IEEE/ACM 2024 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN
ICCAD

Differentiable Edge-based OPC

Guojin Chen[1,2,3], Haoyu Yang[3], Haoxing Ren[3], **Bei Yu**[1], David Z. Pan[2]

[1] The Chinese University of Hong Kong
[2] University of Texas at Austin
[3] NVIDIA Corporation

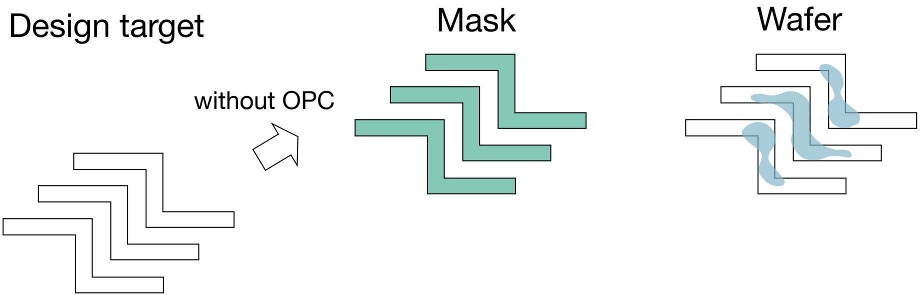
Oct. 28, 2024

Outline

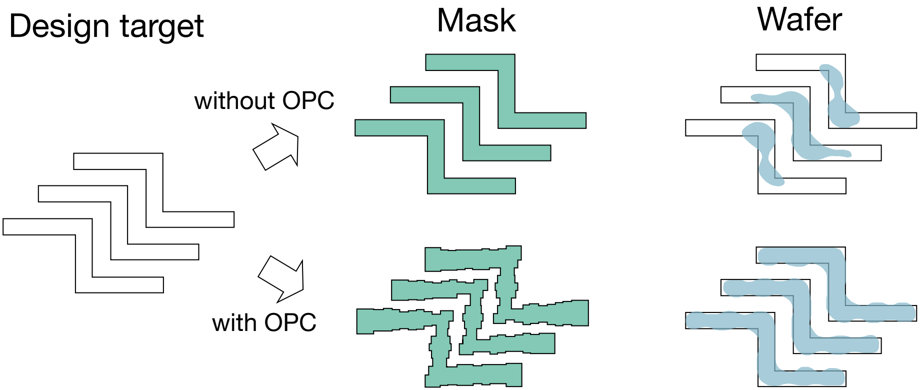
1. Introduction & Motivation
2. Differentiable Edge-based OPC
3. Experimental Results
4. Conclusion

