59 DESIGN AUTOMATION CONFERENCE

JULY 10 - 14, 2022

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

Xplace: An Extremely Fast and Extensible Global Placement Framework

Lixin Liu, Bangqi Fu, Martin D.F. Wong, Evangeline F.Y. Young

CSE Department

The Chinese University of Hong Kong

Global Placement Problem

A fundamental step in VLSI physical design

- Highly affect the circuit’s PPA

Modern circuits contain millions of standard cells

- Highly increase the computational complexity of GP
- Bring huge challenges to the leading-edge global placers

Global Placement Problem

Objective:

- Minimize the total HPWL of all the nets
- Satisfy the cell density constraint

$$\min_{p} HPWL(p) = \min_{p} \sum_{e \in E} HPWL_e(p)$$

$$\text{s.t.} \quad D_b \leq D_t, \forall b \in B$$

Analytical Global Placement:

- A smooth approximation of HPWL
- A density penalty

$$\min_{p} \sum_{e \in E} WL_e(p) + \lambda D(p)$$

[1] J. Lu, H. Zhuang, P. Chen, H. Chang, C.-C. Chang, Y.-C. Wong, L. Sha, D. Huang, Y. Luo, C.-C. Teng, et al., "ePlace-MS: Electrostatics-based placement for mixed-size circuits," IEEE TCAD 2015

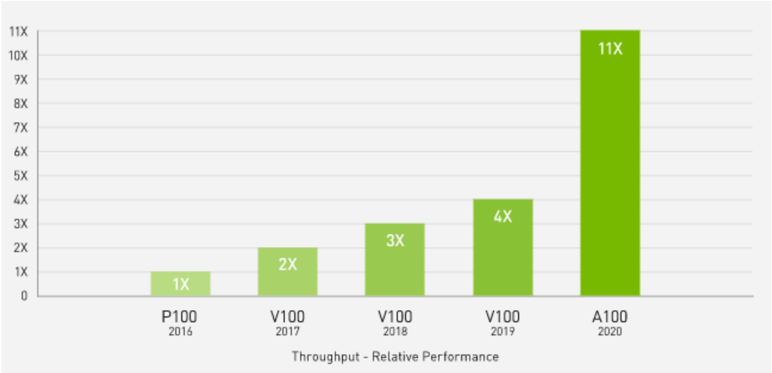
GPU-accelerated Global Placers

- Rapid development of GPU's computational power
- GPU acceleration becomes an important direction

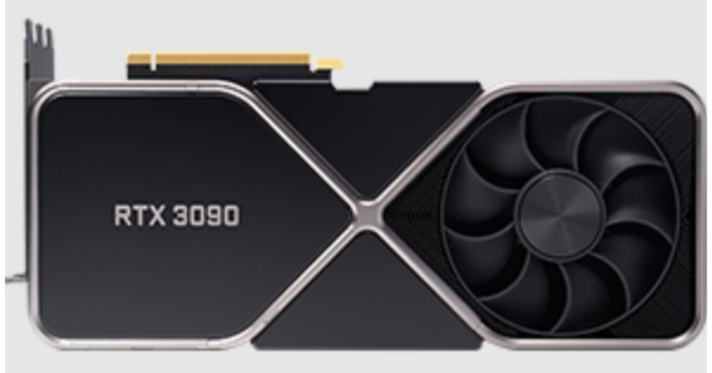
Recently, DREAMPlace[1]

- Implemented the approach of ePlace[2] on GPU
- Produced the SOTA solution quality and performance

It is a big challenge to further improve on DREAMPlace's performance.



