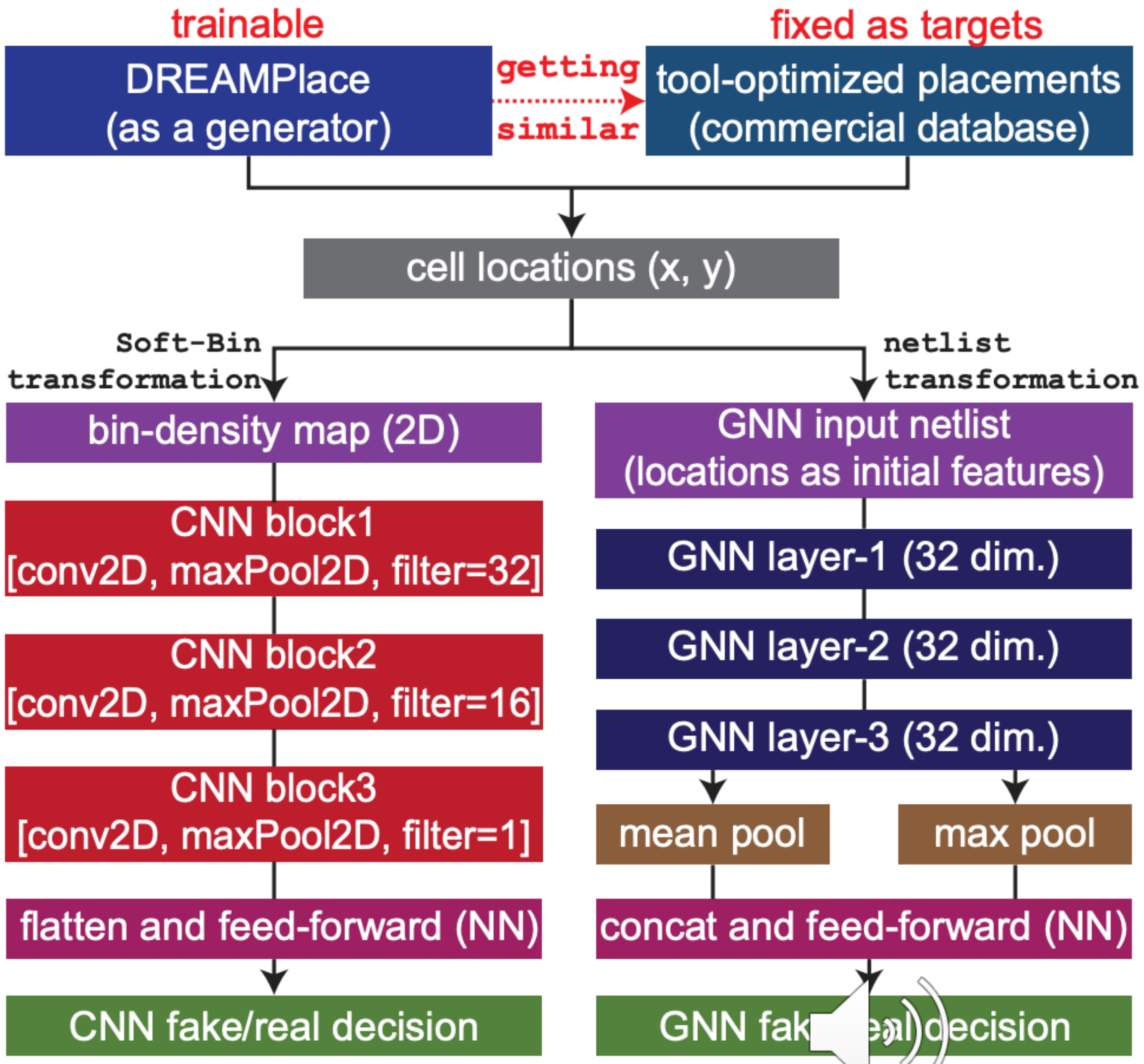


loss at each DP placement iteration

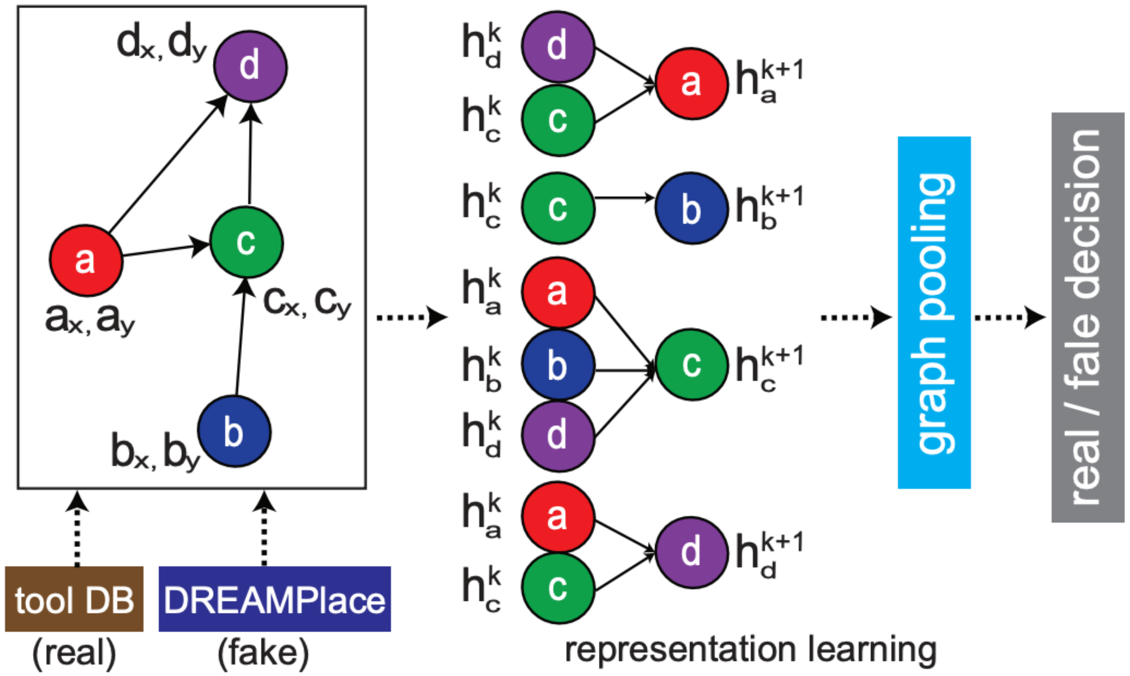
- **WL and Density denotes HPWL and Overflow metrics**
- **DREAM-GAN adds a differentiable similarity loss upon vanilla DP**
  - **Determined by GNN- and CNN-based discriminators**
  - **Added after initial 200 iterations**