Introduction & Motivation

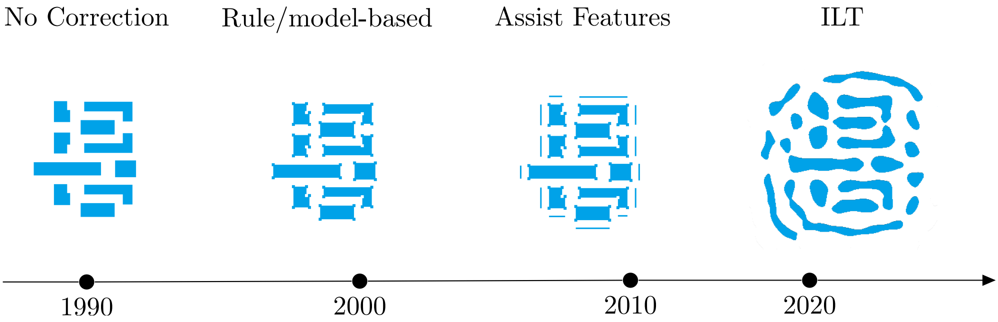
Mask Optimization



Mask Optimization

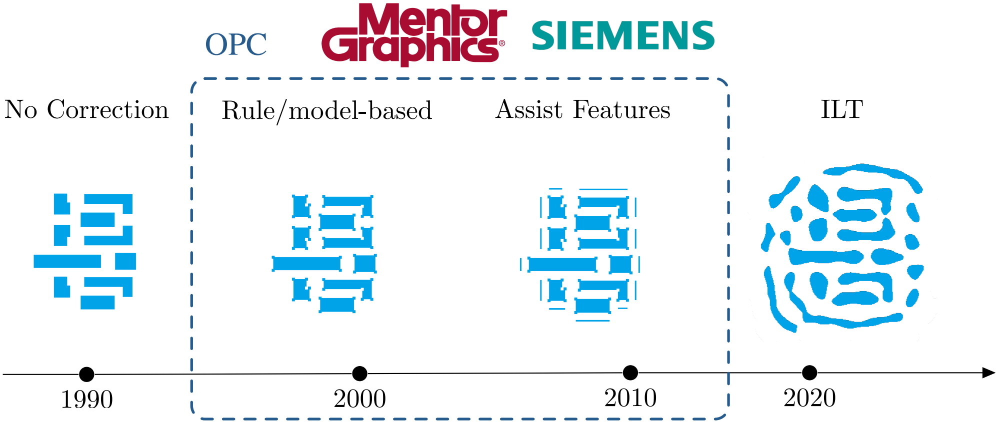


Timeline: OPC vs. ILT



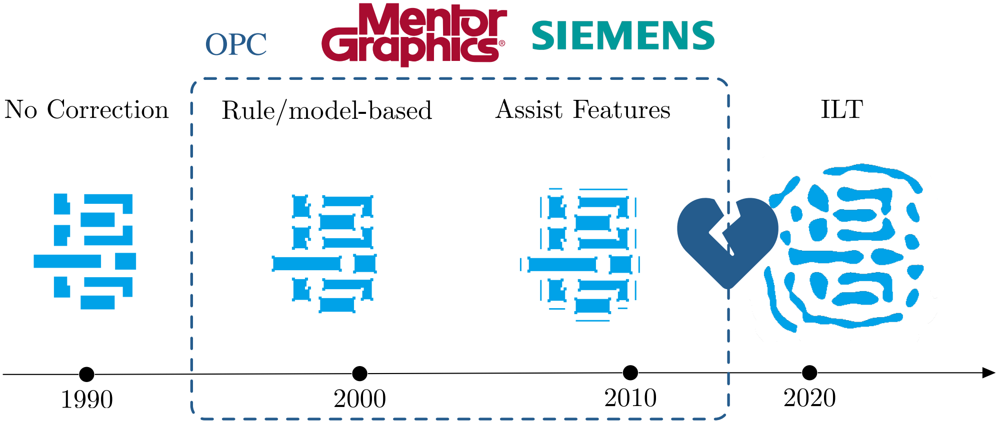
- From simple distortion to complex modification
- Computation becomes more and more complicated
- Pixel-based inverse lithography technology (ILT) is very popular in academia.

Industrial favorable solution: OPC



- Industrial refers to OPC as edge-based model-based methods.
- Due to high manufacture cost, OPC is still the most popular solution in industry.

Industrial favorable solution: OPC



- Industrial refers to OPC as edge-based model-based methods.
- Due to high manufacture cost, OPC is still the most popular solution in industry.
- New technology / innovation can not be applied.

Motivation: Differentiable ~~Edge-based~~ OPC

Question 1:

What's the difference between ILT and industrial model-based OPC?

Question 2:

Why we need Differentiable Edge-based OPC?