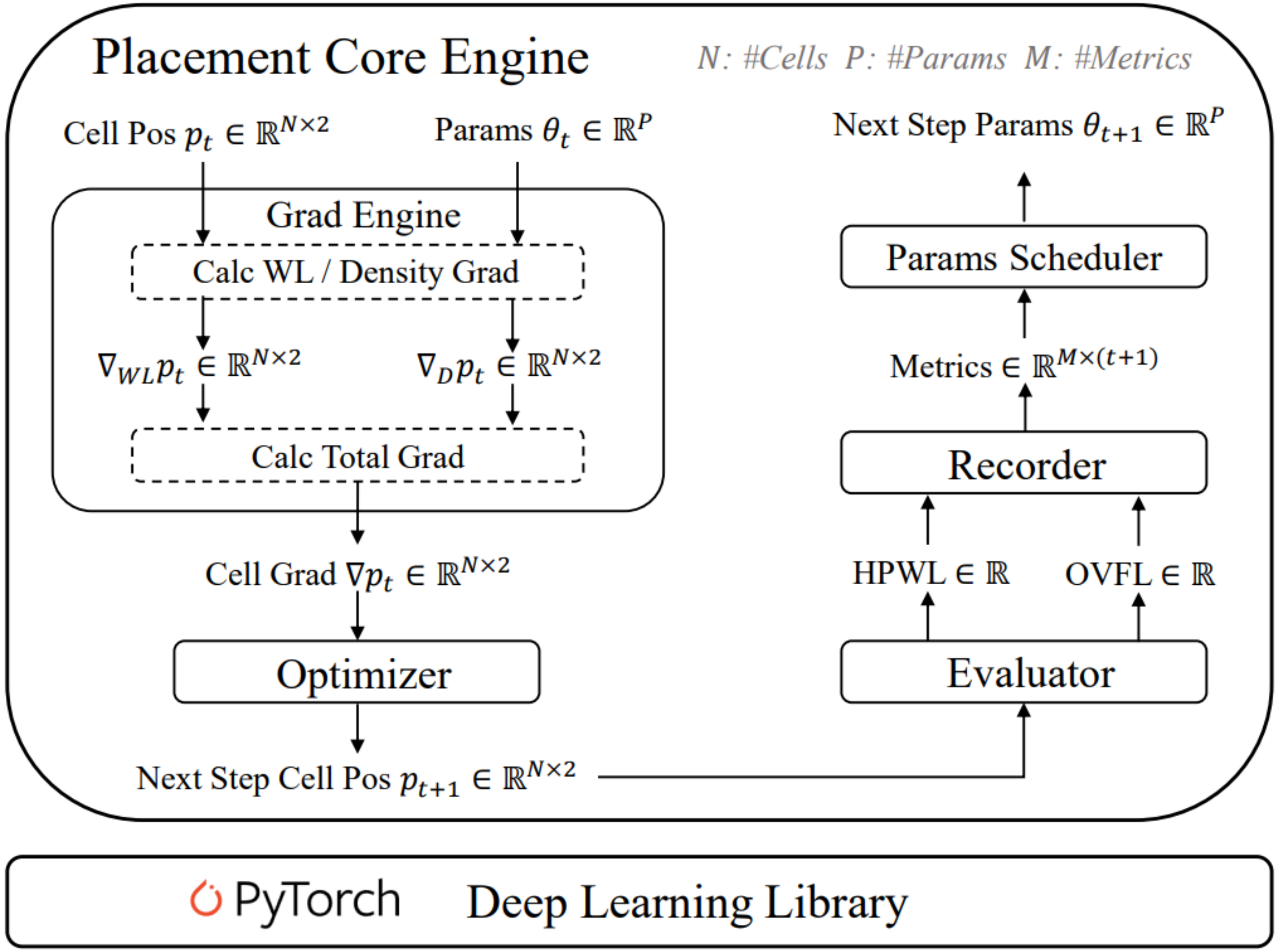
https://www.nvidia.com/en-in/data-center/a100/



https://assets.nvidia.partners/images/png/nvidia-geforce-rtx-3090.png

[1] Y. Lin, Z. Jiang, J. Gu, W. Li, S. Dhar, H. Ren, B. Khailany, and D. Z. Pan, "DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement," IEEE TCAD 2020
[2] J. Lu, H. Zhuang, P. Chen, H. Chang, C.-C. Chang, Y.-C. Wong, L. Sha, D. Huang, Y. Luo, C.-C. Teng, et al., "ePlace-MS: Electrostatics-based placement for mixed-size circuits," IEEE TCAD 2015

Proposed Framework: Xplace



Operator-Level Optimization

1. Wirelength Operator Combination (OC)

Observation: Both the HPWL function and the stable WA wirelength function need the min and max cell positions in a net.

Method: combining the three operators with heavy wirelength-related workload, ***WA wirelength***, ***WA gradient*** and ***HPWL***, into one operator

Result: avoid redundant computation of the min and max function

$$HPWL_e(p) = \left(\boxed{\max_{i\in e} x_i} - \boxed{\min_{i\in e} x_i}\right) + \left(\boxed{\max_{i\in e} y_i} - \boxed{\min_{i\in e} y_i}\right)$$

HPWL function

$$WL_e(x) = \frac{\sum_{i\in e} x_i e^{\frac{x_i - \boxed{\max_{j\in e} x_j}}{\gamma}}}{\sum_{i\in e} e^{\frac{x_i - \boxed{\max_{j\in e} x_j}}{\gamma}}} - \frac{\sum_{i\in e} x_i e^{\frac{\boxed{\min_{j\in e} x_j} - x_i}{\gamma}}}{\sum_{i\in e} e^{\frac{\boxed{\min_{j\in e} x_j} - x_i}{\gamma}}}$$

$$WL_e(y) = \frac{\sum_{i\in e} y_i e^{\frac{y_i - \boxed{\max_{j\in e} y_j}}{\gamma}}}{\sum_{i\in e} e^{\frac{y_i - \boxed{\max_{j\in e} y_j}}{\gamma}}} - \frac{\sum_{i\in e} y_i e^{\frac{\boxed{\min_{j\in e} y_j} - y_i}{\gamma}}}{\sum_{i\in e} e^{\frac{\boxed{\min_{j\in e} y_j} - y_i}{\gamma}}}$$

Stable WA wirelength

Operator-Level Optimization

2. Density Operator Extraction (OE)

Overflow ratio and density computation:

$$OVFL = \frac{\sum_{b \in B} \max(D_b - D_t, 0) A_b}{\sum_{i \in V_{mov}} A_i} \qquad D_b = \frac{\sum_{i \in V} A_i \cap A_b}{A_b}, \ \forall b \in B$$

D_t: target density, D_b: bin b's cell density, A_b and A_t denote the area for bin b and cell i,

Operator-Level Optimization

2. Density Operator Extraction (OE)

Overflow ratio and density computation:

$$OVFL = \frac{\sum_{b \in B} \max(D_b - D_t, 0) A_b}{\sum_{i \in V_{mov}} A_i} \qquad D_b = \frac{\sum_{i \in V} A_i \cap A_b}{A_b}, \ \forall b \in B$$

D_t: target density, D_b: bin b's cell density, A_b and A_t denote the area for bin b and cell i,

Need to insert filler cells inside the electrostatic system [1]