DREAM-GAN Architecture

- Two discriminators:
  - GNN to encode connectivity
  - CNN to encode bin-density map
- Discriminators' outputs are differentiable w.r.t cell locations
- Optimizing fake/real decisions directly impacts (x,y) locations
- Key proposed algorithm:
  - Soft-Bin Transformation → generate differentiable density maps from locations

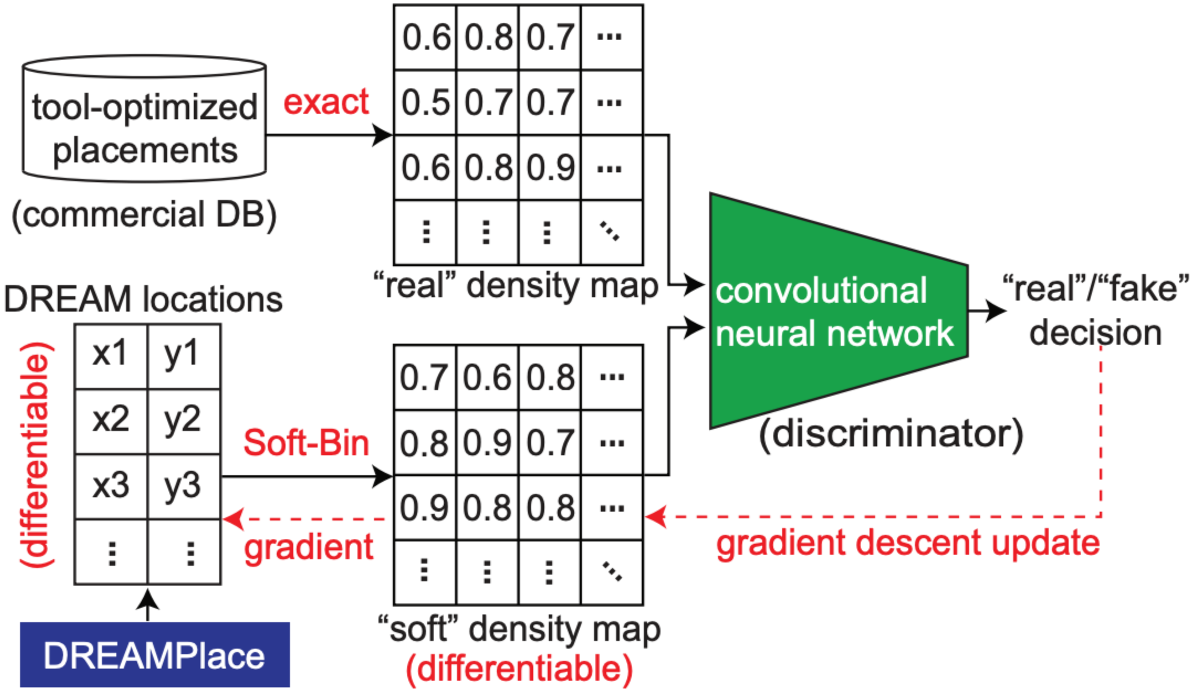


GNN-Based Discriminator



- Follow netlist transformation proposed in [16]
  - Only preserve driver-to-load connections of original hyperedges and introduce skip connections
- Follow GraphSAGE [6] to perform node representation learning
  - cell locations as initial node features
- Perform [mean, max] pooling to obtain graph-level representations