Industrial solution: MEEF-based OPC

MEEF: Mask Error Enhancement Factor

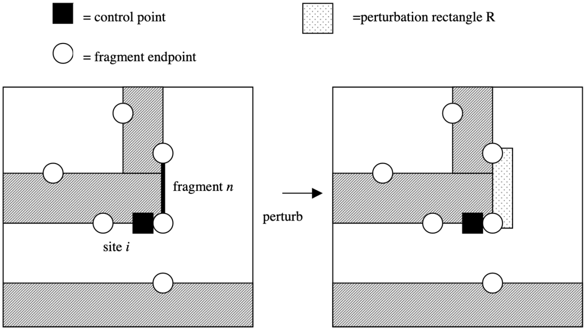
$$\boldsymbol{M} = \begin{bmatrix} \frac{\partial e_1}{\partial d_1} & \frac{\partial e_1}{\partial d_2} & \cdots & \frac{\partial e_1}{\partial d_n} \\ \frac{\partial e_2}{\partial d_1} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_r}{\partial d_1} & \frac{\partial e_r}{\partial d_2} & \cdots & \frac{\partial e_r}{\partial d_n} \end{bmatrix}$$

MEEF is the Jacobian matrix of the edge placement error e_r with respect to the displacement of the n-th edge fragment d_n.

Optimization

$\boldsymbol{e} \approx \boldsymbol{e}_0 + \boldsymbol{M} \cdot \boldsymbol{d}$, where $\boldsymbol{M}$ is the MEEF matrix.

$$\boldsymbol{M} = \begin{bmatrix} \frac{\partial e_1}{\partial d_1} & \frac{\partial e_1}{\partial d_2} & \cdots & \frac{\partial e_1}{\partial d_n} \\ \frac{\partial e_2}{\partial d_1} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_r}{\partial d_1} & \frac{\partial e_r}{\partial d_2} & \cdots & \frac{\partial e_r}{\partial d_n} \end{bmatrix}$$



Drawback of MEEF-based OPC

Drawbacks:

- Complexity: newton-method need Hessian matrix.
- Low performance: linearity assumptions[1].
- Lack co-relation with edge fragments, far away from real lithography process.

Benifits:

- Accumulate look-up table.
- Accumulate recipe for special case handling.

[1] MEEF is pre-calculated and won't change during optimization.

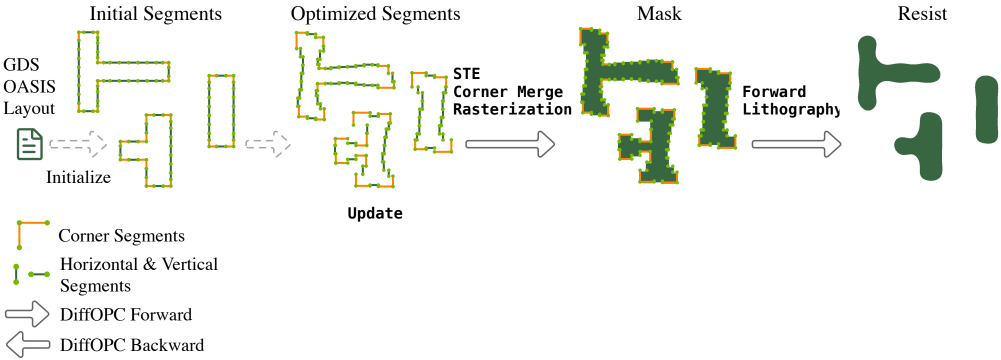
Motivation: Differentiable Edge-based OPC

What We Need:

- Global view: Using gradient to guide the edge movement
- Instead of the fixed MEEF matrix.
- Dynamic, simple, fast, and high-accuracy
- high-performance