$D_{fl,b}$: Bin b's filler density

$$\tilde{D}_b = \frac{\sum_{i \in V \cup V_{fl}} A_i \cap A_b}{A_b} = D_b + \boxed{\frac{\sum_{i \in V_{fl}} A_i \cap A_b}{A_b}}, \ \forall b \in B$$

V: the set of cells, V_{fl}: the set of fillers, $\widetilde{D_b}$: bin b's total density (incl. filler density)

[1] J. Lu, H. Zhuang, P. Chen, H. Chang, C.-C. Chang, Y.-C. Wong, L. Sha, D. Huang, Y. Luo, C.-C. Teng, et al., “ePlace-MS: Electrostatics-based placement for mixed-size circuits,” IEEE TCAD 2015

Operator-Level Optimization

2. Density Operator Extraction (OE)

Overflow ratio and density computation:

$$OVFL = \frac{\sum_{b \in B} \max(D_b - D_t, 0) A_b}{\sum_{i \in V_{mov}} A_i} \qquad D_b = \frac{\sum_{i \in V} A_i \cap A_b}{A_b}, \ \forall b \in B$$

D_t: target density, D_b: bin b's cell density, A_b and A_t denote the area for bin b and cell i,

Need to insert filler cells inside the electrostatic system [1]

$$\tilde{D} = D + D_{fl}$$

Matrix form of the total density map. $\tilde{D}, D, D_{fl} \in \mathbb{R}^{M \times M}$, M is the grid size

[1] J. Lu, H. Zhuang, P. Chen, H. Chang, C.-C. Chang, Y.-C. Wong, L. Sha, D. Huang, Y. Luo, C.-C. Teng, et al., "ePlace-MS: Electrostatics-based placement for mixed-size circuits," IEEE TCAD 2015

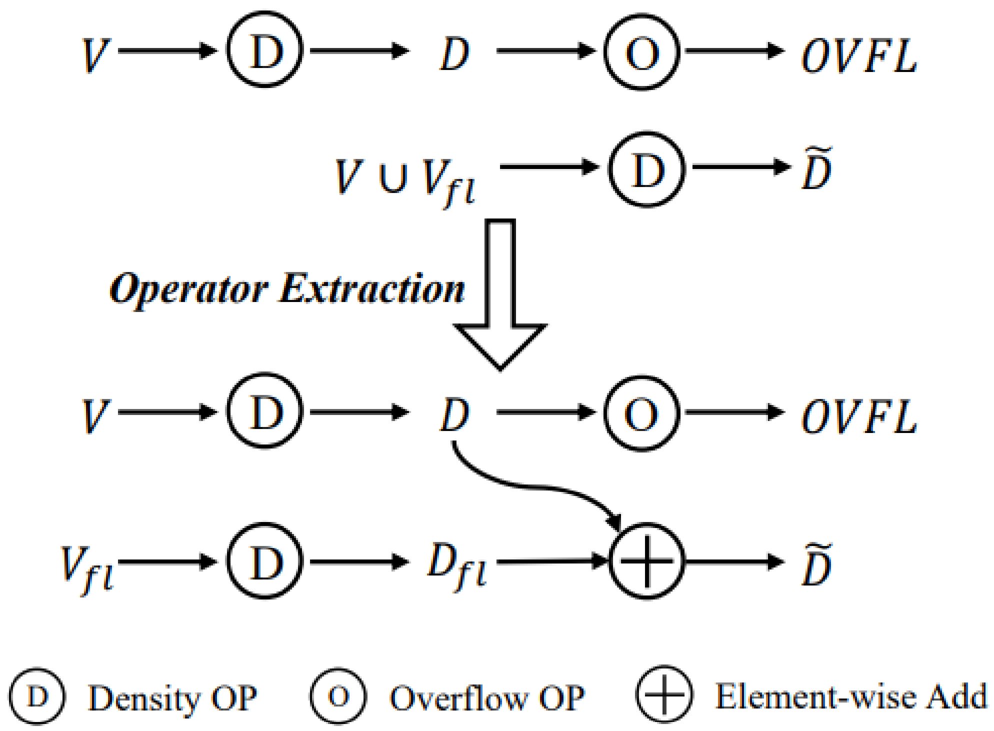
Operator-Level Optimization

2. Density Operator Extraction (OE)

$$OVFL = \frac{\sum_{b \in B} \max(D_b - D_t, 0) A_b}{\sum_{i \in V_{mov}} A_i} \qquad \tilde{D} = D + D_{fl}$$

Observation: Both the calculation of $OVFL$ and total density map $\tilde{D}$ need the cell density map D.

Method: common sub-operator D extraction, compute the cell density map D and the filler density map D_{fl} separately