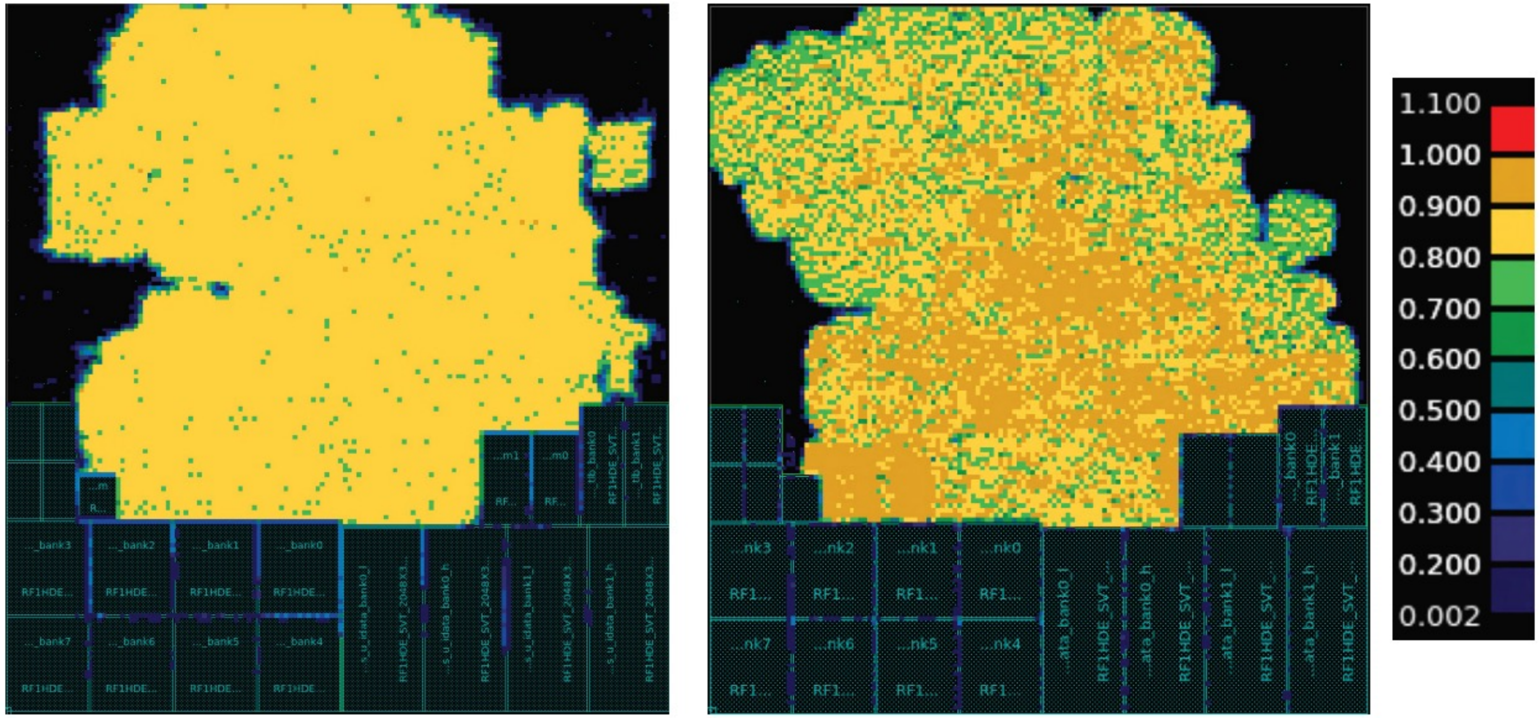
CNN-Based Discriminator



- **Goal: discriminate different placements from bin-density maps**
- **challenge:**
  - **Naïve bin-density map calculation (exact) is not differentiable w.r.t. locations**
    →minimizing/maximizing density will not impact locations
  - **Propose Soft-Bin, a differentiable density map transformation, to solve the issue**

Justification of Bin-Density Map

(same global density target at 0.85)



DREAMPlace　　　　commercial tool

- **Observation:**
  - **Commercial tool has extra intelligence in locally aggregating/loosening cells to improve PPA <u>(while satisfying global density constraints)</u>**

Soft-Bin Transformation

X_1	Y_1
X_i	Y_i
⋮	⋮
X_N	Y_N

(cell locations)

soft-bin transformation →

p1	p2	p3	
p4	**p0**	p5	
p6	p7	p8	

(differentiable density map)

$$p_i = \frac{e^{1/d_i}}{\sum_i e^{1/d_i}}$$

1. Assume cell_i = [x_i, y_i] belongs to **p0** by bin-definition

2. including neighboring bins [p1…p8], calculate distance to bin centers

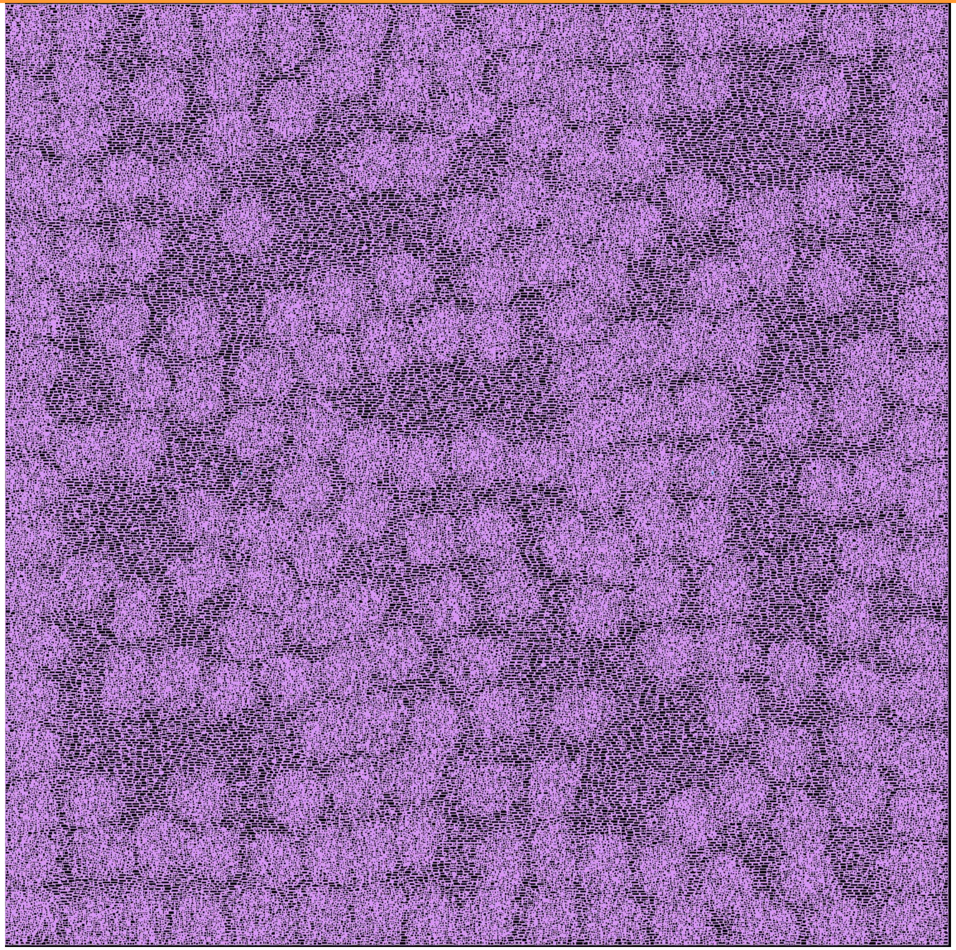
3. we obtain distance vector [d0, …, d8]

