TokMan:Tokenize Manhattan Mask Optimization for Inverse Lithography

Yiwen Wu[*1] **Yuyang Chen**[*1] **Ye Xia**[1] **Yao Zhao**[1] **Jingya Wang**[1]

Xuming He[1] **Hao Geng**[†1] **Jingyi Yu**[†1]

[1]ShanghaiTech University
[*] *Equal contribution*
[†] *Corresponding author*

Abstract

Manhattan representations, defined by axis-aligned, orthogonal structures, are widely used in vision, robotics, and semiconductor design for their geometric regularity and algorithmic simplicity. In integrated circuit (IC) design, Manhattan geometry is key for routing, design rule checking, and lithographic manufacturability. However, as feature sizes shrink, optical system distortions lead to inconsistency between intended layout and printed wafer. Although Inverse Lithography Technology(ILT) is proposed to compensates these effects, learning-based ILT methods, while achieving high simulation fidelity, often generate curvilinear masks on continuous pixel grids, violating Manhattan constraints. Therefore, we propose TokMan, the first framework to formulate mask optimization as a discrete, structure-aware sequence modeling task. Our method leverages a Diffusion Transformer to tokenize layouts into discrete geometric primitives with polygon-wise dependencies and denoise Manhattan-aligned point sequences corrupted by optical proximity effects, while ensuring binary, manufacturable masks. Trained with self-supervised lithographic feedback through differentiable simulation and refined with ILT post-processing, TokMan achieves state-of-the-art fidelity, runtime efficiency, and strict manufacturing compliance on a large-scale dataset of IC layouts.

1 Introduction

Manhattan representations—characterized by axis-aligned, orthogonal structures—play a foundational role in computer vision and robotics by simplifying the geometric complexity of structured environments. These representations exploit the prevalence of right angles in man-made settings, enabling efficient solutions to fundamental tasks such as vanishing point detection [1], camera localization [2], SLAM (simultaneous localization and mapping) [3], and 3D scene reconstruction [4]. By reducing spatial ambiguity and imposing global regularity, Manhattan models facilitate compact encodings and algorithmic tractability, particularly in indoor and urban domains. Beyond perception, this abstraction is also central to architecture and procedural generation, where structured geometry supports scalable and interpretable design.

This same geometric prior is deeply embedded in the semiconductor industry, where integrated circuit (IC) layouts adhere to Manhattan constraints for both algorithmic and manufacturing efficiency. Modern IC designs consist of features such as wires and vias in layout and mask, that are aligned to orthogonal grids, simplifying placement, routing, design rule checking(DRC) [5], and verification workflows. Critically, Manhattan mask are favored in photolithography due to their compatibility with optical projection systems and EDA toolchains [6].

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

The axis-aligned structure enhances manufacturability, particularly as process nodes scale into the nanometer regime.

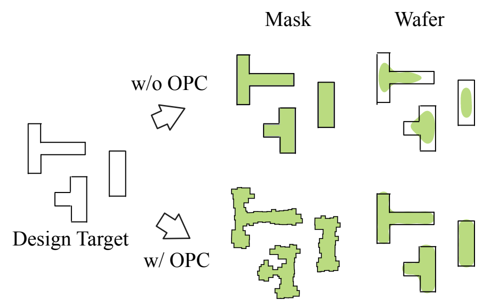

Figure 1: Pattern distortion caused by optical system. It requires mask optimization to be correctly fabricated on the silicon wafer.

However, as lithography technique nodes shrink, Manhattan layouts become more susceptible to pattern-dependent deviations caused by optical phenomena such as light diffraction and interference [7]. These distortions challenge the fidelity of pattern transfer, motivating the adoption of inverse lithography technology (ILT). ILT formulates the task as an inverse imaging problem: given a desired target layout pattern, the objective is to compute a modified design, named mask that reproduces the intended pattern after optical projection. The whole process is called **mask optimization**. Recent advances in both classical mask optimization [8] and deep-learning approaches [9, 10, 11, 12] have yielded powerful ILT solvers. These approaches relax the binary mask into a continuous representation, allowing pixel-wise gradient descent optimization. Despite their outstanding performances, reconverting the continuous mask back to a binary representation undo the benefits gained during the pixel-level optimization. As a result, the final binary mask often deviates from the intended objective and remains suboptimal. Moreover, masks reproduced by these methods frequently contain curvilinear or irregular features that violate the Manhattan constraints essential for manufacturability and increase write-time complexity, thereby negatively impacting time-to-market efficiency.

In this paper, we propose **TokMan**, the first framework to apply a sequence modeling technique approach popularized by large language models(LLMs) [13, 14, 15], to the task of lithography mask generation under strict geometry constraints [16]. Rather than treat Manhattan compliance as a post-hoc constraint, TokMan integrates Manhattan representations inherently throughout the optimization process by formulating mask correction as a sequence tokenization problem over a discrete vocabulary of axis-aligned geometric primitives. This approach aligns with the broader "**tokenize everything**" trend in machine learning, where motion [17], geometry [18], and structure [19] are represented as symbolic sequences. Inspired by works such as MeshGPT [20] for triangle meshes and EgoEgo [21] for human motion reconstruction, our method employs a Diffusion Transformer architecture [22] to generate corrected masks in a tokenized Manhattan domain. In this formulation, token-based modeling is not merely a convenient abstraction—it is a fundamentally aligned representation for Manhattan-constrained layout synthesis. By discretizing geometry into axis-aligned tokens, the model operates natively within the design space dictated by manufacturing constraints, enabling not only structurally faithful mask generation but also scalable, context-aware reasoning across complex polygonal configurations. This intrinsic compatibility renders tokenization a conceptually elegant and practically powerful paradigm for Manhattan mask generation.

TokMan employs a conditional diffusion process that reinterprets the forward lithography as a structured noise addition-where diffraction and interference degrade the mask into a distorted aerial image-allowing ILT process to be cast as a denoising problem [23]. The Diffusion Transformer simultaneously learns to model this inverse process while capturing the spatial dependencies between Manhattan primitives. Training is conducted in a self-supervised fashion, using a large-scale dataset derived from open-source IC libraries [24] that span a wide range of real-world routing and standard cell patterns. For each layout, the model learns to predict token-level geometric corrections conditioned on the full context. Predicted token sequences are rendered into continuous mask representations via `nvdiffrast` [25], and their lithographic printability is evaluated through a differentiable imaging pipeline. The simulated aerial image is compared against the target pattern to compute a reconstruction loss, enabling end-to-end optimization without ground-truth mask annotations. A lightweight ILT refinement is optionally applied after generation to enhance line-edge fidelity while maintaining strict Manhattan compliance. Empirically, our method achieves superior performance over existing baselines in terms of pattern fidelity, runtime efficiency, and manufacturing compliance, offering a scalable path toward practical, learning-based inverse lithography approach under strict manufacturing constraints.