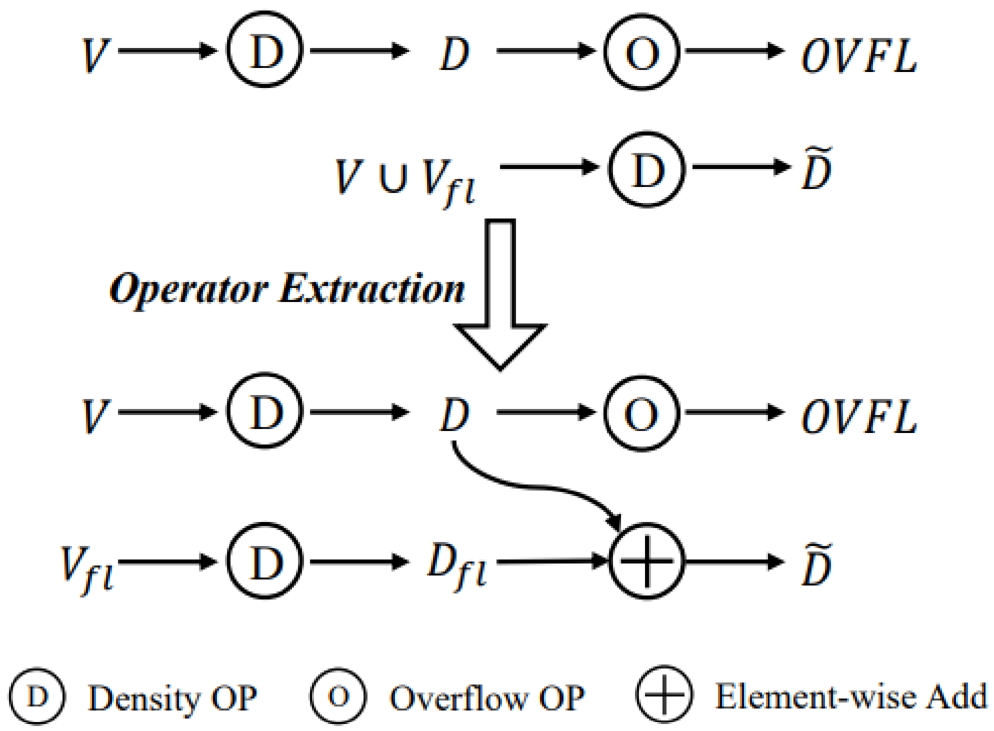
Operator-Level Optimization

2. Density Operator Extraction (OE)

$$OVFL = \frac{\sum_{b \in B} \max(D_b - D_t, 0) A_b}{\sum_{i \in V_{mov}} A_i} \qquad \tilde{D} = D + D_{fl}$$

Observation: Both the calculation of $OVFL$ and total density map $\tilde{D}$ need the cell density map D.

Result: reduce the total computation time of the cell density map D.



Operator-Level Optimization

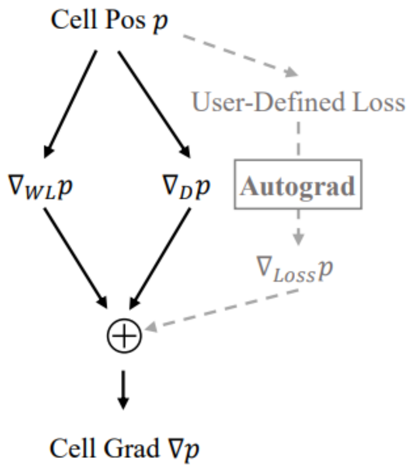
3. Operator Reduction (OR)

Observation:

- The number of forward operators are almost the same as that in the backward
- Invoking the heavy autograd engine will almost **double the number of operators** and bring **large kernel launching overhead on CPU**

Method:

- Avoid invoking the heavy **autograd engine**
- Directly derive the numerical solutions of the WL / density grad
- Assign a weighted accumulated gradient to each cell



Result: Reduce the total kernel launching time

Operator-Level Optimization

3. Operator Reduction (OR)

Other Methods:

- Use in-place operators as much as possible
    - Avoid redundant copying
- Reorder the operators that need sync to the end of the execution queue
    - Reduce the frequency of interrupting the GPU pipeline

Operator-Level Optimization

4. Operator Skipping (OS)

Observation:

- The ratio $r = \frac{\lambda|\nabla D_{x,y}|}{|\nabla WL_{x,y}|}$ is ultra-small in the early placement stage

Method:

- When $(r < 0.01) \wedge (iter < 100)$, the density grad operator will only be executed once per 20 iterations

Result:

- Skip some density grad calculation in early placement stage

Placement-Stage-Aware Parameters Scheduling

Precondition matrix of $\tilde{\mathrm{H}}^{-1} = (\tilde{\mathrm{H}_{\mathrm{W}}} + \lambda\tilde{\mathrm{H}_{\mathrm{D}}})^{-1}$ is applied to accelerate convergence [1]

$$\mathbf{H_W} = \mathrm{diag}(|S_1|, |S_2|, ..., |S_N|) \qquad \mathbf{H_D} = \mathrm{diag}(A_1, A_2, ..., A_N)$$

$|S_i|$: the number of nets connecting cell i, A_i the area of cell i

We introduce the precondition weighted ratio $\omega = \frac{\lambda|\mathbf{H_D}|}{|\mathbf{H_W}|+\lambda|\mathbf{H_D}|} \in [0,1]$ to measure the placement stage

[1] J. Lu, H. Zhuang, P. Chen, H. Chang, C.-C. Chang, Y.-C. Wong, L. Sha, D. Huang, Y. Luo, C.-C. Teng, et al., “ePlace-MS: Electrostatics-based placement for mixed-size circuits,” IEEE TCAD 2015