4. probability vector = softmax (1 / distance_vector)

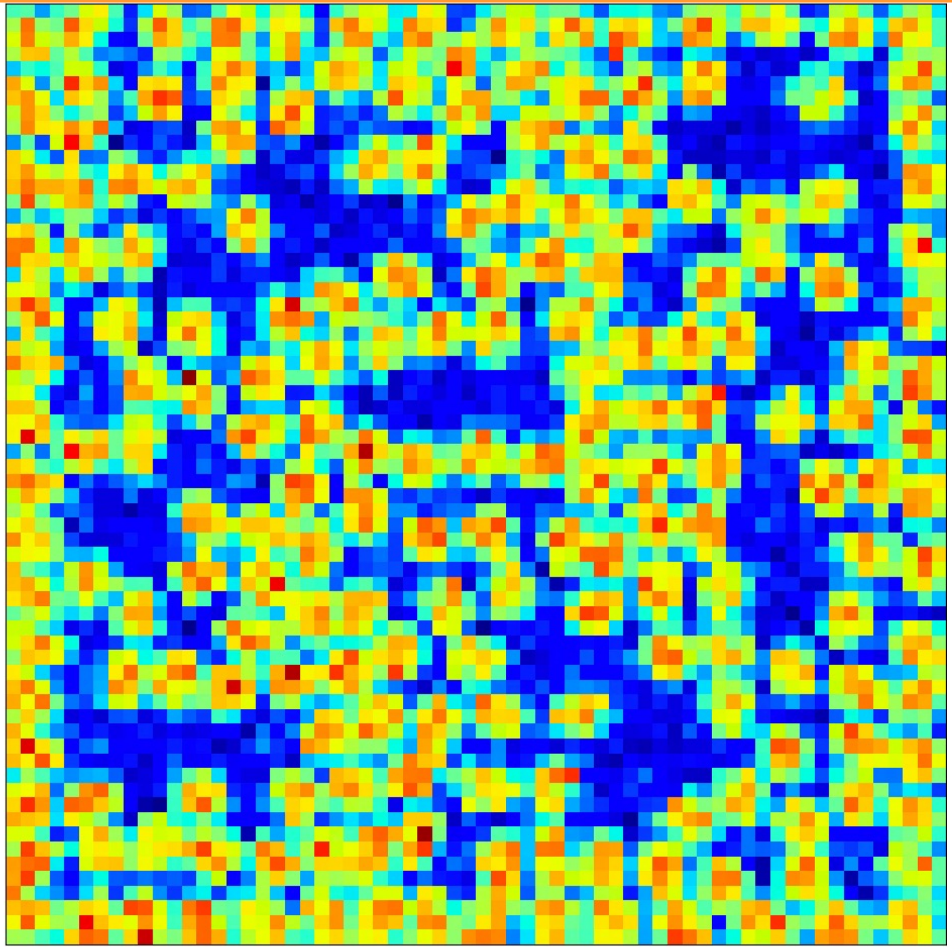
5. area contribution = prob * area_of_cell_i

6. now, gradient descent on achieved bin-density map will impact locations

Cell-Density Visualization on AES (0.85)