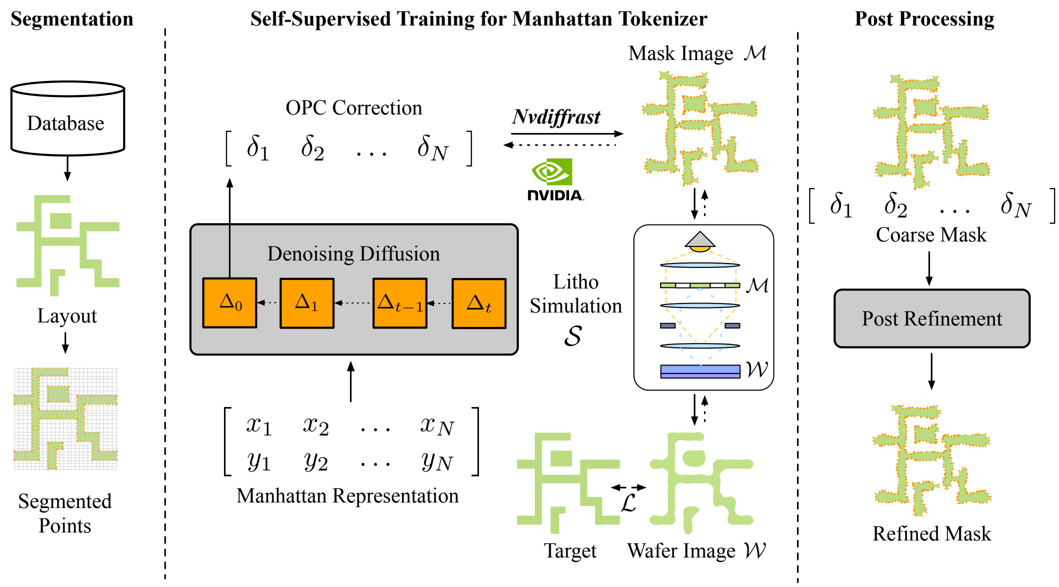

Figure 2: **Overview of TokMan.** The input layout is first segmented into a Manhattan representation via our litho-aware segmentation algorithm. A denoising diffusion model then is trained to tokenize and predict OPC corrections using segmented Manhattan points. These corrected points are rendered to mask image $\mathcal{M}$ by *Nvdiffrast* and passed through a lithography simulator to produce a simulated wafer image $\mathcal{W}$, where loss is used to supervise the generation of OPC correction. Finally, a post process is applied to enhance fidelity of coarse mask after generation.

2 Related Work

Manhattan Representation. The Manhattan representation, or Manhattan world assumption, posits that scenes are predominantly composed of structures aligned along three orthogonal directions. This geometric prior has been extensively utilized in computer vision and machine learning to simplify complex spatial understanding tasks [26]. Introduced by Coughlan and Yuille [27], this assumption enables robust camera calibration and 3D inference using edge orientations and vanishing points. Building upon this, Guo et al. [28] integrated planar constraints derived from the Manhattan assumption into neural implicit representations for 3D scene reconstruction, achieving improved quality in low-texture indoor environments. Additionally, Wakai et al. [29] leveraged the Manhattan world prior for single-image camera calibration, utilizing heatmap regression to estimate vanishing points and camera orientation, demonstrating robustness in challenging scenarios with fisheye distortions. These studies underscore the versatility and effectiveness of the Manhattan world assumption as a structural prior in various computer vision and machine learning applications.

Tokenization with Transformer. Tokenization, originally developed in natural language processing (NLP) to segment text into discrete, meaningful units, has become a new trend in modern machine learning. With the rise of transformer architectures like BERT [30] and GPT [31], its importance has expanded across modalities, enabling unified and scalable representation of diverse data types [32, 33]. In 3D scene understanding, Chen et al. [34] use the Segment Anything Model (SAM) to tokenize 3D point clouds by aligning 2D masks, enhancing semantic segmentation through region-level learning. MeshGPT [20] formulates triangle mesh generation as a sequence modeling task, leveraging geometric tokens for compact and autoregressive 3D synthesis. PointMamba [35] introduces an efficient state space model that imposes token order via space-filling curves, enabling scalable point cloud analysis. In egocentric motion prediction, Chi et al. [21] tokenize full-body trajectories through a conditional transformer-based diffusion framework in the time domain. These developments highlight tokenization's pivotal role in enabling transformers to process complex, high-dimensional data structures, establishing it as a universal mechanism for unifying representation and computation in contemporary machine learning.

Optical Proximity Correction. Optical Proximity Correction (OPC) plays a central role in lithography, aiming to compensate for systematic pattern distortions caused by light diffraction, optical proximity effects, and process non-idealities. As feature sizes approach the resolution limits of lithographic systems, the printed patterns often deviate significantly from the intended layout, making OPC essential for ensuring pattern fidelity. Traditional rule-based and model-based OPC rely on calibrated optical models and heuristic correction strategies, where geometric features are modified iteratively to align simulated contours with the target layout. These methods typically require expensive computational simulations and manual tuning. Recent academic work leverages deep neural networks and GPU acceleration to enhance both runtime and correction fidelity. For instance, long-standing level-set algorithms [36] have been migrated to GPUs. Generative models such as GANs [9] and task-specific neural networks [10, 11, 12] are applied for pixel-level mask synthesis and optimization. Nonetheless, many of these learning-based methods lack explicit manufacturability constraints, limiting their applicability in real-world industrial settings.