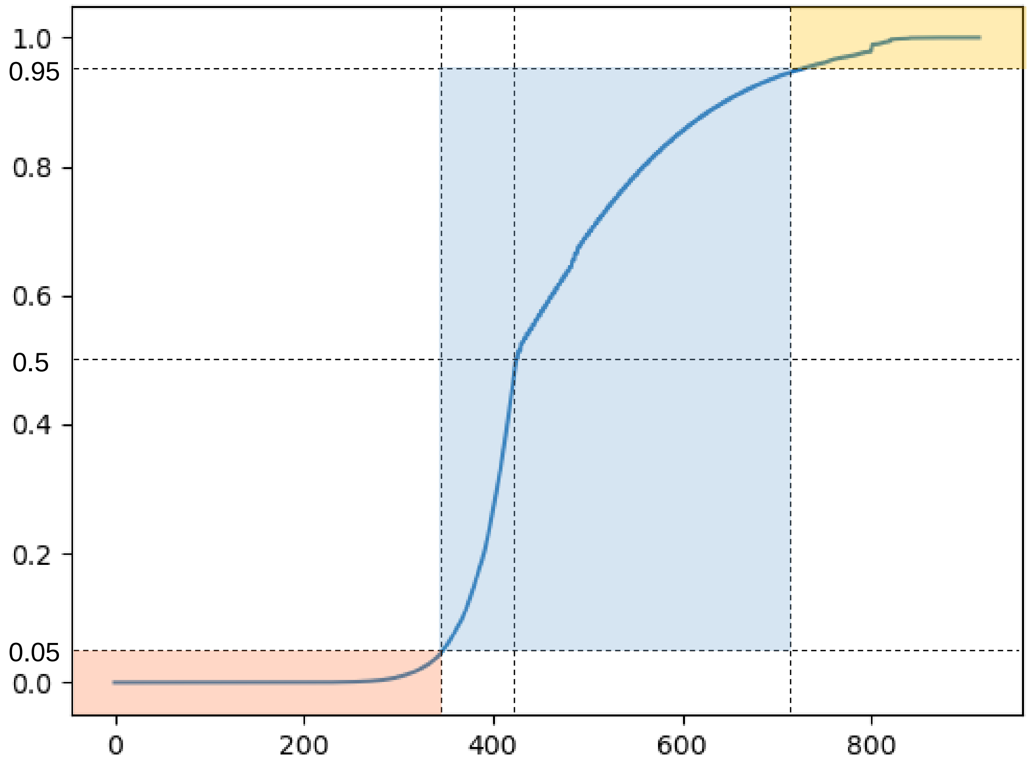
Placement-Stage-Aware Parameters Scheduling

Precondition weighted ratio $\omega = \frac{\lambda|\mathbf{H_D}|}{|\mathbf{H_W}|+\lambda|\mathbf{H_D}|} \in [0,1]$

$\omega > 0.95$ cells are forced to a final position with minimum local penalty

$0.05 < \omega < 0.95$ cells are spreading over the whole map and the overlap ratio significantly decreases

$\omega < 0.05$ wirelength-dominated and cells are driven to the position with minimum wirelength



ISPD 2005 / adaptec1

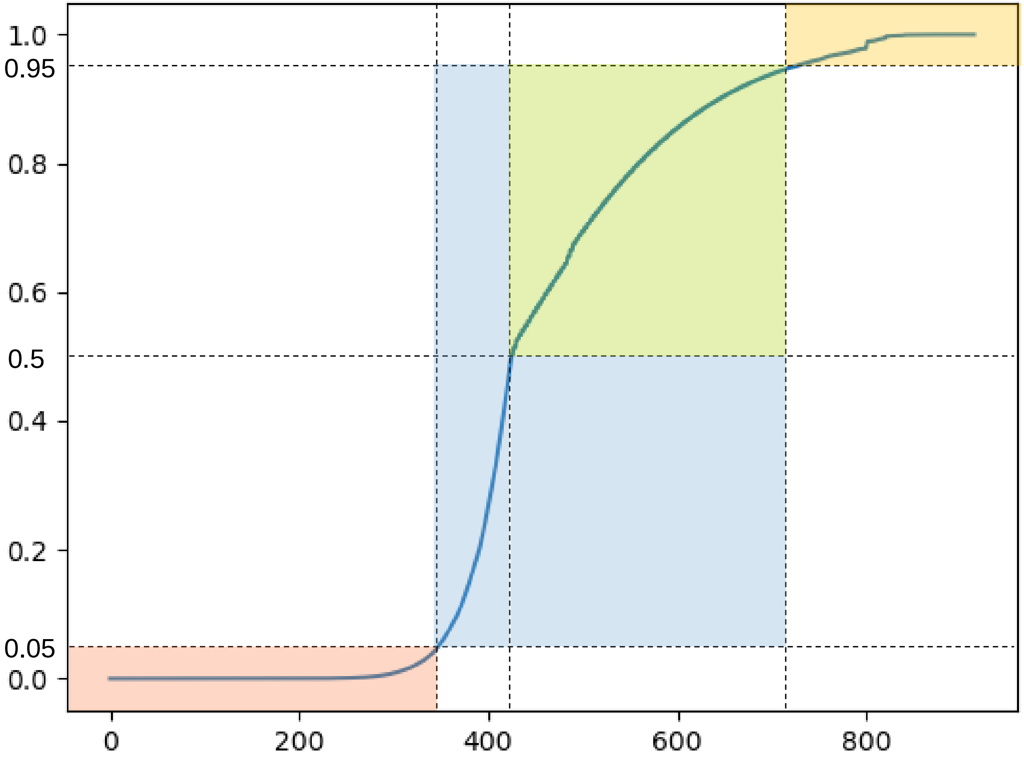
Placement-Stage-Aware Parameters Scheduling

Precondition weighted ratio $\omega = \frac{\lambda |\mathbf{H_D}|}{|\mathbf{H_W}| + \lambda |\mathbf{H_D}|} \in [0,1]$