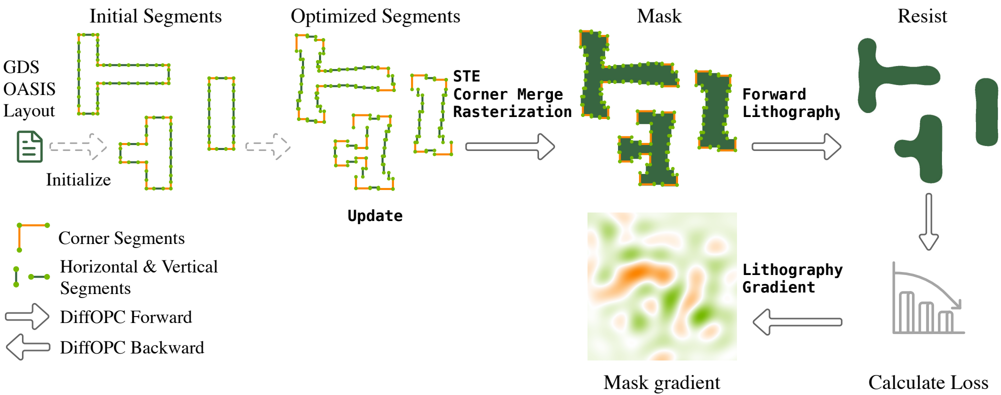
Differentiable Edge-based OPC

Differentiable Edge-based OPC Algorithm



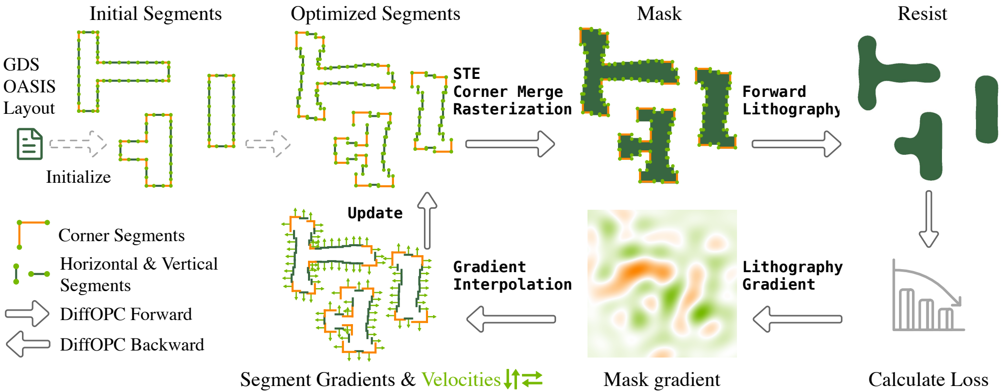
- Fragmentation: process the layout same as traditional OPC.
- Differentiable Rasterization: Edge-based representation -> pixel-based representation.
- Lithography and gradient interpolation: same as ILT.

Differentiable Edge-based OPC Algorithm



- Fragmentation: process the layout same as traditional OPC.
- Differentiable Rasterization: Edge-based representation -> pixel-based representation.
- Lithography and gradient interpolation: same as ILT.

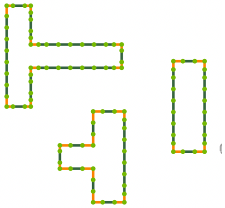
Differentiable Edge-based OPC Algorithm



- Fragmentation: process the layout same as traditional OPC.
- Differentiable Rasterization: Edge-based representation -> pixel-based representation.
- Lithography and gradient interpolation: same as ILT.

Fragmentation

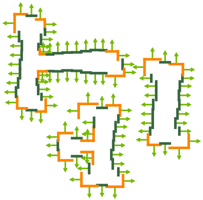
Initial Segments



Metadata: information to re-construct polygons

```
metadata = {
    img_shape,
    polygon_ids,
    fragments,
    direction_vector,
    velocities,
    corner_ids,
    seg_types,
    ...
}
```

Segments for optimization



For each edge, the edge coordinates are optimization parameters.
Fragments velocities are pre-defined based on edge direction.

Edge parameters

For each edge: `[start_x, start_y, end_x, end_y]`.
Layout:

```
[
    [ start_x, start_y, end_x, end_y],
    [ start_x, start_y, end_x, end_y],
    ...
    [ start_x, start_y, end_x, end_y]
]
```

Problem Definition:

Given input edge parameters $\boldsymbol{S}$, the goal is to optimize the edge parameters $\boldsymbol{S}$ to minimize the OPC related loss function L.