3 Preliminaries

Forward Simulation. Lithography is a core technology in semiconductor manufacturing, enabling the transfer of circuit patterns from photomasks onto silicon wafers using light exposure [37]. The forward lithography process models how a given mask pattern produces a printed image on the wafer, governed primarily by optical diffraction and resist interactions [38, 39]. Mathematically, the aerial image $I(x, y)$ formed on the wafer plane can be modeled as coherent or partially coherent imaging of the mask $M(x, y)$ through an optical system:

$$I(x, y) = |\mathcal{F}^{-1}\{H(f_x, f_y) \cdot \mathcal{F}[M(x, y)]\}|^2, \tag{1}$$

where $\mathcal{F}$ denotes the Fourier transform, $H(f_x, f_y)$ is the optical transfer function (OTF) of the lithography system, and $M(x, y)$ is the binary mask pattern. Post-processing steps, such as resist development and etching, are often approximated as thresholding or convolutional smoothing operations on $I(x, y)$.

Inverse Lithography Technology (ILT). The inverse lithography process [40] aims to determine an optimal mask pattern $M(x, y)$ that produces a target pattern $T(x, y)$ on the wafer, effectively solving an inverse problem [41, 42]. This can be formulated as minimizing the discrepancy between the simulated wafer image and the desired layout:

$$\min_{M(x,y)} \mathcal{L}(I(M), T) + \lambda \cdot \mathcal{R}(M), \tag{2}$$

where $\mathcal{L}$ is a loss function measuring pattern fidelity (e.g., L_2 loss, edge mismatch, or critical dimension error), and $\mathcal{R}$ is a regularization term that encourages mask manufacturability and sparsity (e.g., enforcing Manhattan geometry or minimizing sub-resolution features). λ balances fidelity and regularity.

Recent ILT frameworks utilize differentiable lithography simulators to enable gradient-based optimization [43, 44]. Notably, self-supervised learning approaches avoid requiring explicit ground-truth masks by computing the loss between simulated images and target layouts directly, allowing end-to-end optimization:

$$\mathcal{L}_{self} = \|\mathcal{S}(M) - T\|^2, \tag{3}$$

where $\mathcal{S}(M)$ denotes the simulated print image from mask M via forward lithography. Such frameworks facilitate mask learning driven by lithographic feedback, enhancing printability and robustness under process variations.

4 Methods

This section presents our proposed TokMan framework. We begin by describing the layout segmentation strategy, which converts Manhattan circuit geometries into structured representations suitable for learning. We then detail the use of a Diffusion Transformer for generating OPC corrections in token space, followed by the self-supervised training procedure enabled by a differentiable Manhattan renderer and lithography forward simulation. Finally, we introduce a lightweight ILT-based post-processing step to ensure high-fidelity manufacturability at deployment.

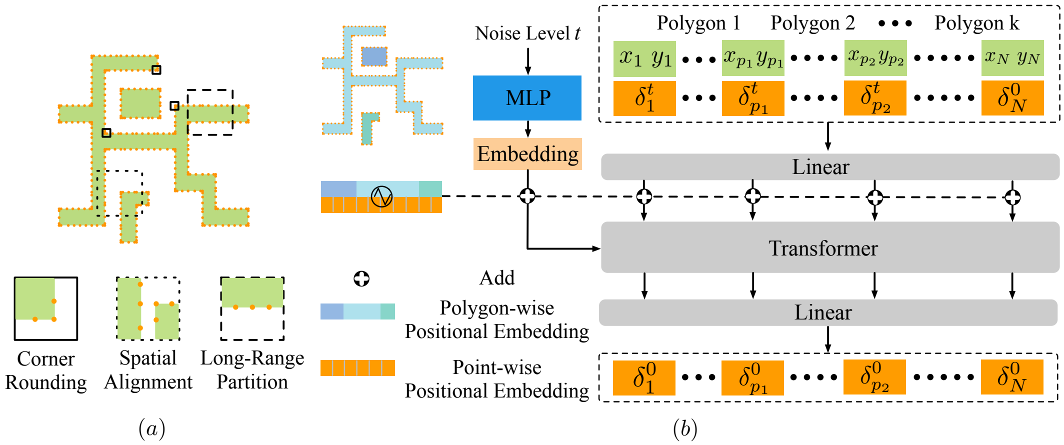

Figure 3: (a) Segmentation Algorithm Visualization. (b) Model Architecture of denoising network in a single step of the reverse diffusion process.

4.1 Segmetation Algorithm

Augmentation for Spatial Perception. Segmentation serves as the foundational step in our method, transforming raw geometric input into a structured form amenable to learning. Starting from sparse polygonal contours in the Manhattan layout, we subdivide each polygon edge into densely sampled, axis-aligned line segments. This densification is essential not only for capturing the full resolution of layout boundaries but also for constructing a high-resolution optimization space tailored for Manhattan representation. These segments produce a sequence of discrete points $\{(x_i, y_i)\}_{i=1}^N$, precisely aligned to horizontal or vertical directions, which serve as the input to our model. By converting sparse geometric data into a dense and grid-aligned format, we impose geometric regularity and uniform sampling across the layout. This regularized structure allows the Diffusion Transformer to operate over a consistent and interpretable representation, enabling it to model correction patterns with both local precision and global coherence. By ensuring consistency and interpretability in the input representation, this step enables the network to reason over fine-grained layout details while maintaining strict alignment with Manhattan geometry.

Lithography-aware Points Segmentation. Beyond structural regularization, our segmentation algorithm is designed to capture key lithographic phenomena and integrate manufacturability constraints, particularly corner rounding and feature coherence under optical proximity effects. First, to mitigate **corner rounding**, we enforce high-resolution segment splits at all detected corners, ensuring that the discretization granularity at curvature-critical regions is maximized. This uniform corner-based refinement enables accurate modeling of sharp edges where lithographic deformation is most pronounced. Second, to enhance connectivity and contextual consistency among **parallel edges**, we implement spatial alignment casting: for any given edge, we detect all parallel edges within a configurable lithography-influence zone and identify their nearby corner vertices. These corner positions are then transposed across space and aligned onto the current edge as candidate segmentation anchors. This facilitates implicit edge-to-edge communication, reinforcing Manhattan structure consistency through spatial correlation. Finally, to maintain manufacturability, the remaining longer segments are adaptively partitioned using evenly distributed anchor points that satisfy minimum printable feature length requirements. Each resulting sub-segment conforms to design-for-manufacturability (DFM) rules, ensuring that the final augmented representation remains both accurate and fabrication-friendly.

