MANHATTAN ASSEMBLY: DIFFERENTIABLE MASK OPTIMIZATION WITH MANHATTAN PRIMITIVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Inverse Lithography Technology (ILT) formulates mask optimization as an inverse-design problem subject to manufacturing constraints, yet existing methods struggle to jointly balance printing fidelity, manufacturability, and scale. We present **Manhattan Assembly**, a primitive-based framework that represents photomasks as axis-aligned rectangular primitives and optimizes their continuous geometry through a differentiable renderer while preserving Manhattan structure by construction. A lithography-guided adaptive refinement strategy adjusts primitive allocation through splitting and pruning, concentrating representation capacity in critical hotspot regions, while global primitive sharing with window-wise computation enables full-chip optimization. Experiments on ICCAD13-S and full-chip layouts from real chip designs demonstrate improved printing fidelity and process-window robustness with reduced mask complexity. In the full-chip setting, Manhattan Assembly reduces L_2 and PVB by 34.2% and 9.1%, respectively, relative to the best SOTA baseline for each metric.

1 INTRODUCTION

Lithography transfers circuit patterns onto wafers through photomasks, lying at the heart of semiconductor manufacturing. As process nodes shrink, diffraction distorts the printed patterns, requiring Optical Proximity Correction (OPC) to compensate for such deviations by modifying mask geometry (Pan et al., 2013; Spence, 2005; Chen et al., 2024b). Inverse Lithography Technology (ILT) formulates this correction as an optimization problem, adjusting the mask guided by lithography simulation to match the target layout (Poonawala & Milanfar, 2007; Ma & Arce, 2007; Shen et al., 2009; Liu & Zakhor, 1992). Beyond printing fidelity, geometric constraints and scalability should be ensured in modern large-scale mask optimization (Yin et al., 2024; Wu et al., 2025b; Pang, 2021). ILT thus poses a challenging inverse-design problem driven by physics, calling for a representation that combines structural control, geometric flexibility, and scalability.

Such structural constraints are inherent to IC layout design: wires and vias are predominantly routed along orthogonal directions to form so-called Manhattan structures (Wu et al., 2025a; Weste & Harris, 2011). This structure persists across downstream design flows, from placement and routing to physical verification, providing a unified geometric prior across stages. As illustrated in Figure 1, pixel-based ILT provides substantial geometric freedom (Poonawala & Milanfar, 2007; Shen et al., 2009; Sun et al., 2023; Jiang et al., 2020), but does not explicitly enforce Manhattan structure, often violating standard manufacturing rules and impeding industrial integration. Conversely, existing Manhattan-based methods optimize predefined edge segments (Chen et al., 2024b; Wu et al., 2025a; Cobb & Zakhor, 1995) with fixed representation capacity, which limits flexibility and fails to align with spatially varying correction demands: most regions require only coarse adjustments, whereas critical hotspots need fine-grained corrections.

Explicit primitive representations offer a promising solution to this challenge. In visual reconstruction, approaches such as 3D Gaussian splatting demonstrate how continuous parameter optimization and adaptive primitive allocation can be unified within a shared spatial representation (Kerbl et al., 2023; Fan et al., 2024). Similar primitive-based representations have also been explored for CAD shape modeling and differentiable vector graphics rendering (Chen et al., 2020; Yu et al., 2022; Li et al., 2020; Jones et al., 2020). In lithographic inverse design, however, the representation must serve a physics-driven objective: the optimized mask is only an intermediary, and the underlying

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

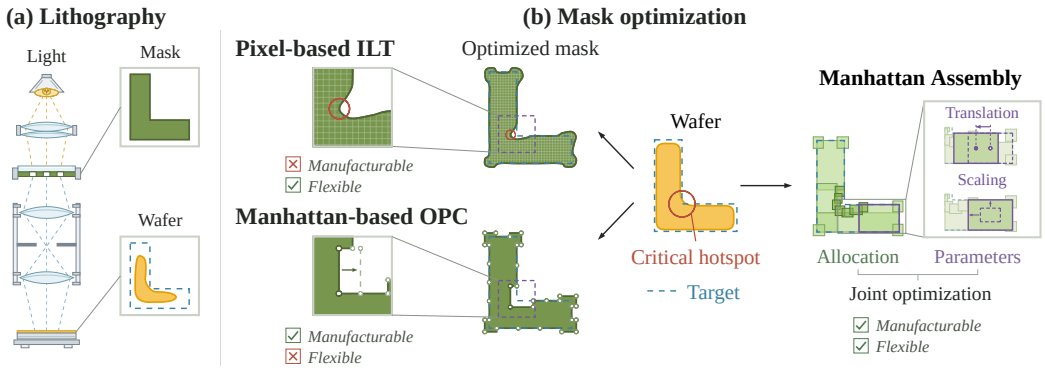

Figure 1: Representation challenge in lithographic mask optimization. (a) Lithographic effects distort the printed pattern from the intended geometry. (b) Pixel-based ILT offers high flexibility at the cost of manufacturability, while Manhattan-based OPC preserves rectilinear structure but provides limited local flexibility, leaving critical hotspots insufficiently corrected.

goal is to match the printed pattern on the wafer with the target layout. Consequently, existing strategies that optimize and allocate primitives against target-image error cannot directly apply; primitive configurations must instead be guided by printing feedback from lithography simulation.

Building on this perspective, we propose **Manhattan Assembly**, a structured primitive representation for lithographic mask optimization under Manhattan constraints. We represent the mask as a union of axis-aligned rectangles, whose continuous positions and sizes enable flexible geometric optimization while preserving Manhattan structure by construction. Beyond optimizing geometry within a fixed parameterization, Manhattan Assembly jointly adapts primitive parameters and spatial allocation to lithographic correction needs. This allows representational capacity to concentrate where additional local flexibility is needed while limiting unnecessary complexity elsewhere. A globally shared primitive set further provides a unified geometric representation across the entire mask, supporting consistent optimization from local patterns to full-chip layouts.

Specifically, Manhattan Assembly maps the primitive composition to a pixel-domain mask through a differentiable rectangle renderer, allowing the primitive geometry to be optimized against wafer-domain printing objectives. A lithography-guided adaptive refinement strategy then dynamically reallocates primitives through two complementary operations: splitting bisects selected primitives at loss plateaus to provide additional degrees of freedom, while pruning removes redundant primitives with negligible impact on the printing loss. To scale to full-chip layouts, Manhattan Assembly performs rendering and simulation over local windows and aggregates gradients across a unified global primitive set, ensuring seamless geometric consistency.