To fully exploit the optimization space

Algorithm 1 Placement-Stage-Aware Parameters Scheduling

$\gamma \leftarrow \gamma_0$ ▷ wirelength coefficient
$\lambda \leftarrow \lambda_0$ ▷ density weight
while *iteration* < ITER and NOT Convergence **do**
  if $0.5 < \omega < 0.95$ and $iteration\%3 \neq 0$ **then**
    SKIP_UPDATE
  else
    $\gamma \leftarrow \gamma \times coef(overflow)$
    $\lambda \leftarrow \lambda \times \mu(\Delta hpwl)$ ▷ Both γ and λ are derived from [10]
  $\omega \leftarrow \frac{\lambda |\mathbf{H_D}|}{|\mathbf{H_W}| + \lambda |\mathbf{H_D}|}$



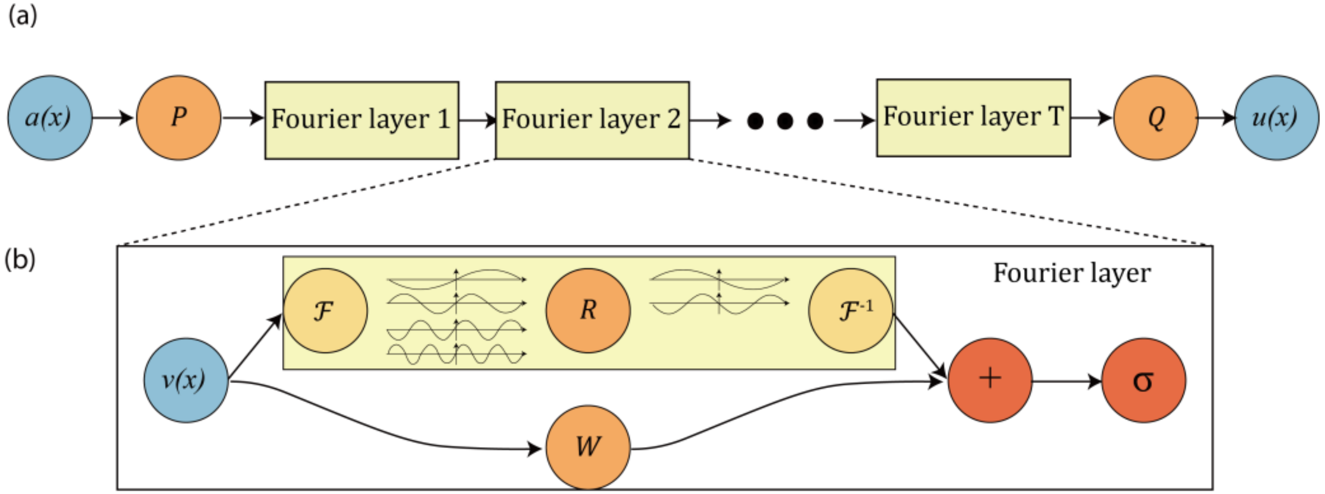
ISPD 2005 / adaptec1

DESIGN AUTOMATION CONFERENCE

Extending the Framework via Neural Enhancement

How to solve a 2D PDE problem by deep learning?

- ✗ Image-to-image networks -> Solve PDE in spatial domain conv
- ✓ 2D Fourier-Neural-Operator (FNO) [1] -> Solve PDE in frequency domain conv



The architecture of the fourier neural operators [1]

Many PDEs can be solved by Fourier transform.

[1] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, “Fourier neural operator for parametric partial differential equations,” in Proc. ICLR, 2021.

Extending the Framework via Neural Enhancement

How to solve a 2D PDE problem by deep learning?

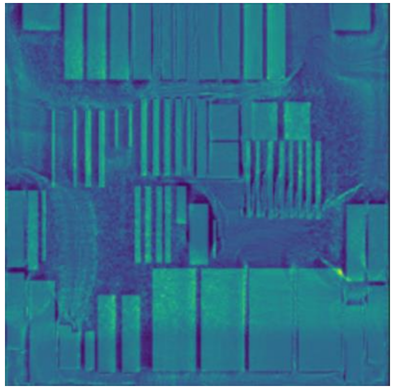
- ✗ Image-to-image networks -> Solve PDE in spatial domain conv
- ✓ 2D Fourier-Neural-Operator (FNO) [1] -> Solve PDE in frequency domain conv

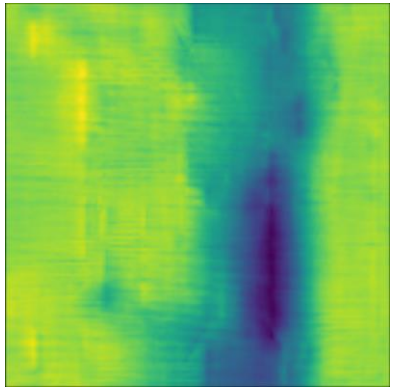
Poisson's Equation

$$\begin{cases} \nabla \cdot \nabla \psi(x,y) = -\rho(x,y), \\ \hat{\mathbf{n}} \cdot \nabla \psi(x,y) = 0, (x,y) \in \partial R, \\ \iint_R \rho(x,y) = \iint_R \psi(x,y) = 0, \end{cases}$$

Electron Distribution ρ -> 2D Density map D of placement

Electric Field $\nabla\psi_x, \nabla\psi_y$ -> moving force on x and y-axis


Density Map


Electric Field

Many PDEs can be solved by Fourier transform.

[1] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, “Fourier neural operator for parametric partial differential equations,” in Proc. ICLR, 2021.