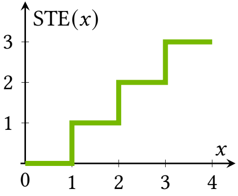
Edgh parameter rounding and clamping: continuous to discrete

$$\bar{x}_i = \mathrm{STE}(x_i),\ \bar{y}_i = \mathrm{STE}(y_i),\ \bar{\boldsymbol{s}}_i = \mathrm{STE}(\boldsymbol{s}_i),\ \bar{\boldsymbol{S}} = \mathrm{STE}(\boldsymbol{S}). \tag{1}$$

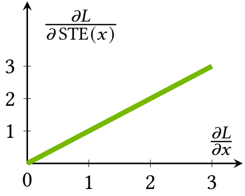
STE is defined as:

$$\begin{aligned} \bar{x} &= \mathrm{STE}(x) = \mathrm{Round}(x), && \triangleright\ \textit{STE forward.} \\ \frac{\partial L}{\partial\, \mathrm{STE}(x)} &= \frac{\partial L}{\partial x}. && \triangleright\ \textit{STE backward.} \end{aligned} \tag{2}$$

- forward pass (left figure) applies the rounding function to $\boldsymbol{S}$
- backward pass (right figure) directly propagates the gradients from $\bar{\boldsymbol{S}}$ to $\boldsymbol{S}$

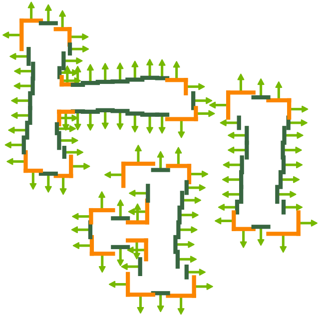


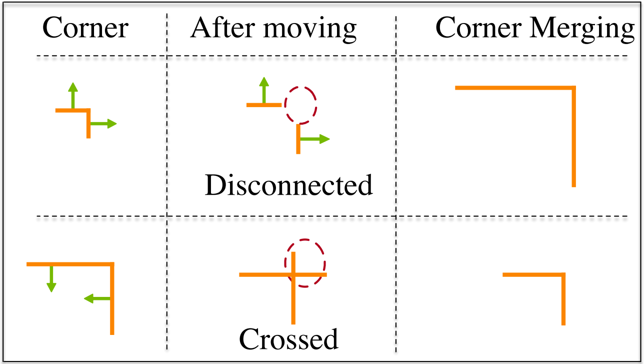
(a) STE forward



(b) STE backward

Corner edge merging



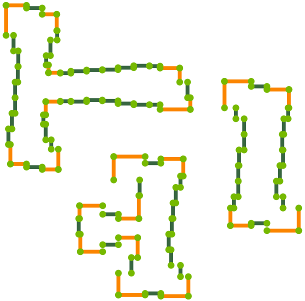


After moving edges,

- The **non-corner edges** can be connected automatically without changing endpoints.
- The **corner edges** may be disconnected or overlapped / crossed with each other. Need further processing.

Differentiable rasterization

Optimized Segments
(Edge representation)



① STE

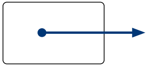
② Corner Merge

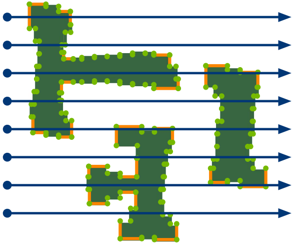
③ Differentiable Rasterization

Mask
(Pixel representation)

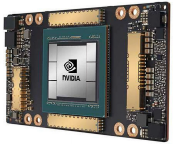
Differentiable Rasterization: **Edge-based** representation -> **pixel-based** representation.