On ICCAD13 (Banerjee et al., 2013), Manhattan Assembly achieves the lowest L_2 while maintaining PVB comparable to the strongest baseline (Wu et al., 2025a; Yang & Ren, 2026). On full-chip layouts from real designs, it leads on both metrics, reducing L_2 by 34.2% and PVB by 9.1% relative to the strongest baseline on each (Wu et al., 2025b;a). Additional mask-rule evaluations show better compliance with width, spacing, and notch constraints than the baselines, underscoring the practical value of the proposed representation.

2 Related Work

Pixel-level Mask Optimization. Optical proximity correction (OPC) modifies mask geometries to compensate for lithographic distortions, while inverse lithography technology (ILT) formulates this as an inverse problem driven by lithography simulation (Chen et al., 2024b; Poonawala & Milanfar, 2007). Pixel-domain methods represent masks on dense grids for geometric freedom. Within numerical optimization, MultiILT adopts multi-resolution grids (Sun et al., 2023), while CurvyILT combines corner smoothing with differentiable morphology for curvilinear generation and rule compliance (Yang & Ren, 2025; 2026). For full-chip layouts, FuILT coordinates tile-wise optimization via gradient fusion in overlapping regions, mitigating boundary artifacts (Yin et al., 2024). Raster masks are also widely used in learning-based synthesis (Yang et al., 2018; Chen et al., 2022; Ma &

Hao, 2023; Zhu et al., 2024), extending to full-chip optimization in LVM-MO (Wu et al., 2025b) and diffusion-based generation in HiFiMO (Wang et al., 2025a). However, Manhattan structure is not explicitly encoded in pixel-domain variables. Manhattan Assembly instead represents masks as unions of axis-aligned rectangles, preserving Manhattan geometry throughout optimization.

Geometric Representations for Mask Optimization. Geometric parameterizations explicitly control mask geometry (Cobb & Zakhor, 1995). Block-based OPC uses overlapping basis blocks for improved manufacturability (Ma et al., 2013), later extended with level-set optimization and adaptive step sizes for faster convergence (Huang et al., 2025). DiffOPC propagates lithography gradients to edge-segment positions (Chen et al., 2024b), while TokMan generates Manhattan-aligned polygons as token sequences via a diffusion transformer (Wu et al., 2025a). Zheng et al. (2026) adaptively insert B-spline control points near critical regions within a curvilinear boundary representation. These methods share a key challenge: matching local geometric capacity to spatially varying correction demands while controlling mask complexity. Manhattan Assembly targets adaptive capacity allocation under Manhattan constraints, combining continuous optimization of rectangle positions and sizes with lithography-guided splitting and pruning, allowing correction capacity to grow in critical regions while pruning primitives with negligible impact on fidelity. Globally shared primitive parameters further coordinate updates across computational windows for full-chip optimization.

Primitive-Based Representation and Refinement. Primitive-based representations construct complex geometries from parameterized elements (Tulsiani et al., 2017; Paschalidou et al., 2019; Sharma et al., 2018), including convex parts via half-space intersections in BSP-Net (Chen et al., 2020) and quadric CSG compositions in CAPRI-Net (Yu et al., 2022). In vector graphics and neural rendering, DiffVG and 3D Gaussian Splatting (3DGS) optimize primitives via differentiable rasterization (Li et al., 2020; Kerbl et al., 2023). Beyond continuous updates, adaptive refinement dynamically adjusts capacity: 3DGS densifies primitives via positional gradients and prunes by opacity (Kerbl et al., 2023), later refined via gradient reformulation, pixel coverage, frequency regularizers, multiscale structures, or loss landscapes (Ye et al., 2024; Zhang et al., 2024b;a; Lyu et al., 2026; Wang et al., 2025b). Adapting such strategies to mask representation requires handling union geometry, where overlapping regions contribute once and each primitive's contribution depends on the remaining assembly, as well as how geometric changes propagate through simulation to fidelity. Manhattan Assembly probes local correction demands via child-gradient discrepancies after geometry-preserving subdivisions, and evaluates pruning by the printing-loss impact of removal.

3 METHOD

As illustrated in Figure 2, this section first introduces the differentiable Manhattan representation and its rectangle rendering scheme, upon which we formulate primitive-based mask optimization. We then adjust the primitive composition via lithography-guided adaptive refinement, and enable full-chip scalability through global primitive sharing.

3.1 MANHATTAN PRIMITIVE REPRESENTATION

We represent the mask geometry using a set of N axis-aligned rectangular primitives $\mathcal{P} = \{p_i\}_{i=1}^{N}$, where each primitive p_i is specified by a parameter vector

$$\theta_i = (x_i, y_i, w_i, h_i), \tag{1}$$

where $(x_i, y_i) \in \mathbb{R}^2$ denotes its center position, and $w_i, h_i > 0$ denote its half-width and half-height, respectively.

This representation offers two main advantages. First, compared with dense pixel-domain representations, rectangular primitives provide an explicit, compact, and continuous geometric representation. Pixel-based methods optimize the mask directly on a pixel grid, where the variables do not explicitly encode the Manhattan geometric prior (Poonawala & Milanfar, 2007; Shen et al., 2009). Conversely, our representation describes the mask using continuous position and scale parameters, ensuring that geometric updates preserve the Manhattan structure throughout optimization.

Second, compared with existing Manhattan-based methods that adjust predefined polygon boundaries with limited flexibility (Cobb & Zakhor, 1995; Chen et al., 2024b; Wu et al., 2025a), our

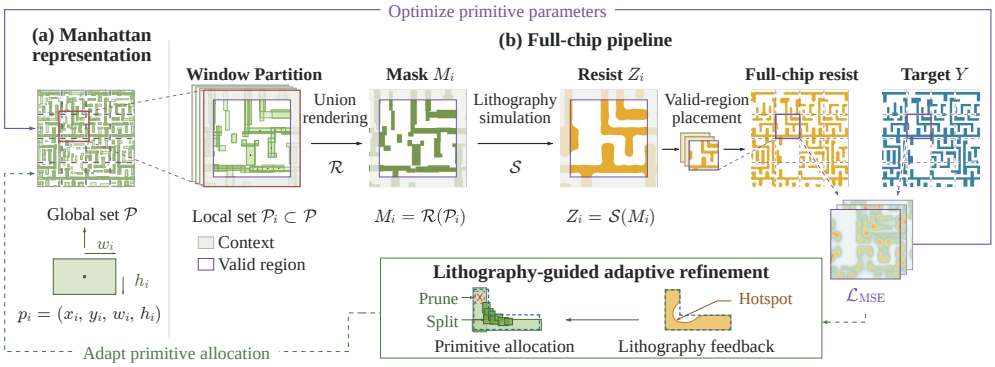

Figure 2: Overview of Manhattan Assembly. (a) The full-chip mask is represented by a globally shared set $\mathcal{P}$ of axis-aligned rectangular primitives. (b) For each window i, a local primitive subset $\mathcal{P}_i \subset \mathcal{P}$ is extracted and rendered into a local mask M_i, which is subsequently processed by lithography simulation to obtain the resist pattern Z_i. Predictions within the valid regions are assembled into the full-chip resist, while the resulting printing feedback guides adaptive primitive allocation and updates the shared global parameters.