Extending the Framework via Neural Enhancement

$$\text{Input } I = \{D; M_x; M_y\} \begin{cases} \text{Density map } D \\ M_x(x,y) = \frac{x}{X} \\ M_y(x,y) = \frac{y}{Y} \end{cases}$$

X, Y are the map sizes

Input transform: $I_m = FC(I)$

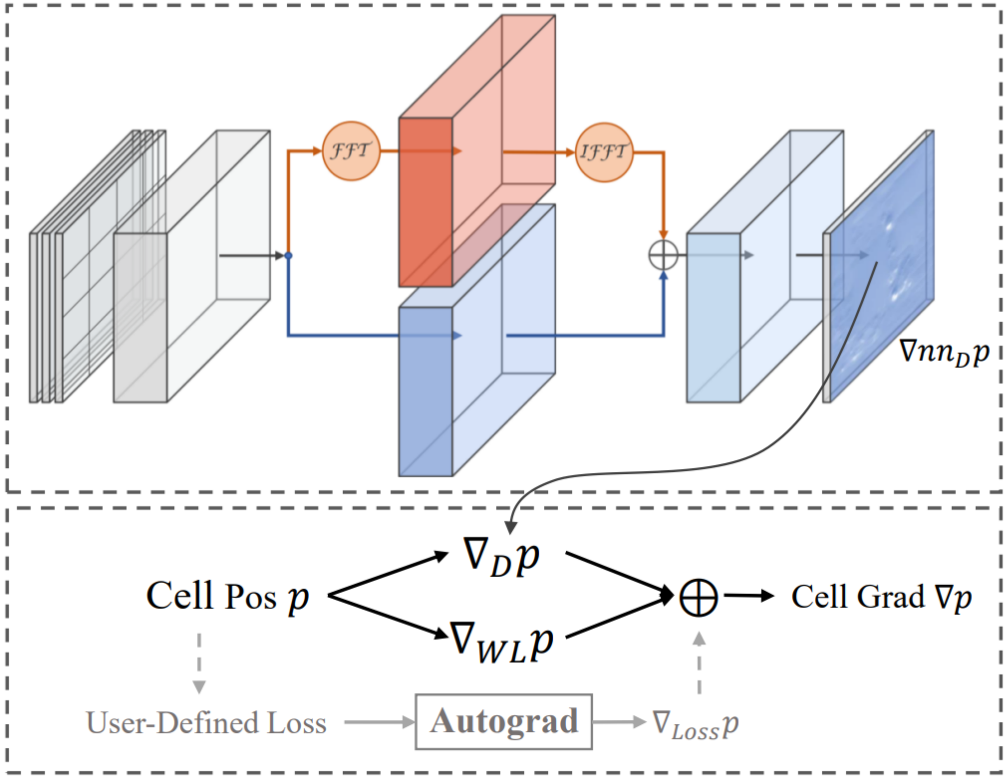
$$Freq_{layer}(I_m) = \mathcal{F}^{-1}\left(\mathcal{W}^T \cdot L(\mathcal{F}(I_m))\right)$$

$$O(I_m) = GELU\left(Conv_{2D}(I_m) + Freq_{layer}(I_m)\right)$$

Output transform: $FC^{-1}\left(O(I_m)\right)$

$\mathcal{W}$: linear transform, $\mathcal{F}$: FFT, $\mathcal{F}^{-1}$: IFFT

FC: fully-connected layer, L: low-pass-filter



Relative L2 Loss:

$$L_2(\mathbf{x_i}, f(\mathbf{x_i};\theta)) = ||f(\mathbf{x_i},\theta) - \mathbf{y_i}||_2 / ||\mathbf{y_i}||_2$$

Extending the Framework via Neural Enhancement

Model Training Data Collection

1. ISPD 2005 contest benchmarks with their respective macros
2. Standard cells are **randomly** generated at a starting position
3. Pushed cells all over the map with only the density objective $D(p)$
4. The density map and electric fields are used as training data and labels

Why train the model in low-resolution data

1. The resolution of the input maps will not affect the convolution results
2. Low frequency components describe the global information
3. Improve the adaptability of the model and speedup inference

Extending the Framework via Neural Enhancement

How to apply the nn-predicted density gradient $\nabla_{nn} D_{x,y}$

Smooth function: $\sigma(\omega) = 1 - 1/(1 - 5e^{\omega/0.05-0.5})$

Total gradient: $\nabla' D_{x,y} = (1 - \sigma)\nabla D_{x,y} + \sigma \nabla_{nn} D_{x,y}$

Experimental Results

Validation on Contest Benchmarks

Benchmarks	DREAMPlace[1]			Xplace		
	HPWL	GP/s	DP/s	HPWL	GP/s	DP/s
adaptec1	72.89	4.15	34.9	72.93	1.35	35.8
adaptec2	81.84	3.73	46.2	81.04	1.58	45.4
adaptec3	191.68	4.54	88.1	190.94	2.38	89.6
adaptec4	173.45	4.90	95.4	172.41	2.85	96.1
bigblue1	89.39	4.03	42.3	89.12	1.47	42.1
bigblue2	136.57	4.68	129.3	136.56	2.41	127.2
bigblue3	302.58	8.05	207.9	301.36	5.49	209.8
bigblue4	742.95	13.38	459.7	741.18	11.65	463.1
Sum	1791.36	47.46	1103.6	1785.6	29.18	1109.0
Ratio	1.003	1.626	0.995	1.000	1.000	1.000