Ray casting algorithm: CUDA-accelerated

	Ray casting	**Number of intersections**
Point Inside Polygon		Odd number
Point Outside Polygon		Even number



Computing all the points **in parallel**
Using GPU

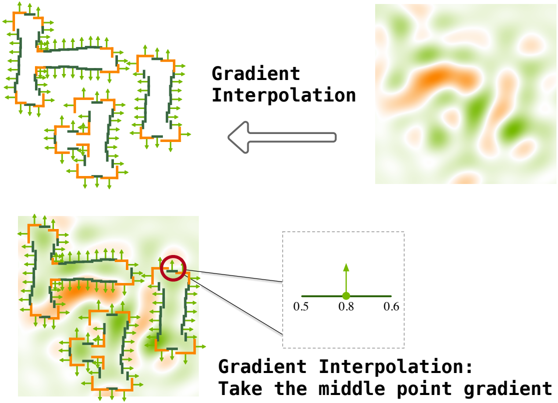


Using CUDA to accelerate the ray casting algorithm, parallelizing the ray casting process to rasterize the edges.

Loss calculation and gradient interpolation

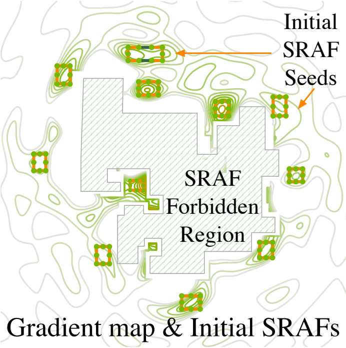
Edge gradient
(Edge representation)

Mask gradient
(Pixel representation)



- Loss calculation: same as ILT.
- Gradient interpolation: **middle point** of the edge's gradient value

Gradient map-based SRAF seeds generation



(a)

(b)

DiffOPC SRAF insertion and optimization.

Experimental Results

Results on ICCAD13

Table: Comparison with ILT methods on ICCAD13 dataset.

		ICCAD'20 NeuralILT					DAC'23 MultiILT					MultiILT (Post-MRC)					DiffOPC				
		L2	PVB	EPE	#shots	TAT	L2	PVB	EPE	#shots	TAT	L2	PVB	EPE	#shots	TAT	L2	PVB	EPE	#shots	TAT
c1	215344	50795	63695	8	743	13.57	40779	50661	3	307	3.49	45940	54949	7	275	18.42	38661	55156	3	107	10.62
c2	169280	36969	60232	3	571	14.37	34201	44322	2	186	3.47	37035	45085	3	167	13.64	29548	45610	0	104	10.65
c3	213504	94447	85358	52	791	9.72	66486	71527	22	308	3.47	79751	82213	35	261	18.71	64706	93773	19	121	11.52
c4	82560	17420	32287	2	209	10.40	10942	21500	0	233	3.47	13111	32330	1	204	19.24	12054	25053	0	80	6.04
c5	281958	42337	65536	3	631	10.04	30231	51277	0	374	3.47	39236	60069	1	296	13.23	31774	56966	0	129	6.72
c6	286234	39601	59247	5	745	11.11	30741	44982	0	365	3.47	37493	56581	1	300	16.14	31791	52997	0	129	10.33
c7	229149	25424	50109	0	354	9.67	17101	40294	0	196	3.50	19133	48156	0	155	13.57	17847	45791	0	96	6.59
c8	128544	15588	25826	0	467	11.81	11935	20357	0	243	3.47	13917	28910	0	201	22.09	11641	23172	0	78	6.52
c9	317581	52304	68650	2	653	9.68	35805	57930	0	435	3.50	45659	70023	1	387	14.51	36595	65732	0	141	10.11
c10	102400	10153	22443	0	423	11.46	8825	18470	0	114	3.48	9715	22979	0	88	18.23	8184	17923	0	76	5.12
Average		38503.8	53338.3	7.5	558.7	11.18	28704.6	42132.0	2.7	276.1	3.48	34099	50130	4.9	233.4	16.78	**28280**	48217	**2.2**	**106.1**	8.42
Ratio		1.36	1.11	3.41	5.27	1.33	1.02	0.87	1.23	2.60	0.41	1.21	1.04	2.23	2.20	1.99	**1.00**	1.00	**1.00**	**1.00**	1.00

- Best L2 error, lowest EPE error.
- Reduced 60% manufacturing cost.

Results on Larger Dataset

Table: Comparison with ILT methods on larger dataset.