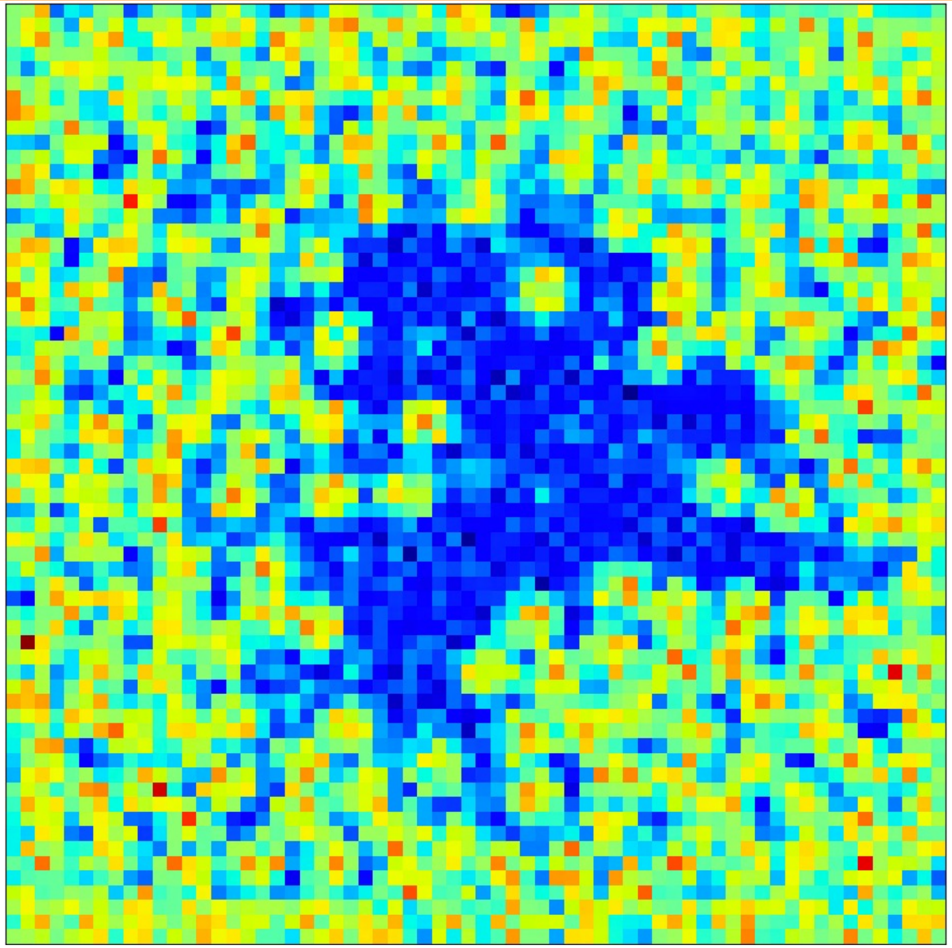


ICC2 default
(gui snapshot)

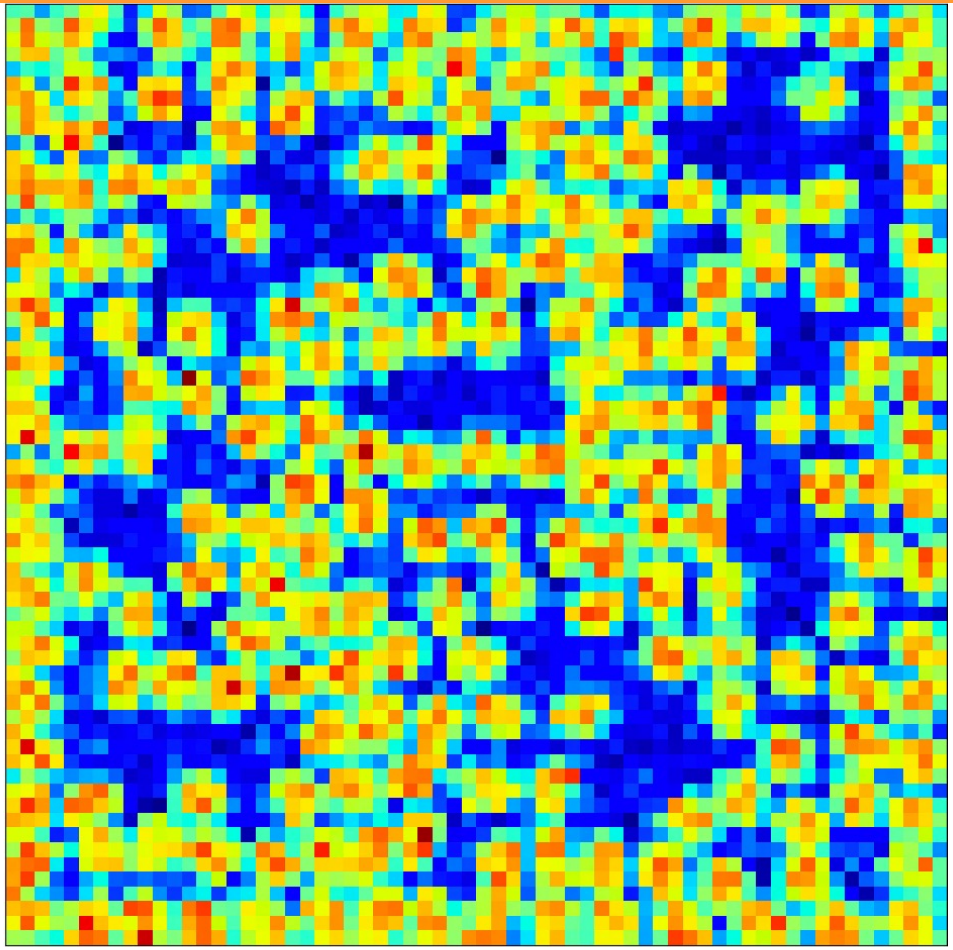


Soft-Bin Cell Density Map
(differentiable)

DreamPlace vs ICC2 on AES (0.85)