4.2 Manhattan Layout Tokenization

Diffusion Transformer for Manhattan Representation. Given a decomposed layout consisting of Manhattan-aligned polygons, we first extract a set of uniformly sampled edge points $\{(x_i, y_i)\}_{i=1}^N$, where each point lies on either a horizontal or vertical polygon edge. These coordinates form a fixed, structured matrix $\mathbf{X} \in \mathbb{R}^{N\times 2}$, which serves as the conditioning input for the model.

The objective of OPC in Manhattan representation is to predict a set of per-point corrections $\Delta = \{\delta_i\}_{i=1}^N$, where each $\delta_i \in \mathbb{R}$ denotes the scalar offset applied along a single axis (horizontal correction for vertical edges and vice versa). These corrections are learned via a conditional diffusion framework, where the correction field Δ is the target variable being progressively noised and denoised.

To simulate the correction refinement process, we apply a forward diffusion process over the correction vector Δ, generating a noisy version Δ_t at timestep t:

$$q(\Delta_t \mid \Delta_0) = \mathcal{N}(\Delta_t; \sqrt{\bar{\alpha}_t}\Delta_0, (1 - \bar{\alpha}_t)\mathbf{I}), \tag{4}$$

where Δ_0 is the clean ground-truth correction from self-supervised learning, $\{\beta_t\}_{t=1}^{T}$ is a fixed noise schedule, and $\bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s)$. This process introduces Gaussian noise to simulate uncertainty and process-induced variation in mask design.

The Diffusion Transformer (DiT) is trained to reverse this process by predicting the added noise ϵ from the noisy correction Δ_t, conditioned on the fixed layout geometry $\mathbf{X}$ and the timestep t:

$$\mathcal{L}_{\text{denoise}} = \mathbb{E}_{\Delta_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(\Delta_t, t, \mathbf{X})\|^2\right], \tag{5}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, and $\Delta_t = \sqrt{\bar{\alpha}_t}\Delta_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$.

To enhance the model's spatial understanding, we incorporate two forms of positional encoding alongside $\mathbf{X}$: (1) *Point-level encoding*, using sinusoidal embeddings of (x_i, y_i); and (2) *Polygon-level encoding*, assigning a shared embedding to all points belonging to the same polygon, promoting shape-level consistency.

By constraining each Δ_i to a single axis (depending on the orientation of its associated edge), the model learns to predict minimal, axis-aligned displacements that are compatible with standard manufacturing rules. The final output has shape $(N, 1)$, with each scalar correction applied to the corresponding layout point.

Through this formulation, our method integrates two complementary modeling paradigms. The diffusion component explicitly simulates the inverse lithography process, treating optical proximity effects as a structured forward diffusion that corrupts an ideal mask, and framing mask generation as a reverse denoising process that refines corrections to recover the intended pattern. In parallel, the Transformer serves as a powerful sequence model that captures lithography-aware correction patterns by attending to both local geometry and global layout context. By tokenizing the Manhattan layout into a discrete sequence of axis-aligned primitives, it operates directly within the geometric and manufacturing constraints of the design space, enabling it to reason over structural regularities and long-range interactions in compact token domain. Together, the diffusion mechanism ensures a physically grounded formulation of OPC, while the Transformer enables the learning of complex, context-dependent correction behaviors. The architecture captures both local geometry and long-range context, and the reverse diffusion process refines corrections in a geometry-consistent and manufacturable way.

Self-supervised Learning by `nvdiffrast` Renderer. In alignment with the OPC objective—where the corrected mask should produce a wafer image that closely matches the target layout—we supervise the DiT using a differentiable lithographic simulation pipeline.

After the DiT predicts scalar offsets for the input point set, the adjusted points define new Manhattan-aligned segments, which are rasterized into a binary mask image $M \in \{0,1\}^{H\times W}$ using `nvdiffrast` [25], a GPU-accelerated differentiable renderer. This mask is then passed through a differentiable lithography simulator, which models projection optics and resist behavior to generate an aerial image $A \in \mathbb{R}^{H\times W}$.

The reference layout is similarly rasterized into a binary target image $T \in \{0,1\}^{H\times W}$. The aerial image is compared against the target layout using a pixel-wise Mean Squared Error (MSE) loss:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}(A_{ij} - T_{ij})^2. \tag{6}$$

This loss signal is backpropagated through both the lithography simulation and the mask rasterization steps, enabling end-to-end training. The model thus learns to adjust the layout geometry not by direct supervision from ground truth OPC masks, but by minimizing the deviation between simulated and desired lithographic outcomes. The entire pipeline—geometry encoding, mask generation, rendering, and aerial image prediction—is differentiable, supporting fully self-supervised learning.

Bench	Metrics	GAN-OPC [9]	Neural-ILT [10]	DevelSet [8]	Multi-ILT [11]	IL-ILT [12]	Ours
ICCAD13-S	EPE	8.86	6.51	5.20	3.18	2.70	**2.16**
	$\#shots$	861.5	1463.1	955.4	6734.8	4948.4	**287.6**
	TAT	15.69	14.37	1.53	1.23	2.87	**0.89**
ICCAD13-L	EPE	4.73	4.29	3.64	2.22	1.80	**1.38**
	$\#shots$	1823.3	2616.4	2195.8	6786.9	7916.4	**642.3**
	TAT	16.89	11.20	1.54	**1.33**	2.92	1.57

Table 1: Comparison of SOTA methods on ICCAD13 benchmarks

4.3 Post-Processing for Manufacturability

Although TokMan produces geometrically valid and lithographically informed mask predictions, model training is statistical in nature and optimized across large-scale datasets. In practice, layout-specific variations may require additional fine-tuning to ensure manufacturability under strict process windows.

To address this, we apply a lightweight ILT refinement step after inference. This post-processing module performs localized corrections to the DiT output, further minimizing aerial image error while preserving Manhattan structure. Since the model's prediction already adheres closely to the correct pattern, ILT operates in a narrow solution space, improving convergence speed and eliminating the risk of generating non-Manhattan features. This hybrid approach—combining learning-based generalization with deterministic refinement—ensures that each generated mask meets industrial-grade requirements for fidelity, printability, and integration into EDA workflows.