		ICCAD'20 NeuralILT					DAC'23 MultiILT					MultiILT (Post-MRC)					DiffOPC				
		L2	PVB	EPE	#shots	TAT	L2	PVB	EPE	#shots	TAT	L2	PVB	EPE	#shots	TAT	L2	PVB	EPE	#shots	TAT
L1	494560	79933	120577	12	669	20	64020	93060	3	628	3.48	80403	101194	11	529	22.66	57178	97979	3	247	11.49
L2	448496	86995	104266	15	556	12	52072	84733	1	553	3.46	72261	91673	8	491	18.07	63288	85388	2	109	11.89
L3	492720	133281	152718	70	766	15	95174	116687	30	641	3.49	118860	125013	65	564	20.49	81120	120828	22	267	14.69
L4	361776	43797	92137	0	455	14	33076	67839	1	523	3.47	41526	76582	2	479	21.43	31531	70713	0	177	8.53
L5	561174	69521	122115	3	808	19	55013	100120	0	670	3.46	76176	111861	6	528	16.00	53484	102675	0	258	7.69
L6	565450	73790	117359	2	764	19	57386	94863	0	670	3.45	76644	108667	5	501	18.74	56581	97980	0	293	13.21
L7	445365	49031	92320	0	531	19	32947	73799	0	648	3.45	40838	84006	0	503	18.19	42091	84836	0	222	9.91
L8	407760	47409	84971	0	478	16	41265	67797	0	493	3.48	43475	73021	0	426	20.49	32482	68687	0	198	8.53
L9	596797	93922	115028	5	614	14	70385	108998	0	541	3.48	84857	120426	4	514	18.04	60748	111449	0	226	11.44
L10	381616	28028	80127	0	452	19	30091	62206	0	546	3.46	36767	67807	0	452	20.95	28334	63274	0	188	8.78
Average		70570.7	108161.8	10.7	609.3	16.7	53142.9	87010.2	3.5	591.3	3.47	67181	96025	10.1	498.7	19.51	**50684**	90381	**2.7**	**218.5**	10.62
Ratio		1.39	1.20	3.96	2.79	1.57	1.05	0.96	1.30	2.71	0.33	1.33	1.06	3.74	2.28	1.84	**1.00**	1.00	**1.00**	**1.00**	1.00

- Best L2 error, lowest EPE error.
- Reduced 64% manufacturing cost.

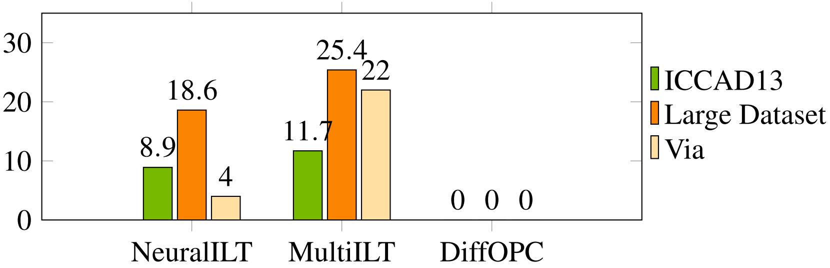
Comparasion with MEEF-based OPC

Table: Comparison with traditional MEEF EBOPC on ICCAD 2013 benchmark.

	MEEF-based EBOPC					DiffOPC w./o. SRAFs				
	L2	PVB	EPE	#shots	TAT	L2	PVB	EPE	#shots	TAT
c1	52310	60296	14	67	13	42177	57981	4	79	5.53
c2	36498	52124	2	60	11	31198	50474	2	58	5.35
c3	90824	103100	59	87	12	71643	81219	26	92	6.52
c4	12144	30663	2	34	9	14771	32059	0	30	3.29
c5	31832	60792	0	84	14	33986	61796	0	89	5.24
c6	30612	55751	0	98	14	33578	56752	0	85	5.44
c7	15343	48968	0	59	11	17928	48886	0	60	4.16
c8	11851	26149	0	33	9	12805	25942	0	43	3.82
c9	38858	71288	0	93	14	39543	73183	0	97	4.70
c10	6562	21024	0	26	9	8167	21332	0	19	3.39
Avg.	32683.4	53015.5	7.7	64.1	11.6	**30579.6**	**50962.4**	**3.2**	65.2	**4.74**
Ratio	1.07	1.04	2.33	0.98	2.44	**1.00**	**1.00**	**1.00**	1.00	**1.00**

- DiffOPC exhibits lower average PVB, EPE, and TAT, respectively.