primitives optimize their positions and scales independently. As the overall mask is formed by the union of these primitives, its boundary can freely change topology without being restricted to a fixed polygonal connectivity. Moreover, this explicit parameterization facilitates incorporating geometric constraints, such as minimum feature size and spacing, directly into optimization to promote manufacturability.

3.2 Differentiable Rectangle Rendering

The rectangle set $\mathcal{P}$ defines the mask geometry, while lithography simulation takes a pixel-domain mask as input. We therefore construct a differentiable rectangle renderer $\mathcal{R}$ that maps the primitive set to a mask $M = \mathcal{R}(\mathcal{P})$. For each primitive p_i, its rectangular region is

$$B_i = [x_i - w_i,\ x_i + w_i] \times [y_i - h_i,\ y_i + h_i]. \tag{2}$$

Each pixel value $M(u) \in [0, 1]$ is the fraction of its area covered by the geometric union $\bigcup_i B_i$, so overlapping regions contribute only once.

The union coverage is piecewise polynomial in the geometric parameters and differentiable almost everywhere. Let $\ell_{i,u}^{\{R,L,T,B\}}$ denote the exposed edge lengths of primitive B_i on the union boundary within pixel u, with area a_u. At differentiable configurations, the pixel derivatives are

$$\begin{aligned} \frac{\partial M(u)}{\partial x_i} &= \frac{\ell_{i,u}^{R} - \ell_{i,u}^{L}}{a_u}, & \frac{\partial M(u)}{\partial y_i} &= \frac{\ell_{i,u}^{T} - \ell_{i,u}^{B}}{a_u}, \\ \frac{\partial M(u)}{\partial w_i} &= \frac{\ell_{i,u}^{R} + \ell_{i,u}^{L}}{a_u}, & \frac{\partial M(u)}{\partial h_i} &= \frac{\ell_{i,u}^{T} + \ell_{i,u}^{B}}{a_u}. \end{aligned} \tag{3}$$

At rectangle contacts or coincident boundaries, the union coverage may be nondifferentiable. For the backward pass, shared interfaces inside the union contribute no boundary terms, while exposed edge segments on the union boundary provide gradients for the corresponding parameters.

3.3 Primitive-Based Mask Optimization

Given the current primitive composition $\mathcal{P}$, Manhattan Assembly optimizes the collection of continuous geometric parameters $\Theta = (\theta_1, \ldots, \theta_N)$, and we denote the induced primitive set by $\mathcal{P}(\Theta)$. At each iteration, the primitive set $\mathcal{P}(\Theta)$ is converted by the renderer $\mathcal{R}$ into a pixel-domain mask, which is then passed through the differentiable lithography model $\mathcal{S}$ to predict the wafer pattern.

$$\min_{\Theta} \mathcal{L}_{\text{MSE}} \left(\mathcal{S} \left(\mathcal{R} \left(\mathcal{P}(\Theta) \right) \right), Y \right), \tag{4}$$

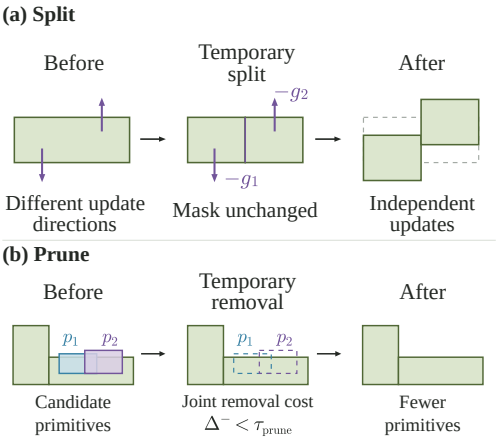

Figure 3: Lithography-Guided Adaptive Refinement Strategy.

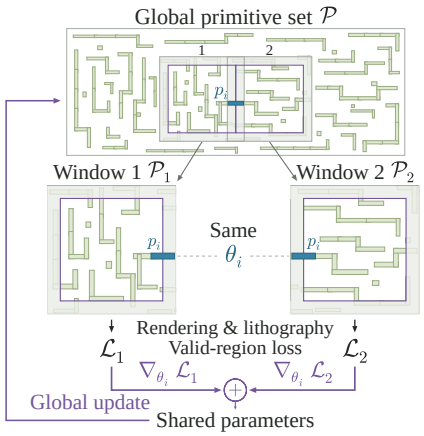

Figure 4: Global primitive sharing with valid region gradients accumulated across windows.

where Y denotes the target wafer pattern. The wafer-domain printing error provides the supervisory signal for geometric optimization, and its gradient is backpropagated through the lithography model and rectangle renderer to update the primitive parameters.

To promote manufacturability, we impose geometric constraints on primitive width and height, spacing between disjoint primitives, and notch size in the assembled mask. These constraints are enforced throughout the optimization of Θ.

3.4 Lithography-Guided Adaptive Refinement Strategy

Different parts of the mask require different degrees of correction during optimization. We therefore adapt the primitives' spatial allocation according to lithographic feedback, alongside their parameter optimization: splitting introduces independent degrees of freedom in parts requiring finer adjustment, while pruning removes primitives with negligible contribution, as shown in Figure 3.

Split A plateau in the printing loss suggests limited gains from further parameter optimization, without revealing where new degrees of freedom are needed: different parts of a primitive may require opposing corrections, causing their gradient contributions to cancel despite persistent local errors. We therefore use temporary splits to assess these local gradient demands separately.

For each primitive, we bisect its rectangle along either the horizontal or vertical centerline into two sub-rectangles whose union equals the original rectangle, so the temporary split leaves the rendered mask, wafer prediction and printing loss unchanged.