5 Experiments

5.1 Experimental Settings

Training Settings. TokMan model uses 6 Transformer blocks with embedding dimension 512 and 16 heads of self-attention, followed by a linear projection layer for opc correction output. We implement our work wth PyTorch-Geometric toolkit and train our model on the platform that possesses 32x NVIDIA H20 Graphics Cards, requiring approximately 80 GB of memory for batch size of 40 and test on 1x A100 Graphics Card. For lithography simulator settings, the photoresist intensity threshold for lithography settings is set at 0.225, and sigmoid steepness is 50. The lithography wavelength is 193 nm with a defocus range of ± 30 nm and a dose range of $\pm 3\%$. The resolution of the mask and corresponding wafer image are all $1nm$/pixel.

Dataset. To train our model effectively on Manhattan-aligned layout patterns, we construct a large-scale dataset derived from the publicly available benchmark [24]. Specifically, we extract polygonal shapes from the original layouts and systematically regenerate new samples by rearranging these polygons based on realistic placement characteristics observed in actual design environments. This data generation strategy ensures that the resulting samples remain consistent with typical layout design constraints while significantly enriching the diversity of polygon combinations. In total, we generate 89,697 layout samples as our training data. We evaluate our method on the ICCAD2013 benchmark [45], which includes simple and complex cases: ICCAD13-S and ICCAD13-L.

Evaluation Metrics. To comprehensively assess the performance of our OPC approach, we adopt three key metrics: **Edge Placement Error (EPE)**, **mask shots**, and **Turn-Around Time (TAT)**.

Edge Placement Error (EPE) [46] is used to evaluate the lithographic fidelity of the optimized mask. It measures the difference between the wafer image simulated from the corrected mask and the intended target layout. While prior works typically use an ℓ_2 distance metric that compares the wafer and target images pixel-wise, this approach is overly influenced by background regions and fails to emphasize pattern fidelity around critical edges. In contrast, we adopt the industry-standard EPE metric, as shown in Figure 5, where merit points are placed along the edges of the target layout to measure their distance to the corresponding contour in the wafer image. Both inner errors and outer errors are taken into consideration.

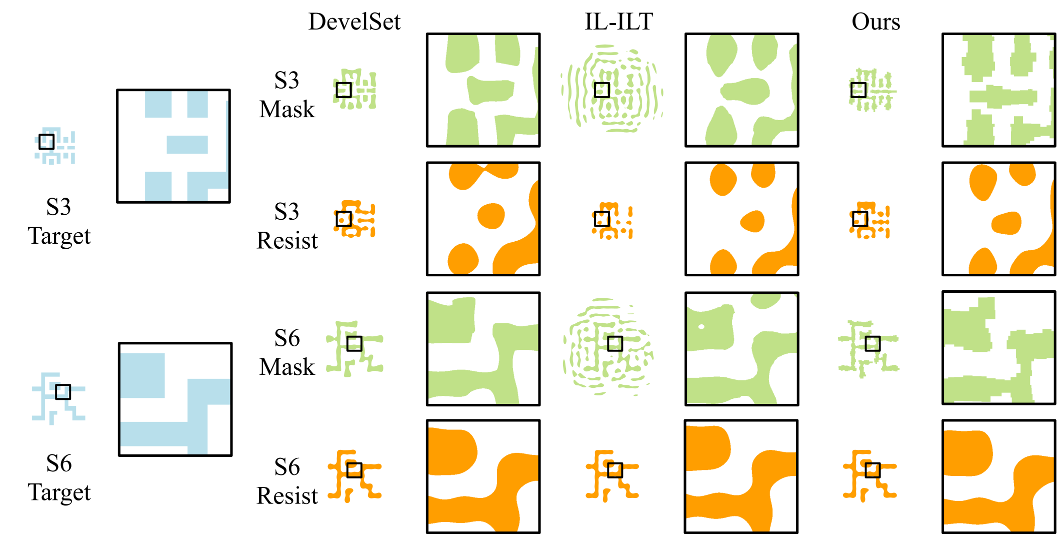

Figure 4: Results Visualization.

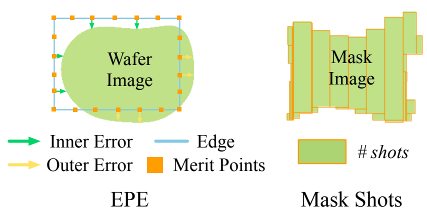

Figure 5: Metrics used for Comparison.

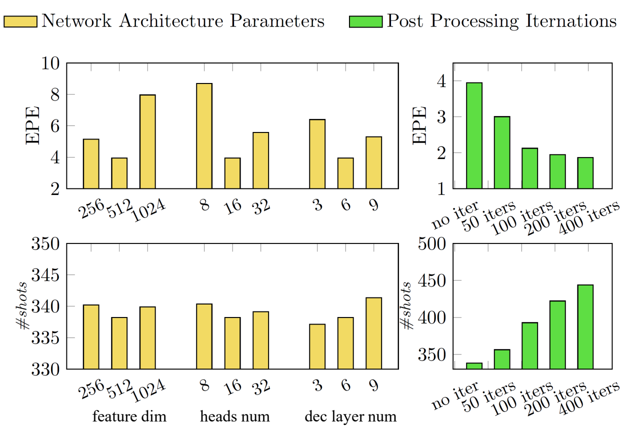

Figure 6: Ablation Study.

Formally, given N sampled points $\{p_i\}_{i=1}^N$ on the target layout boundary, and their respective distances d_i to the wafer image contour, we define the mean EPE as:

$$\text{EPE} = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{7}$$

Unlike Zheng et al. [24], who report only the number of violations exceeding a given threshold, our mean-based formulation provides a continuous and more sensitive indicator that better reflects lithographic performance improvements.

Mask shots [47] measure the manufacturability of the generated mask. In practice, the mask layout is decomposed into a series of rectangular fragments, or "shots", which are individually written by an Electron Beam Lithography (EBL) system, as shown in Figure 5. Fewer mask shots imply a shorter writing time and higher production efficiency. Therefore, we count the number of shots required to fabricate each layout and use this value to evaluate the mask efficiency of both our method and competing baselines.

Turn-Around Time (TAT) quantifies the computational efficiency of each method. For optimization-based approaches, we record the total runtime under a fixed number of iterations. For model-based baselines, we measure the forward inference time. For our method, we report the sum of inference time and post-processing duration (e.g., ILT refinement). All TAT measurements exclude file I/O, such as reading input or saving results, to ensure fair benchmarking.