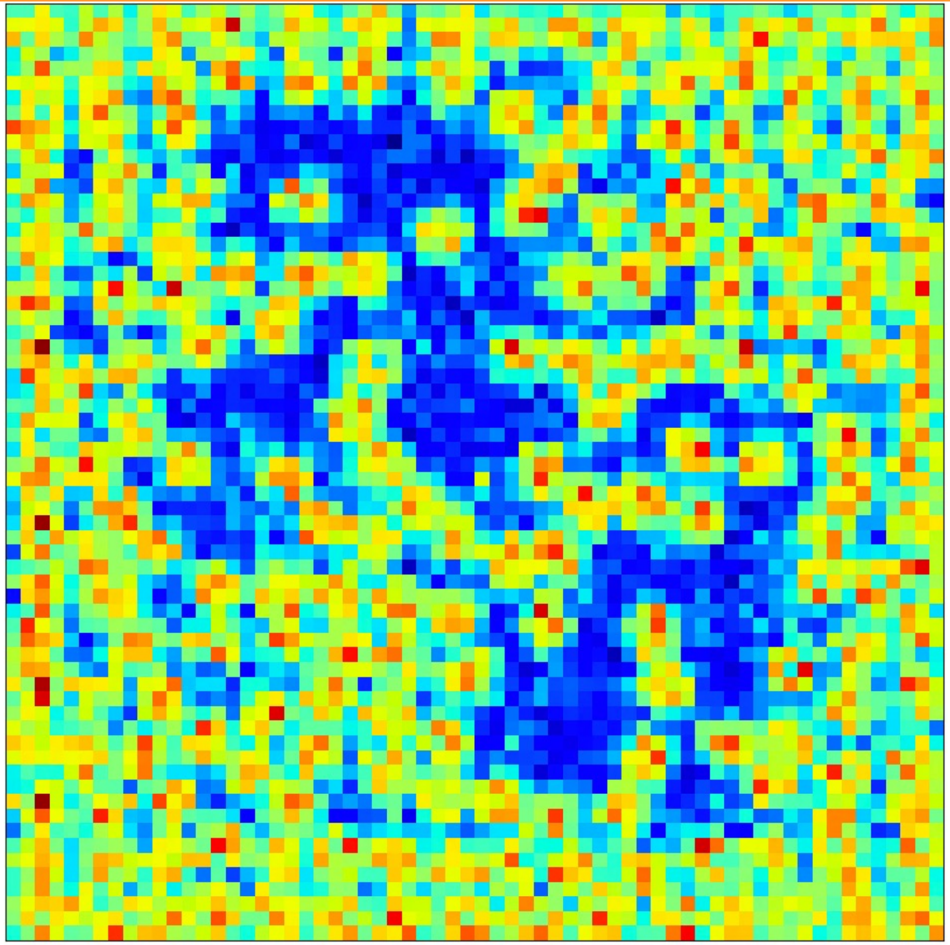
original DREAMPlace



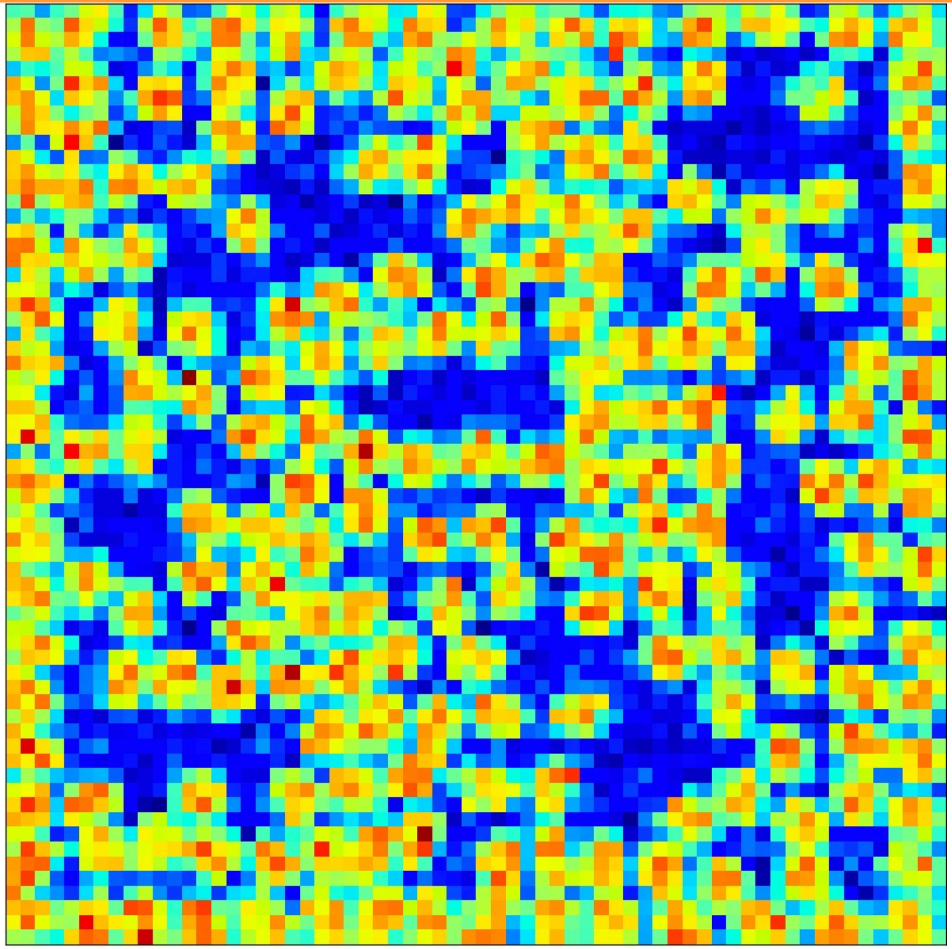
ICC2 default

- Observation: ICC2 has higher density discrepancy

DREAM-GAN vs ICC2 on AES (0.85)