For each split direction d, we temporarily replace p_i with its two sub-rectangles and propagate the printing loss through $\mathcal{S} \circ \mathcal{R}$ to obtain their parameter gradients g_1 and g_2. We define

$$m_{i,d}^{\text{diff}} = \left\| (g_1 - g_2)_{I_d} \right\|_2, \qquad m_{i,d}^{\text{sum}} = \left\| (g_1 + g_2)_{I_d} \right\|_2, \tag{5}$$

where I_d selects (y, h) for a vertical split and (x, w) for a horizontal split. We then select the direction with the larger $m_{i,d}^{\text{diff}}$, denoting the corresponding metrics by m_i^{diff} and m_i^{sum}. Aggregating them over all primitives, we define

$$\rho = \frac{\sum_{i=1}^{N} m_i^{\text{diff}}}{\sum_{i=1}^{N} \left(m_i^{\text{diff}} + m_i^{\text{sum}} \right)}. \tag{6}$$

When ρ exceeds the preset threshold τ_{split}, we execute a split step only if it satisfies the geometric constraints. Within a fixed budget of new primitives, we rank the eligible primitives by m_i^{diff} and prioritize those with larger gradient discrepancies.

Prune As parameters are updated and splitting continues, we prune primitives that become redundant based on the change in printing loss induced by their removal. For any candidate subset $Q \subseteq \mathcal{P}$, we define the removal cost $\Delta^-(Q \mid \mathcal{P})$ as the loss increase caused by removing Q:

$$\Delta^-(Q \mid \mathcal{P}) = \mathcal{L}_{\mathrm{MSE}}(\mathcal{S}(\mathcal{R}(\mathcal{P} \setminus Q)), Y) - \mathcal{L}_{\mathrm{MSE}}(\mathcal{S}(\mathcal{R}(\mathcal{P})), Y). \tag{7}$$

For each primitive, we first compute the loss change caused by its isolated removal. Primitives with removal costs below τ_{prune} are treated as candidates and ranked in ascending order of their costs. To account for dependencies among overlapping primitives, we construct the removal set Q greedily. Starting from $Q = \emptyset$, we traverse the ranked candidates in order and add p_i to Q whenever $\Delta^-(Q \cup \{p_i\} \mid \mathcal{P}) < \tau_{\mathrm{prune}}$, provided that the resulting assembly satisfies the geometric constraints.

After each round of primitive adaptation, we continue optimizing the primitives' positions and scales on the updated $\mathcal{P}$, and perform further splitting and pruning based on subsequent lithographic feedback, allowing the geometric parameters and primitive allocation to adapt jointly.

3.5 Global Primitive Sharing for Full-Chip Scaling

To scale optimization to large layouts, we represent the full-chip mask with a globally shared primitive set $\mathcal{P}$ and perform rendering and lithography simulation over fixed-size windows while optimizing all primitive parameters globally. Each window contains a mutually non-overlapping valid region with surrounding context to account for optical proximity effects, illustrated in Figure 4, and the printing loss is evaluated only within the valid region.

A primitive p_i may participate in multiple windows while sharing the same parameters θ_i. Its gradient therefore accumulates contributions from all corresponding windows:

$$\nabla_{\theta_i} \mathcal{L} = \sum_{w:\, p_i \in \mathcal{P}_w} \nabla_{\theta_i} \mathcal{L}_w, \tag{8}$$

where $\mathcal{P}_w$ denotes the primitives rendered within window w, and $\mathcal{L}_w$ is its printing loss. Windowing is thus used only for rendering and evaluation, while parameters and refinement signals are updated on the global primitive set, avoiding the stitching artifacts common to other full-chip methods.

4 Experiments

4.1 Experimental Setup

Settings. Manhattan Assembly is implemented in PyTorch (Paszke et al., 2019), with differentiable rectangle rendering accelerated by custom CUDA kernels. All experiments are conducted on NVIDIA A100 GPUs, using a single GPU for ICCAD13 (Banerjee et al., 2013) and four GPUs for CHIP-L. For ICCAD13, we initialize 500 rectangular primitives with centers placed on a uniform grid and optimize them for 500 iterations. For lithography simulation, we use the official simulator for ICCAD13 and TorchLitho (Chen et al., 2024a) for full-chip evaluation on CHIP-L. Complete optimization and lithography settings are provided in Supplementary Material A.

Dataset. We evaluate on the ICCAD13 benchmark (Banerjee et al., 2013) and our large-scale CHIP-L dataset derived from real chip designs. ICCAD13 comprises the simpler ICCAD13-S and the more complex ICCAD13-L, both with layout size $2.048\,\mu\mathrm{m} \times 2.048\,\mu\mathrm{m}$. CHIP-L is constructed from three popular open-source designs, PicoRV32 (Wolf, 2015), NoCRouter (Galimberti et al., 2018), and DarkRISCV (Samsoniuk, 2018), by generating GDSII layouts using commercial layout generation tool and open-source FreePDK45 (Stine et al., 2007) design kit and extracting the first metal layer M1. To facilitate reproducibility and future research, all CHIP-L test cases will be publicly released for community evaluation.

Evaluation Metrics. We use L_2, Process Variation Band (PVB), Edge Placement Error (EPE), mask shots, and Turn-Around Time (TAT) to evaluate performance in terms of lithographic fidelity, process robustness, mask complexity, and computational efficiency (Zheng et al., 2023). L_2 measures the pixel-wise discrepancy between the wafer pattern and the target layout under nominal

Table 1: Comparison with state-of-the-art ILT / OPC methods on ICCAD13.

Method	ICCAD13-S					ICCAD13-L				
	L_2 ↓	PVB ↓	EPE ↓	*#shots* ↓	TAT ↓	L_2 ↓	PVB ↓	EPE ↓	*#shots* ↓	TAT ↓
Pixel-based ILT Methods										
Neural-ILT (Jiang et al., 2020)	35,524	46,819	6.51	845.1	11.20	67,547	98,490	4.15	1,506.4	14.37
FuILT (Yin et al., 2024)	44,119	47,120	7.46	391.6	9.14	85,192	110,381	5.84	1,005.1	9.28
LVM-MO (Wu et al., 2025b)	26,405	46,078	2.56	954.2	**0.37**	50,871	98,965	1.86	1,035.2	**0.49**
CurvyILT (Yang & Ren, 2026)	28,001	44,522	2.59	3,637.4	8.03	57,886	95,151	2.14	6,112.5	8.20
Manhattan-based OPC Methods										
DiffOPC (Chen et al., 2024b)	28,280	48,217	2.26	**106.3**	8.42	74,322	116,319	4.48	**335.8**	102.29
TokMan-coarse (Wu et al., 2025a)	23,728	50,745	3.17	225.4	1.12	47,028	129,195	2.10	480.6	1.74
TokMan-dense (Wu et al., 2025a)	21,867	53,319	2.16	288.7	1.89	40,340	138,947	1.62	642.3	2.47
Ours	**19,849**	**44,200**	**1.90**	257.9	9.73	**39,365**	**90,016**	**1.56**	487.8	10.46

process conditions (Poonawala & Milanfar, 2007), while PVB measures the range of variation in printed patterns across different process conditions (Su et al., 2016; Shen et al., 2011).