5.2 Comparison with SOTA Methods

We compare our method against several recent state-of-the-art approaches, including GAN-OPC, Neural-ILT, Devel-Set, Multi-ILT, and IL-ILT. All methods are evaluated using the same metrics introduced earlier.

In terms of EPE, as shown in Table 1, which directly reflects the accuracy of the resist pattern after photolithographic simulation, our method achieves over 20% improvement over the SOTA, IL-ILT. This improvement is not only numerical but also visually significant. As shown in our visualization Figure 4, our method yields printed wires with fine alignment and topological consistency to the intended circuit, especially in fine and densely packed regions.

We also observe substantial gains in mask shot count, illustrated in Table 1, a key indicator of how easily a mask can be manufactured. Thanks to the Manhattanized mask design produced by our framework, we achieve an approximately 4× reduction in mask shot count compared to other methods, indicating that our approach is far more production-friendly.

In terms of TAT, our method maintains a speed comparable to the fastest existing techniques. Despite introducing ILT refinement, our optimizations ensure the overall pipeline remains efficient and practical for deployment.

5.3 Ablation Study

We conduct ablation experiments on the augmented dataset with 100,000 training iterations. All other configurations follow Section 5. We evaluate both lithographic fidelity (EPE) and manufacturability (#shots), as summarized in Figure 6.

Architecture Parameters. We investigate the effect of key architectural hyperparameters, including feature dimension, number of attention heads, and number of Transformer layers. As shown on the left side of Figure 6, increasing the feature dimension from 256 to 512 yields a clear improvement in EPE, while further enlargement provides diminishing returns. Adjusting the number of heads or layers exhibits no consistent benefit and occasionally degrades performance, likely due to overfitting to local geometric patterns. The #shots metric remains relatively stable across configurations, indicating that network capacity primarily influences lithographic accuracy rather than manufacturability.

ILT Refinement. We further examine the effect of post-processing iterations on mask quality. As shown on the right of Figure 6, additional ILT iterations substantially reduce EPE but lead to a steady increase in #shots and computational cost. Performance gains saturate beyond 100 iterations, where further refinement slightly improves fidelity but at the expense of higher mask complexity. This suggests that moderate ILT refinement (around 100 iterations) offers the best trade-off between accuracy and efficiency.

6 Discussion

Limitations. While our approach offers notable improvements in lithographic fidelity and mask manufacturability, several limitations remain. First, although operating in the Manhattan space significantly reduces optimization redundancy and simplifies pattern representation, it currently lacks scalability in certain aspects—most notably, the inability to freely incorporate Manhattan sub-resolution assist features (SRAFs) within the framework. Second, all evaluations in this work are conducted on sliced layout datasets, which isolate regions for targeted optimization. As a result, the method has not yet been tested on full-chip, unsliced layouts, where scalability and context-aware performance are critical. Finally, while our results show strong promise on synthetic and research-oriented data, further validation on real industrial production datasets is essential to assess generalizability and practical deployment viability.

Conclusions. This paper presents TokMan, a tokenized Manhattan representation framework for optical proximity correction under strict geometric constraints. By aligning the OPC task with a sequence modeling paradigm, TokMan leverages a conditional Diffusion Transformer to operate directly on axis-aligned layout tokens, capturing both local and global spatial dependencies while inherently respecting Manhattan compliance. The model interprets lithographic distortion as a structured noise process and learns to reverse it through self-supervised denoising, eliminating the need for ground-truth masks. Experimental results demonstrate that TokMan not only achieves high pattern fidelity and manufacturability but also offers improved runtime efficiency compared to traditional and learning-based ILT methods. This work highlights the effectiveness of combining discrete geometric priors with modern generative modeling, advancing scalable and production-ready solutions for layout correction in semiconductor manufacturing.

Future Work. While TokMan is designed for Manhattan-constrained mask correction, the broader idea of tokenizing structured geometry may inspire a new paradigm for inverse problems across computational imaging. In many domains—ranging from remote sensing and medical imaging to crystallography and 3D reconstruction—observed signals are projections or distortions of latent, structure-regular domains. By introducing a discrete, symbolic representation aligned with domain-specific priors, token-based modeling can disentangle complex imaging processes and enable robust, interpretable, and scalable reconstruction pipelines. This perspective opens up promising directions for structured inverse rendering, symbolic image reconstruction, and hybrid physical-learning systems.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under Grants W2431046 and 62350610269. It is also sponsored by Natural Science Foundation of Shanghai (Project No.25JD1403000), and by MoE Key Lab of Intelligent Perception and Human-Machine Collaboration (ShanghaiTech University), the Shanghai Frontiers Science Center of Human-centered Artificial Intelligence, and the HPC Platform of ShanghaiTech University.

References

[1] Xiaohu Lu, Jian Yaoy, Haoang Li, Yahui Liu, and Xiaofeng Zhang. 2-line exhaustive searching for real-time vanishing point estimation in manhattan world. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 345–353. IEEE, 2017.

[2] Mahdi Yazdanpour, Guoliang Fan, and Weihua Sheng. Online reconstruction of indoor scenes with local manhattan frame growing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[3] Eunju Jeong, Jina Lee, Suyoung Kang, and Pyojin Kim. Linear four-point lidar slam for manhattan world environments. *IEEE Robotics and Automation Letters*, 8(11):7392–7399, 2023.

[4] Miriam Schönbein and Andreas Geiger. Omnidirectional 3d reconstruction in augmented manhattan worlds. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 716–723. IEEE, 2014.

[5] Cadence Design Systems. PCB Manhattan routing techniques, 2020.

[6] Photomask, 2025. Revision dated 12 May 2025.

[7] S. et al. Lin. Projection optical lithography. *Materials Today*, 8(6):28–35, 2005.

[8] Guojin Chen, Ziyang Yu, Hongduo Liu, Yuzhe Ma, and Bei Yu. DevelSet: Deep neural level set for instant mask optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 42(12):5020–5033, 2023.

[9] Haoyu Yang, Shuhe Li, Zihao Deng, Yuzhe Ma, Bei Yu, and Evangeline F. Y. Young. GAN-OPC: mask optimization with lithography-guided generative adversarial nets. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2020.

[10] Bentian Jiang, Lixin Liu, Yuzhe Ma, Bei Yu, and Evangeline F. Y. Young. Neural-ILT 2.0: Migrating ILT to domain-specific and multitask-enabled neural network. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2022.