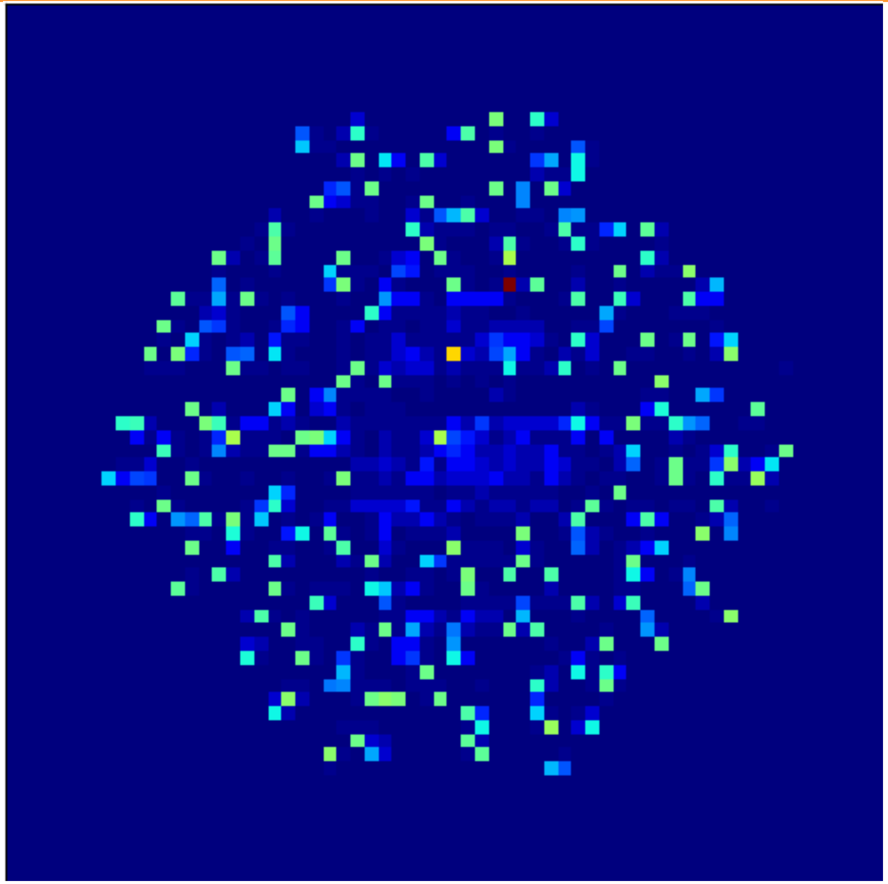
DREAM-GAN



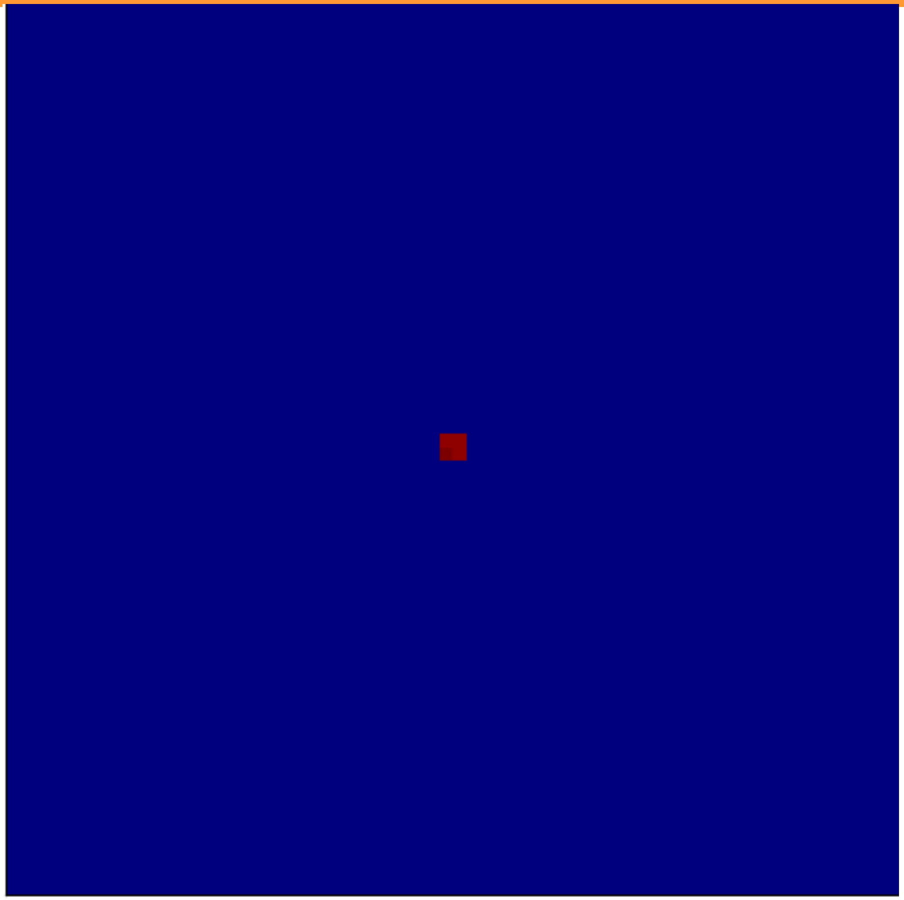
ICC2 default

- Arguably more similar with DREAM-GAN

Animation on AES (0.85): DP vs DREAM-GAN



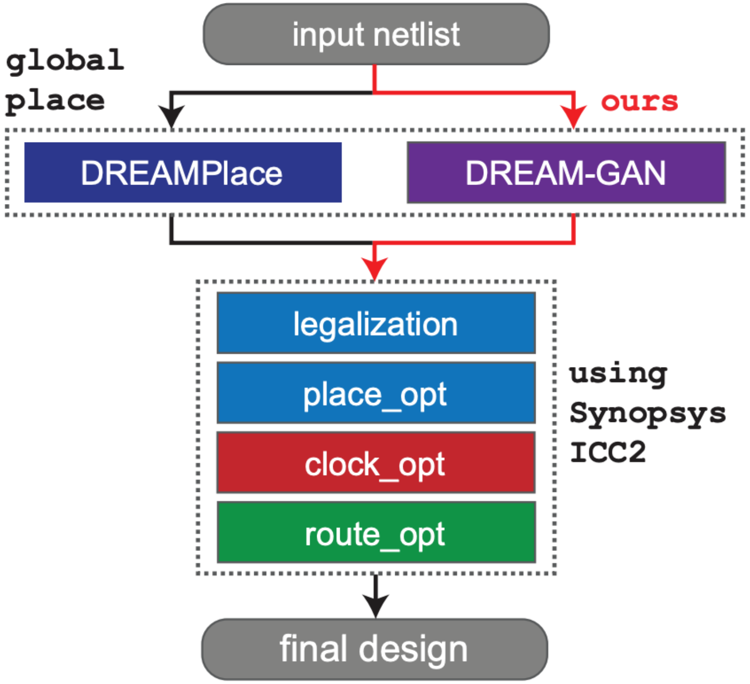
DREAMPlace
WL: 1946366 um, TNS: -183.14 ns,
#vio: 3258, tot. power: 607.8 uW