EPE measures edge placement accuracy by evaluating the displacement between target and printed contours (Wu et al., 2025a). We place measurement points along the target contour and compute their mean distances (in nm) to the wafer contour, accounting for both inward and outward edge shifts. Unlike threshold-based EPE violation counts, this continuous metric captures edge displacement below the violation threshold and complements the pixel-wise evaluation provided by L_2.

Mask shots measure the manufacturability and writing efficiency of the generated mask (Chua et al., 2011). In Electron Beam Lithography (EBL), the final mask geometry is decomposed into rectangular write units, each corresponding to one shot. A smaller shot count generally indicates lower writing effort and higher fabrication efficiency. We compute this metric from the final decomposed mask geometry, independently of the number of primitives used during optimization.

TAT measures the computation time, in seconds, required to generate the final mask (Wu et al., 2025a). For optimization-based methods, we report the runtime of the full optimization pipeline. For learning-based methods, we report the inference time together with any subsequent refinement actually applied. For Manhattan Assembly, timing covers the entire optimization pipeline. Formal metric definitions and evaluation details appear in Supplementary Material B.

4.2 Comparison with State-of-the-Art Methods

Comparison on ICCAD13. As shown in Table 1, Manhattan Assembly achieves the lowest L_2, PVB, and EPE across both ICCAD13 subsets, using fewer shots than TokMan-dense and all pixel-based ILT baselines. This demonstrates the effectiveness of adaptive primitive allocation in balancing correction capacity and mask complexity under Manhattan constraints. DiffOPC and TokMan-coarse use fewer shots but incur larger printing errors, reflecting a clear quality-complexity trade-off. Qualitative results are provided in Supplementary Material C.

Scaling to Large Layouts. Table 2 evaluates how performance scales with layout size, up to full chips. The benchmark consists of three subsets of increasing scale, each containing 10 cases. CHIP-L3 contains full-chip layouts of $61.4\,\mu\text{m} \times 59.3\,\mu\text{m}$, from which CHIP-L1 ($4.0\,\mu\text{m} \times 4.0\,\mu\text{m}$) and CHIP-L2 ($8.2\,\mu\text{m} \times 8.1\,\mu\text{m}$) are locally cropped while preserving the original geometric scale. Since these subsets are evaluated with TorchLitho rather than the ICCAD13 simulator, PVB values follow a different trend and are not directly comparable to those on ICCAD13. For a fair comparison, the learning-based baselines TokMan and LVM-MO are retrained on the dataset from which CHIP-L is drawn, following the configurations in their original papers, and evaluated on the held-out CHIP-L test cases. Across all three subsets, Manhattan Assembly achieves the lowest L_2, PVB, EPE, and shot count. The only exception is turnaround time, where LVM-MO is the fastest. As a learning-based method, LVM-MO produces masks in a single forward pass, but its outputs offer no mechanism for further per-case refinement, leaving it with lower accuracy and higher mask complexity. In contrast, our primitive-based representation is optimized directly against the lithography model for each layout, trading additional runtime for consistently higher fidelity. The EPE gain

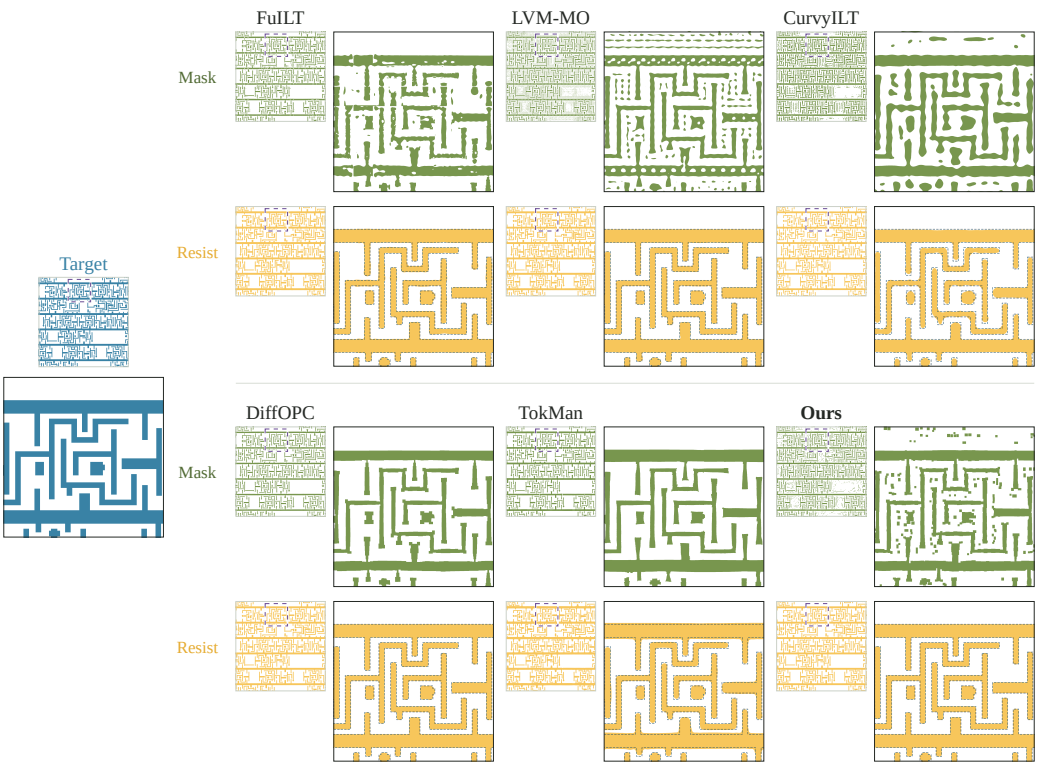

Figure 5: Qualitative comparison of masks and resists on partial full-chip layouts.