[11] Shuyuan Sun, Fan Yang, Bei Yu, Li Shang, and Xuan Zeng. Efficient ILT via multi-level lithography simulation. In *ACM/IEEE Design Automation Conference (DAC)*, 2023.

[12] Haoyu Yang and Haoxing Ren. ILILT: implicit learning of inverse lithography technologies. In *International Conference on Machine Learning (ICML)*, 2024.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[14] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020.

[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[16] Marko Sarstedt, Susanne J Adler, Lea Rau, and Bernd Schmitt. Using large language models to generate silicon samples in consumer and marketing research: Challenges, opportunities, and guidelines. *Psychology & Marketing*, 41(6):1254–1270, 2024.

[17] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision (ECCV)*, pages 580–597. Springer, 2022.

[18] Norman Mu, Jingwei Ji, Zhenpei Yang, Nate Harada, Haotian Tang, Kan Chen, Charles R Qi, Runzhou Ge, Kratarth Goel, Zoey Yang, et al. MoST: Multi-modality scene tokenization for motion prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14988–14999, 2024.

[19] Limei Wang, Kaveh Hassani, Si Zhang, Dongqi Fu, Baichuan Yuan, Weilin Cong, Zhigang Hua, Hao Wu, Ning Yao, and Bo Long. Learning graph quantized tokenizers. In *International Conference on Learning Representations (ICLR)*, 2024.

[20] Nihal Siddeek et al. MeshGPT: Generating triangle meshes with decoder-only transformers. *arXiv preprint arXiv:2302.04181*, 2023.

[21] Jiaman Li, Karen Liu, and Jiajun Wu. Ego-body pose estimation via ego-head pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17142–17151, 2023.

[22] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4195–4205, 2023.

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

[24] Su Zheng, Haoyu Yang, Binwu Zhu, Bei Yu, and Martin D.F. Wong. LithoBench: Benchmarking AI computational lithography for semiconductor manufacturing. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[25] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020.

[26] Xiaoguo Zhang, Guo Wang, Ye Gao, Huiqing Wang, and Qing Wang. An improved building reconstruction algorithm based on manhattan world assumption and line-restricted hypothetical plane fitting. *Mathematical Problems in Engineering*, 2020(1):9267854, 2020.

[27] James M Coughlan and Alan L Yuille. The Manhattan world assumption: Regularities in scene statistics which enable Bayesian inference. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 13, 2000.

[28] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the Manhattan-world assumption. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5521–5530, 2022.

[29] Nobuhiko Wakai, Satoshi Sato, Yasunori Ishii, and Takayoshi Yamashita. Deep single image camera calibration by heatmap regression to recover fisheye images under manhattan world assumption. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12345–12354, 2023.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 4171–4186, 2019.

[31] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 35:27730–27744, 2022.

[32] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1439–1449, 2021.

[33] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*, pages 4651–4664. PMLR, 2021.

[34] Zhimin Chen, Liang Yang, Yingwei Li, Longlong Jing, and Bing Li. SAM-guided masked token prediction for 3d scene understanding. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[35] Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024.

[36] 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 *ACM International Symposium on Physical Design (ISPD)*, 2024.

[37] Chris A Mack. PROLITH: a comprehensive optical lithography model. In *Optical Microlithography IV*, volume 538, pages 207–220. SPIE, 1985.

[38] Chris A Mack. Thirty years of lithography simulation. In *Optical Microlithography XVIII*, volume 5754, pages 1–12. SPIE, 2005.

[39] Chris Mack. *Fundamental principles of optical lithography: the science of microfabrication*. John Wiley & Sons, 2008.

[40] Linyong Pang. Inverse lithography technology: 30 years from concept to practical, full-chip reality. *Journal of Micro/Nanopatterning, Materials, and Metrology*, 20(3):030901–030901, 2021.

[41] Daniel S Abrams and Linyong Pang. Fast inverse lithography technology. In *Optical Microlithography XIX*, volume 6154, pages 534–542. SPIE, 2006.

[42] Yuri Granik. Fast pixel-based mask optimization for inverse lithography. *Journal of Micro/Nanolithography, MEMS and MOEMS*, 5(4):043002–043002, 2006.

[43] Guojin Chen, Haoyu Yang, Haoxing Mark Ren, Bei Yu, and David Z Pan. Differentiable Edge-based OPC. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–9, 2024.

[44] Guojin Chen, Hao Geng, Bei Yu, and David Z Pan. Open-source differentiable lithography imaging framework. In *DTCO and Computational Patterning III*, volume 12954, pages 118–127. SPIE, 2024.

[45] Shayak Banerjee, Zhuo Li, and Sani R. Nassif. Iccad-2013 cad contest in mask optimization and benchmark suite. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 271–274, 2013.

[46] Yoshishige Sato, Shang-Chieh Huang, Kotaro Maruyama, and Yuichiro Yamazaki. Edge placement error measurement in lithography process with die to database algorithm. In *Metrology, Inspection, and Process Control for Microlithography XXXIII*, volume 10959, pages 61–70. SPIE, 2019.

[47] 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, pages 830–840. SPIE, 2011.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our contributions are mentioned in last two paragraph in introduction and discussed amply in method, which are aligned.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The pipeline and network architecture is clearly displayed in figure 2 and figure 3. More detailed network parameters are also mentioned in experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We used publicly available datasets, which is mentioned in experiments. For privacy, we will consider to release our code in future.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This part is mentioned in section 5 in detail.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiments do not include error bars or statistical significance tests, as our evaluation focuses on deterministic metrics (e.g., edge placement error and mask shot count) computed on standardized benchmarks. Each method is evaluated on the full test set with averaged results reported.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: These details are mentioned in section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our articles do not involve ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We included the future work discussing our approach applied in boarder domains.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited all the assets we used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We proposed our diffusion transformer model in this paper. We designed and trained our network from scratch and the process is carefully explained in method.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Additional Results of TokMan

To further demonstrate the robustness and generalizability of our TokMan framework, we provide detailed comparisons across 20 benchmark layouts from the ICCAD2013-S/L datasets. As shown in Table 2, our method consistently outperforms previous state-of-the-art OPC approaches—including GAN-OPC[9], Neural-ILT[10], Devel-Set[8], Multi-ILT[11], and IL-ILT[12]—across all three key evaluation metrics: Edge Placement Error (EPE), mask shot count, and Turn-Around Time (TAT).