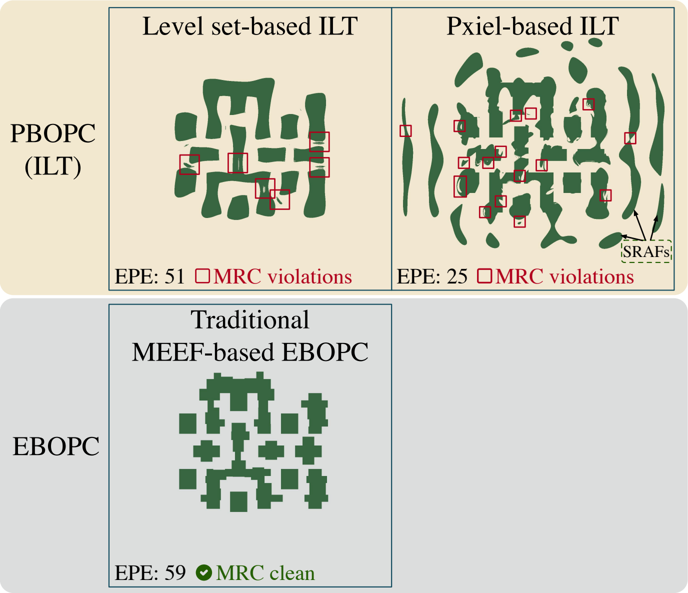
MRC violations across methods and datasets



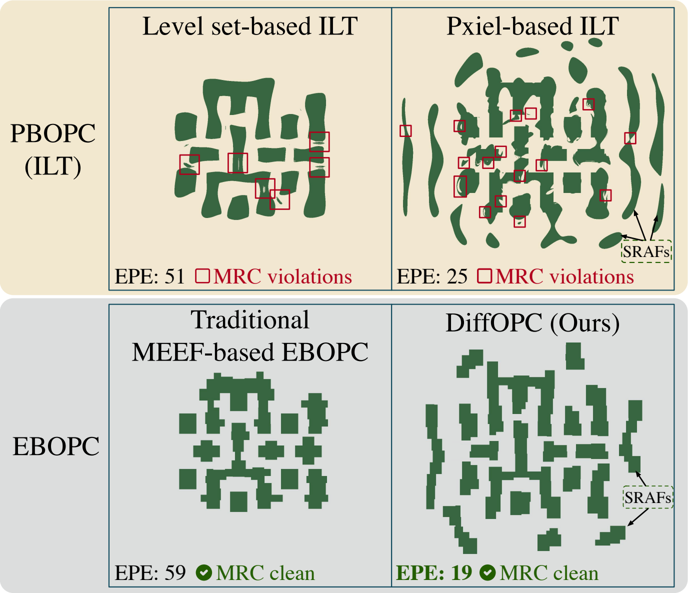
MRC violations across methods and datasets

Conclusion

Conclusion


PBOPC
(ILT)
Level set-based ILT
EPE: 51
MRC violations
Pxiel-based ILT
SRAFs
EPE: 25
MRC violations
EBOPC
Traditional
MEEF-based EBOPC
EPE: 59
MRC clean

Conclusion


PBOPC
(ILT)
Level set-based ILT
EPE: 51 MRC violations
Pxiel-based ILT
SRAFs
EPE: 25 MRC violations
EBOPC
Traditional
MEEF-based EBOPC
EPE: 59 MRC clean
DiffOPC (Ours)
SRAFs
EPE: 19 MRC clean

THANK YOU!