Table 2: Large-scale comparison with state-of-the-art ILT / OPC methods on CHIP-L.

Method	CHIP-L1					CHIP-L2					CHIP-L3				
	L_2 ↓	PVB ↓	EPE ↓	*#shots* ↓	TAT ↓	L_2 ↓	PVB ↓	EPE ↓	*#shots* ↓	TAT ↓	$L_{2\,(\times 10^6)}$ ↓	$\text{PVB}_{(\times 10^6)}$ ↓	EPE ↓	$\textit{\#shots}_{(\times 10^6)}$ ↓	TAT ↓
Pixel-based ILT Methods															
FuILT	505,498	449,213	7.15	26,843.6	113.04	1,357,315	1,484,051	4.09	103,967.9	426.25	87.426	104.674	4.49	7.164	8,736.47
LVM-MO	385,022	420,644	4.26	62,967.5	**0.47**	1,315,358	1,405,514	3.65	248,994.4	**1.08**	86.976	93.184	3.89	15.884	**86.43**
CurvyILT	517,563	427,360	4.53	25,999.1	9.82	1,772,387	1,426,127	4.45	102,102.0	39.72	124.963	98.741	4.04	6.982	3,184.62
Manhattan-based OPC Methods															
DiffOPC†	573,005	516,344	8.56	9,571.7	117.50	2,432,707	2,171,214	8.68	38,384.2	197.78	158.936	150.287	9.14	2.641	7,650.88
TokMan†	485,101	366,535	5.21	3,564.5	36.48	2,424,631	1,314,295	5.22	17,973.6	144.25	169.824	90.532	5.47	1.248	6,318.56
Ours	**183,584**	**356,994**	**3.13**	**1,687.8**	142.81	**406,191**	**1,293,438**	**3.47**	**13,866.5**	200.20	**57.192**	**82.337**	**3.54**	**0.913**	7,427.01

† Results obtained using a divide-and-conquer strategy.

is most pronounced over DiffOPC and TokMan, whose divide-and-conquer implementations stitch independently optimized tiles; we attribute this gap to our globally shared primitives, which reduce stitching artifacts at tile boundaries. Figure 5 compares the resulting masks and resist patterns, where this difference is visually evident.

4.3 Mask Rule Check

We evaluate mask manufacturability using the commercial tool, comparing against CurvyILT and DiffOPC, the representative baselines in their respective paradigms. Specifically, we evaluate violations across three critical MRC constraints: minimum width (20 nm), minimum spacing (20 nm), and notch size (10 nm). As shown in Figure 6, Manhattan Assembly achieves zero violations across all three constraints on both ICCAD13-S and ICCAD13-L. These results support the effectiveness of primitive-level geometric constraints in substantially reducing mask-level violations, thereby improving the rule compliance of the generated masks.

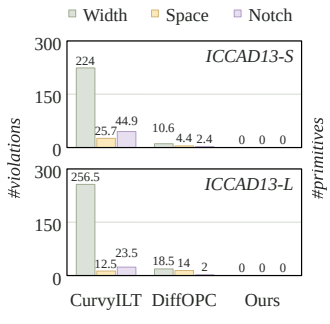

Figure 6: MRC violations.

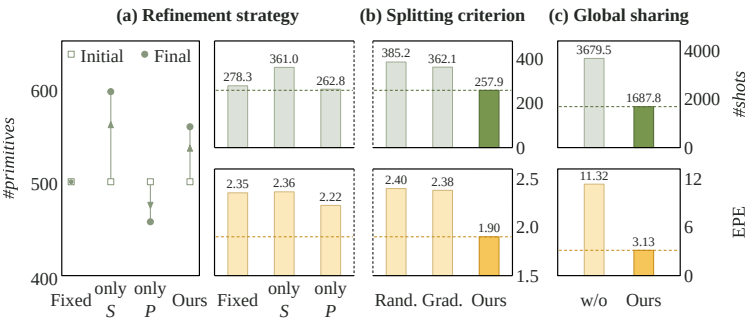

Figure 7: Ablation studies.

4.4 Ablation Studies

Refinement Strategy. Figure 7(a) evaluates the effect of splitting (S) and pruning (P) on ICCAD13-S, with all variants initialized with 500 primitives. Splitting introduces local degrees of freedom for geometric correction, whereas pruning removes redundant primitives to control mask complexity. Combining both operations reduces EPE and mask shots compared with Fixed and either operation alone, supporting the complementary roles of capacity allocation and redundancy removal. Although pruning alone yields the fewest primitives, the joint strategy achieves fewer mask shots with lower EPE, indicating that effective primitive allocation matters more than simply minimizing the primitive count.

Splitting Criterion. On ICCAD13-S, we compare our gradient-discrepancy criterion m_i^{diff} against random splitting (Rand.) and parent-gradient magnitude (Grad.) under identical initialization, split schedules, and budgets. As shown in Figure 7(b), our criterion achieves lower EPE with fewer mask shots than both alternatives. Random selection does not exploit local gradient structure, while gradient cancellation within a parent primitive can obscure opposing correction demands. Evaluating the gradients of the two sub-rectangles created by a temporary split reveals these local demands without changing the current mask geometry.

Global Primitive Sharing. On CHIP-L1, global primitive sharing reduces full-image EPE while also reducing mask shots compared with independent window optimization, as shown in Figure 7(c). Independent window optimization can lead to inconsistent corrections across partition boundaries. Sharing primitive parameters and accumulating gradients across windows coordinates these corrections, mitigating stitching mismatches and unnecessary mask fragmentation. These results support partitioning computation while maintaining a globally shared geometric representation, preserving correction accuracy and mask compactness across windows.

5 Discussion

Limitation. This work focuses on Manhattan-constrained mask optimization, where axis-aligned primitives provide structural control. Extending the framework to curvilinear masks would require corresponding primitive representations and adaptation rules. Our experiments cover benchmark and full-chip layouts under lithography simulation. Further evaluation across process settings and integration of process-specific mask rules would broaden the framework's practical applicability.

Conclusion. We presented Manhattan Assembly, a primitive-based framework for continuous mask optimization that preserves Manhattan geometry by construction. Differentiable rectangle rendering connects geometric parameters to lithographic objectives, while lithography-guided adaptive refinement adjusts primitive allocation according to local correction needs. Global primitive sharing coordinates optimization across windows and enables full-chip scaling. Experiments on ICCAD13 and CHIP-L demonstrate a favorable balance between printing fidelity and mask complexity across the evaluated layout scales. Together, these results establish adaptive primitive assembly as an effective approach to mask optimization, combining structural control, geometric flexibility, and full-chip scalability within a unified framework.