DREAM-GAN
WL: 1755417 um, TNS: -137.29 ns,
vio: 2944, tot. power: 601.9 W

Experimental Setup

- We compare DP and DREAM-GAN using a commercial PD flow
  - Implemented by Synopsys ICC2
  - For each design, we perform sweeping to generate 50 tool-optimized placements
- We only use DP / DREAM-GAN to perform global placement (without legalization)
  - Macros (if any) will be prefixed and non-touched during global place



ICC2 parameters	type (values)	description
set_qor_strategy	enum (3)	set optimization priority
low_power_effort	enum (4)	effort in low power optimization
congestion_effort	enum (3)	effort in congestion optimization
is_timing_driven	bool (2)	is timing-driven placement
is_power_driven	bool (2)	is power-driven placement
buffer_aware	bool (2)	buffering of high-fanout nets
coarse_density	float ([0.7,0.9])	density of global placement
target_route_density	float ([0.7,0.9])	density of early global routing

parameter sweeping to generate DB for DREAM-GAN

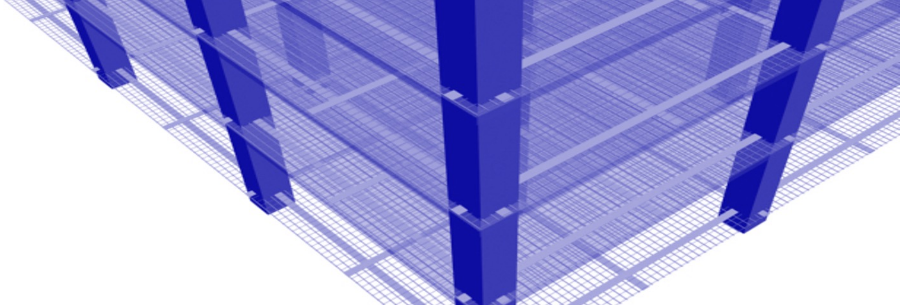
Optimization Results

design (# cells)	PD stage	DREAMPlace [11] wns (ns)	DREAMPlace [11] TNS (ns)	DREAMPlace [11] # vios	DREAMPlace [11] total WL (um)	DREAMPlace [11] total power (mW)	DREAM-GAN (ours) wns (ns)	DREAM-GAN (ours) TNS (ns)	DREAM-GAN (ours) # vios	DREAM-GAN (ours) total WL (um)	DREAM-GAN (ours) total power (mW)
CPU-1 (220K)	global place	-2.05	-13498	19558	374130	200.1	-1.46	-10601	18425	3546577	193.5
	place opt	-1.74	-6197	13018	4034908	194.7	-1.52	-6024	12697	3870333	179.6
	clock opt	-0.30	-45.89	681	4163129	144.4	-0.24	-34.28	473	4041709	140.1
	route opt	-0.26	-22.4	464	4166459	144.3	-0.18	-21.11	446	4050908 (-2.7%)	141.9 (-1.6%)
CPU-2 (580K)	global place	-432.97	-5634543	48869	12382802	25142.4	-432.98	-5324323	45644	11110278	25098.2
	place opt	-608.91	-7218793	40780	12654907	13244.1	-608.74	-7202230	40544	11493278	12431.0
	clock opt	-0.20	-61.48	1726	17769476	488.1	-0.23	-48.28	1505	16305060	455.0
	route opt	-0.17	-45.83	1405	17765081	490.5	-0.14	-28.61	942	16287654 (-8.3%)	454.2 (-7.4%)
CPU-3 (121K)	global place	-2.13	-8437.48	11730	1711937	149.2	-1.96	-8057.19	11435	1691131	147.8
	place opt	-0.54	-164.78	2466	1439469	155.8	-0.48	-138.74	1981	1413154	153.1
	clock opt	-0.51	-37.68	414	1588135	141.9	-0.57	-32.98	359	1518498	137.7
	route opt	-0.49	-41.21	1207	1582822	143.0	-0.35	-36.24	1023	1520481 (-3.9%)	138.9 (-2.9%)
VGA (57K)	global place	-2.2	-13999.49	16630	2418386	345.5	-1.65	-8057.19	11435	1691131	342.2
	place opt	-0.07	-2.06	188	1426981	279.8	-0.10	-2.55	171	1456516	276.5
	clock opt	-0.16	-7.46	441	1579559	327.8	-0.14	-5.37	398	1536218	322.4
	route opt	-0.17	-13.73	1712	1586940	333.5	-0.14	-7.06	1050	1542569 (-2.8%)	329.1 (-1.3%)
LDPC (46K)	global place	-1.14	-1411.74	2184	1289738	225.8	-1.10	-1331.30	2048	1233014	219.5
	place opt	-0.25	-292.49	2192	1454863	255.5	-0.21	-217.76	-217.76	1390693	248.6
	clock opt	-0.20	-156.62	1897	1857624	255.4	-0.16	-98.47	1757	1785355	248.4
	route opt	-0.24	-198.72	1976	1878969	261.8	-0.18	-123.94	1846	1803729 (-4.0%)	255.0 (-2.6%)

- all metrics are reported using ICC2 (including global place stage)
- improvements last firmly to the post-route stage

Conclusion and Future Work

- We present DREAM-GAN
  - Optimize DP solution quality using generative adversarial learning
- We show that tool’s and DP’s placements are inherently different
  - obvious difference in cell locations
  - salient difference bin-density maps
- We believe GAN provides a promising way to perform optimization
  - optimization without knowing blackboxed algorithms or constrains
- In the future, we aim to explore transfer learning across designs



Thank You for Listening!   Q&A