ISPD 2005

Benchmarks	DREAMPlace [14]				Xplace			
	HPWL	OVFL-5	GP/s	DP/s	HPWL	OVFL-5	GP/s	DP/s
des_perf_1	1107.5	65.28	3.71	1.31	**1106.7**	64.35	1.14	1.23
fft_1	411.7	56.19	3.59	0.67	**411.3**	56.34	1.17	0.61
fft_2	**374.0**	47.72	4.28	0.69	374.3	47.49	1.18	0.64
fft_a	627.6	35.12	3.60	0.61	**625.6**	34.7	1.29	0.70
fft_b	**845.7**	51.82	3.57	0.74	846.2	52.02	1.28	0.73
matrix_mult_1	2129.2	81.02	3.71	1.61	**2116.4**	81.69	1.29	1.44
matrix_mult_2	2163.3	77.61	3.97	1.63	**2152.9**	77.95	1.23	1.48
matrix_mult_a	3036.8	48.10	4.04	2.79	**3031.7**	48.34	1.29	3.68
superblue12	25803.0	92.45	8.91	16.37	**25783.8**	93.18	4.64	17.29
superblue14	**23015.5**	63.56	4.63	11.49	23017.1	64.34	1.60	13.74
superblue19	15633.1	61.82	4.56	8.26	**15544.1**	62.39	1.46	6.60
des_perf_a†	2020.5	53.27	3.66	2.04	**1998.6**	52.32	1.18	1.67
des_perf_b†	**1610.3**	54.65	3.66	1.70	1612.6	53.64	1.27	1.58
edit_dist_a†	4217.9	80.30	3.97	2.31	**4198.7**	80.10	1.45	2.13
matrix_mult_b†	2786.7	44.86	3.82	1.98	**2765.7**	44.98	1.29	1.89
matrix_mult_c†	**2672.9**	42.13	4.07	2.07	2675.2	42.20	1.29	1.89
pci_bridge32_a†	361.8	30.55	3.54	0.82	**356.0**	30.36	1.08	0.69
pci_bridge32_b†	741.1	22.89	6.77	1.04	**714.2**	22.75	1.12	1.04
superblue11_a†	**33411.2**	54.51	5.59	13.33	33528.3	54.78	2.87	12.73
superblue16_a†	25600.9	65.85	4.38	10.60	**25505.1**	65.85	1.91	11.08
Sum	148571	1129.70	88.03	82.06	148364	1129.77	31.03	82.84
Ratio	1.001	1.000	2.837	0.991	1.000	1.000	1.000	1.000

ISPD 2015

Experimental Results

Ablation Studies of the Operator-Level Optimization Techniques

Methods	OR	OC	OE	OS	adaptec1	adaptec2	adaptec3	adaptec4	bigblue1	bigblue2	bigblue3	bigblue4	Avg
Ratio	-	-	-	-	234%	194%	136%	124%	198%	140%	123%	121%	159%
	✓	-	-	-	110%	109%	113%	115%	105%	115%	119%	118%	113%
	✓	✓	-	-	107%	107%	107%	108%	104%	108%	113%	112%	108%
	✓	✓	✓	-	104%	102%	104%	104%	102%	104%	106%	105%	104%
Xplace	Ratio				100%	100%	100%	100%	100%	100%	100%	100%	100%
	GP / Iter Time (ms)				1.478	1.671	2.325	2.688	1.572	2.441	4.974	10.018	-
DREAMPlace	Ratio				462%	345%	288%	254%	376%	288%	199%	158%	296%
	GP / Iter Time (ms)				6.832	5.769	6.699	6.840	5.915	7.023	9.904	15.831	-

Experimental Results

Neural-Enhanced Performance

Benchmarks	DREAMPlace			Xplace			Xplace-NN		
	HPWL	GP/s	DP/s	HPWL	GP/s	DP/s	HPWL	GP/s	DP/s
adaptec1	72.89	4.15	34.9	72.93	**1.35**	35.8	**72.84**	2.53	34.5
adaptec2	81.84	3.73	46.2	**81.04**	**1.58**	45.4	81.17	2.91	45.2
adaptec3	191.68	4.54	88.1	**190.94**	**2.38**	89.6	191.04	3.47	88.4
adaptec4	173.45	4.90	95.4	172.41	**2.85**	96.1	**172.38**	4.12	94.0
bigblue1	89.39	4.03	42.3	89.12	**1.47**	42.1	**89.07**	2.75	41.8
bigblue2	136.57	4.68	129.3	136.56	**2.41**	127.2	**136.27**	3.56	129.0
bigblue3	302.58	8.05	207.9	301.36	**5.49**	209.8	**301.26**	7.66	210.6
bigblue4	742.95	13.38	459.7	741.18	**11.65**	463.1	**740.44**	15.07	465.7
Sum	1791.36	47.46	1103.6	1785.6	29.18	1109.0	1784.47	42.07	1109.2
Ratio	1.003	1.626	0.995	1.000	1.000	1.000	0.999	1.442	1.000

Conclusions and Future Works

Conclusions

We develop Xplace, a new, fast and extensible GPU accelerated GP framework built on top of PyTorch, to consider factors at operator-level optimization.

- **Efficiency**: Xplace achieves around 3x speedup per GP iter with better quality compared to DREAMPlace
- **Extensiblity**: we plug into Xplace a novel Fourier neural network and illustrate a possibility of adopting neural guidance in analytical global placement

Future Works

- Handling additional constraints in placement like routability and fence regions

Q&A