REFERENCES

Shayak Banerjee, Zhuo Li, and Sani R. Nassif. ICCAD-2013 CAD contest in mask optimization and benchmark suite. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 271–274, 2013. doi: 10.1109/iccad.2013.6691131.

Guojin Chen, Wanli Chen, Qi Sun, Yuzhe Ma, Haoyu Yang, and Bei Yu. DAMO: Deep Agile Mask Optimization for Full-Chip Scale. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(9):3118–3131, 2022. doi: 10.1109/tcad.2021.3116511.

Guojin Chen, Hao Geng, Bei Yu, and David Z. Pan. Open-source differentiable lithography imaging framework. In *DTCO and Computational Patterning III*, volume 12954, pp. 129540J, 2024a. doi: 10.1117/12.3009980.

Guojin Chen, Haoyu Yang, Haoxing Mark Ren, Bei Yu, and David Z. Pan. Differentiable Edge-based OPC. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–9, 2024b. doi: 10.1145/3676536.3676764.

Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. BSP-Net: Generating Compact Meshes via Binary Space Partitioning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 45–54, 2020. doi: 10.1109/cvpr42600.2020.00012.

Gek Soon Chua, Wei Long Wang, Byoung IL Choi, Yi Zou, Cyrus Tabery, Ingo Bork, Tam Nguyen, and Aki Fujimura. Optimization of mask shot count using MB-MDP and lithography simulation. In *Photomask Technology 2011*, volume 8166, pp. 816632, 2011. doi: 10.1117/12.897037.

Nicolas B. Cobb and Avideh Zakhor. Fast, low-complexity mask design. In *Optical/Laser Microlithography VIII*, volume 2440, pp. 313–327, 1995. doi: 10.1117/12.209263.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. LightGaussian: Unbounded 3D Gaussian Compression with 15x Reduction and 200+ FPS. In *Advances in Neural Information Processing Systems*, volume 37, pp. 140138–140158, 2024. doi: 10.52202/079017-4447.

Andrea Galimberti, Filippo Testa, and Alberto Zeni. NoCRouter: RTL Router Design in SystemVerilog. GitHub repository, 2018. URL `https://github.com/agalimberti/NoCRouter`. Accessed: September 24, 2026.

Chaojun Huang, Xu Ma, Shengen Zhang, Mu Lin, Néstor Porras-Díaz, and Gonzalo R. Arce. Block-based inverse lithography technology with adaptive level-set algorithm. *Optics & Laser Technology*, 182:112211, 2025. doi: 10.1016/j.optlastec.2024.112211.

Bentian Jiang, Lixin Liu, Yuzhe Ma, Hang Zhang, Bei Yu, and Evangeline F. Y. Young. Neural-ILT: Migrating ILT to Neural Networks for Mask Printability and Complexity Co-optimization. In *Proceedings of the 39th International Conference on Computer-Aided Design*, pp. 1–9, 2020. doi: 10.1145/3400302.3415704.

R. Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy J. Mitra, and Daniel Ritchie. ShapeAssembly: Learning to Generate Programs for 3D Shape Structure Synthesis. *ACM Transactions on Graphics*, 39(6):234:1–234:20, 2020. doi: 10.1145/3414685.3417812.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. doi: 10.1145/3592433.

Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics*, 39(6):193:1–193:15, 2020. doi: 10.1145/3414685.3417871.

Y. Liu and A. Zakhor. Binary and phase shifting mask design for optical lithography. *IEEE Transactions on Semiconductor Manufacturing*, 5(2):138–152, 1992. doi: 10.1109/66.136275.

Linjie Lyu, Ayush Tewari, Jianchun Chen, Thomas Leimkuehler, and Christian Theobalt. Faster 3D Gaussian Splatting Convergence via Structure-Aware Densification. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–10, 2026. doi: 10.1145/3799902.3811212.

Xing-Yu Ma and Shaogang Hao. Inverse lithography physics-informed deep neural level set for mask optimization. *Applied Optics*, 62(33):8769–8779, 2023. doi: 10.1364/ao.503332.

Xu Ma and Gonzalo R. Arce. Generalized inverse lithography methods for phase-shifting mask design. *Optics Express*, 15(23):15066–15079, 2007. doi: 10.1364/oe.15.015066.

Xu Ma, Zhiyang Song, Yanqiu Li, and Gonzalo R. Arce. Block-based mask optimization for optical lithography. *Applied Optics*, 52(14):3351–3363, 2013. doi: 10.1364/ao.52.003351.

David Z. Pan, Bei Yu, and Jhih-Rong Gao. Design for Manufacturing With Emerging Nanolithography. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(10): 1453–1472, 2013. doi: 10.1109/tcad.2013.2276751.

Linyong (Leo) Pang. Inverse lithography technology: 30 years from concept to practical, full-chip reality. *Journal of Micro/Nanopatterning, Materials, and Metrology*, 20(03):030901, 2021. doi: 10.1117/1.jmm.20.3.030901.

Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics Revisited: Learning 3D Shape Parsing Beyond Cuboids. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10344–10353, 2019. doi: 10.1109/cvpr.2019.01059.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32, pp. 8024–8035, 2019. URL https://papers.neurips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Amyn Poonawala and Peyman Milanfar. Mask Design for Optical Microlithography—An Inverse Imaging Problem. *IEEE Transactions on Image Processing*, 16(3):774–788, 2007. doi: 10.1109/TIP.2006.891332.

Marcelo Samsoniuk. DarkRISCV. GitHub repository, 2018. URL https://github.com/darklife/darkriscv. Accessed: September 24, 2026.

Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. CSGNet: Neural Shape Parser for Constructive Solid Geometry. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5515–5523, 2018. doi: 10.1109/cvpr.2018.00578.

Yijiang Shen, Ngai Wong, and Edmund Y. Lam. Level-set-based inverse lithography for photomask synthesis. *Optics Express*, 17(26):23690–23701, 2009. doi: 10.1364/oe.17.023690.

Yijiang Shen, Ningning Jia, Ngai Wong, and Edmund Y. Lam. Robust level-set-based inverse lithography. *Optics Express*, 19(6):5511–5521, 2011. doi: 10.1364/oe.19.005511.

Chris Spence. Full-chip lithography simulation and design analysis: how OPC is changing IC design. In *Emerging Lithographic Technologies IX*, volume 5751, pp. 1–14, 2005. doi: 10.1117/12.608020.

James E. Stine, Ivan Castellanos, Michael Wood, Jeff Henson, Fred Love, W. Rhett Davis, Paul D. Franzon, Michael Bucher, Sunil Basavarajaiah, Julie Oh, and Ravi Jenkal. FreePDK: An Open-Source Variation-Aware Design Kit. In *2007 IEEE International Conference on Microelectronic Systems Education (MSE'07)*, pp. 173–174, 2007. doi: 10.1109/MSE.2007.44.

Yu-Hsuan Su, Yu-Chen Huang, Liang-Chun Tsai, Yao-Wen Chang, and Shayak Banerjee. Fast Lithographic Mask Optimization Considering Process Variation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(8):1345–1357, 2016. doi: 10.1109/tcad.2015.2514082.

Shuyuan Sun, Fan Yang, Bei Yu, Li Shang, and Xuan Zeng. Efficient ILT via Multi-level Lithography Simulation. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2023. doi: 10.1109/dac56929.2023.10247704.

Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning Shape Abstractions by Assembling Volumetric Primitives. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2635–2643, 2017. doi: 10.1109/cvpr.2017.160.

Haoyu Wang, Yiming Xu, Yu Feng, Hao Geng, and Shisheng Xiong. HiFiMO: A High-Resolution Flexible Mask Optimization Framework with Diffusion Model. *Journal of Micro/Nanopatterning, Materials, and Metrology*, 24(3):033202, 2025a. doi: 10.1117/1.JMM.24.3.033202.

Peihao Wang, Yuehao Wang, Dilin Wang, Sreyas Mohan, Zhiwen Fan, Lemeng Wu, Ruisi Cai, Yu-Ying Yeh, Zhangyang Wang, Qiang Liu, and Rakesh Ranjan. Steepest Descent Density Control for Compact 3D Gaussian Splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26663–26672, 2025b. doi: 10.1109/cvpr52734.2025.02483.

Neil H. E. Weste and David Money Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley, 4 edition, 2011.

Claire Xenia Wolf. PicoRV32: A Size-Optimized RISC-V CPU. GitHub repository, 2015. URL `https://github.com/YosysHQ/picorv32`. Accessed: September 24, 2026.

Yiwen Wu, Yuyang Chen, Ye Xia, Yao Zhao, Jingya Wang, Xuming He, Hao Geng, and Jingyi Yu. TokMan: Tokenize Manhattan Mask Optimization for Inverse Lithography. In *Advances in Neural Information Processing Systems*, volume 38, pp. 60274–60299, 2025a. URL `https://openreview.net/forum?id=hsf4gi5iEN`.

Yiwen Wu, Yuyang Chen, Shuo Yin, Nan Wang, Tao Wu, Xuming He, Hao Geng, and Jingyi Yu. LVM-MO: A Large Vision Model Pioneer on Full-Chip Mask Optimization. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pp. 1–7, 2025b. doi: 10.1109/dac63849.2025.11132784.

Haoyu Yang and Haoxing Ren. GPU-Accelerated Inverse Lithography Towards High Quality Curvy Mask Generation. In *Proceedings of the 2025 International Symposium on Physical Design*, pp. 42–50, 2025. doi: 10.1145/3698364.3705343.

Haoyu Yang and Haoxing Ren. CurvyILT 2.0: GPU-Accelerated Inverse Lithography Targeting Full-Chip Optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 45(10):4828–4840, 2026. doi: 10.1109/TCAD.2026.3661842.

Haoyu Yang, Shuhe Li, Yuzhe Ma, Bei Yu, and Evangeline F. Y. Young. GAN-OPC: Mask Optimization with Lithography-guided Generative Adversarial Nets. In *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2018. doi: 10.1109/dac.2018.8465816.

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. AbsGS: Recovering Fine Details in 3D Gaussian Splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 1053–1061, 2024. doi: 10.1145/3664647.3681361.

Shuo Yin, Wenqian Zhao, Li Xie, Hong Chen, Yuzhe Ma, Tsung-Yi Ho, and Bei Yu. FuILT: Full Chip ILT System With Boundary Healing. In *Proceedings of the 2024 International Symposium on Physical Design*, pp. 13–20, 2024. doi: 10.1145/3626184.3633315.

Fenggen Yu, Zhiqin Chen, Manyi Li, Aditya Sanghi, Hooman Shayani, Ali Mahdavi-Amiri, and Hao Zhang. CAPRI-Net: Learning Compact CAD Shapes with Adaptive Primitive Assembly. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11768–11778, 2022. doi: 10.1109/cvpr52688.2022.01147.

Jiahui Zhang, Fangneng Zhan, Muyu Xu, Shijian Lu, and Eric Xing. FreGS: 3D Gaussian Splatting with Progressive Frequency Regularization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21424–21433, 2024a. doi: 10.1109/cvpr52733.2024.02024.

Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-GS: Density Control with Pixel-Aware Gradient for 3D Gaussian Splatting. In *Computer Vision – ECCV 2024*, pp. 326–342, 2024b. doi: 10.1007/978-3-031-72655-2_19.

Su Zheng, Haoyu Yang, Binwu Zhu, Bei Yu, and Martin Wong. LithoBench: Benchmarking AI Computational Lithography for Semiconductor Manufacturing. In *Advances in Neural Information Processing Systems*, volume 36, pp. 30243–30254, 2023. doi: 10.52202/075280-1315.

Yihan Zheng, Lihui Liu, Yanqiu Li, Weichen Huang, and He Yang. Demand-driven dynamic control points insertion for high-fidelity and efficient curvilinear optical proximity correction. *Optics Express*, 34(5):7261–7277, 2026. doi: 10.1364/oe.582323.

Binwu Zhu, Su Zheng, Ziyang Yu, Guojin Chen, Yuzhe Ma, Fan Yang, Bei Yu, and Martin D. F. Wong. L2O-ILT: Learning to Optimize Inverse Lithography Techniques. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(3):944–955, 2024. doi: 10.1109/tcad.2023.3323164.

AI USE STATEMENT

Generative AI tools were used to improve the clarity, readability, and language of author-written text, and to assist with literature retrieval and discovery. All retrieved references and AI-assisted edits were independently reviewed and verified by the authors. The authors take full responsibility for the final content of this work.