Compared to the SOTA, IL-ILT, our method achieves an average of over 20% improvement in EPE, and a 4-7× reduction in mask shots, underscoring its manufacturability advantage. Importantly, this performance is achieved with minimal runtime overhead, and even with ILT refinement included, our TAT remains competitive with the fastest baselines.

In addition to numerical metrics, Figure 8 and 9 presents qualitative comparisons on several representative layouts, which is more critical than others. The masks generated by TokMan are not only visually cleaner and Manhattan-aligned, but also produce resist patterns that closely match the desired targets, validating the effectiveness of our discrete token correction process. Notably, TokMan avoids the wavy, curvilinear artifacts observed in other methods, preserving edge integrity and topological consistency, while effectively preventing feature bridging and line-end breakage, which helps ensure the functional reliability of the underlying circuit.

These comprehensive results reinforce TokMan's strength in balancing fidelity, manufacturability, and efficiency, offering a production-ready solution for OPC under strict design rules.

Bench	GAN-OPC[9]			Neural-ILT[10]			DevelSet[8]			Multi-ILT[11]			IL-ILT[12]			Ours		
	EPE	$\#shots$	TAT	EPE	$\#shots$	TAT	EPE	$\#shots$	TAT	EPE	$\#shots$	TAT	EPE	$\#shots$	TAT	EPE	$\#shots$	TAT
S1	11.56	960	10.54	7.70	1561	10.41	6.09	1159	1.53	4.48	7505	0.84	3.88	5416	2.95	3.67	332	1.16
S2	10.30	754	10.46	5.30	1337	10.34	5.27	767	1.52	4.00	6772	0.84	3.92	5192	2.87	3.03	290	1.27
S3	32.92	1476	10.49	29.56	1964	10.34	15.07	1334	1.55	9.75	6906	1.30	7.97	5446	2.90	7.56	374	1.05
S4	8.22	498	11.08	4.22	904	10.35	5.88	303	1.54	2.55	7076	1.34	2.03	4220	2.83	0.33	132	0.69
S5	4.32	1190	20.22	4.64	1948	16.36	4.09	1187	1.51	2.16	6742	1.31	1.81	5710	2.87	1.40	312	0.75
S6	5.01	1134	15.30	3.24	1786	16.81	3.31	1313	1.52	2.43	7170	1.31	1.96	5687	2.87	1.60	392	0.71
S7	2.99	613	15.09	2.07	1555	14.96	2.75	713	1.54	1.22	5856	1.32	0.84	5145	2.85	0.67	278	0.82
S8	3.59	349	19.46	3.17	1009	14.96	3.41	635	1.55	1.97	6999	1.31	1.64	3970	2.86	1.30	169	0.69
S9	7.00	1229	22.24	3.90	1813	26.13	4.19	1565	1.52	2.54	6485	1.38	2.12	5618	2.87	1.53	454	0.97
S10	2.68	412	21.98	1.25	754	13.01	1.90	578	1.53	0.65	5837	1.38	0.81	3080	2.83	0.51	143	0.75
L1	5.03	1971	22.09	4.04	2755	10.35	3.88	2404	1.52	2.40	6999	1.35	2.21	8061	2.93	1.99	725	1.85
L2	5.31	1679	22.98	5.02	2498	10.35	3.54	2025	1.51	2.52	6774	1.30	2.02	8252	2.93	1.62	620	1.96
L3	10.59	2115	22.51	13.35	3342	10.36	7.59	2606	1.52	5.16	7394	1.33	3.86	8655	2.96	3.08	750	1.69
L4	4.17	1216	19.80	3.55	2245	10.36	3.46	1570	1.54	1.54	6851	1.32	1.40	7740	2.90	0.76	514	1.56
L5	4.05	2401	22.13	3.56	2752	10.36	3.45	2503	1.53	1.85	6503	1.30	1.63	7992	2.93	1.14	724	1.22
L6	3.58	2197	17.40	2.76	2690	10.38	3.27	2519	1.67	2.59	7047	1.41	1.86	8257	2.93	1.33	732	1.69
L7	3.27	1557	10.50	2.35	2319	10.37	2.68	1769	1.51	1.46	6387	1.35	1.00	7310	2.89	0.90	515	1.25
L8	3.20	1335	10.49	2.53	2575	10.40	2.76	1913	1.54	1.42	6536	1.32	1.22	7939	2.91	0.97	541	1.09
L9	5.30	2616	10.48	3.44	2845	10.42	3.45	2739	1.52	2.27	6636	1.33	1.87	7433	2.94	1.23	778	1.39
L10	2.77	1146	10.48	2.30	2143	18.61	2.32	1910	1.53	1.00	6742	1.28	0.98	7525	2.90	0.82	524	2.00

Table 2: Comparison of six OPC methods across 20 benchmark layouts with evaluation metrics: EPE, mask shots, and TAT.

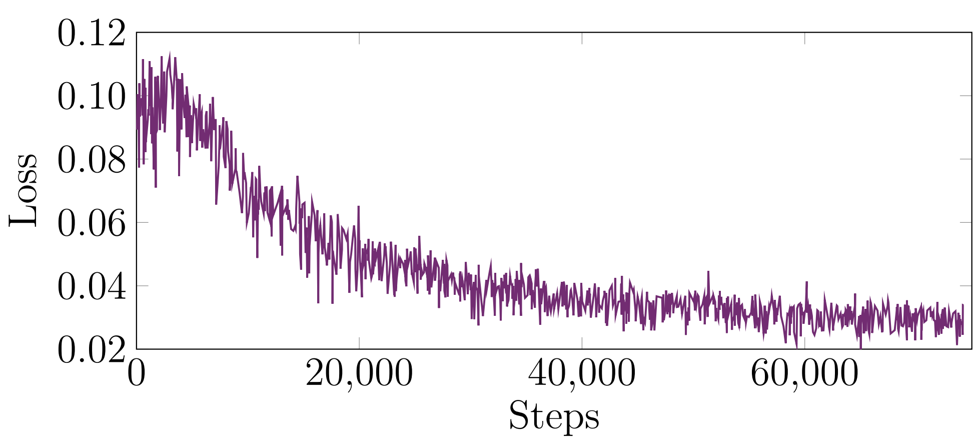

Figure 7: Training loss.

B Results Visualization

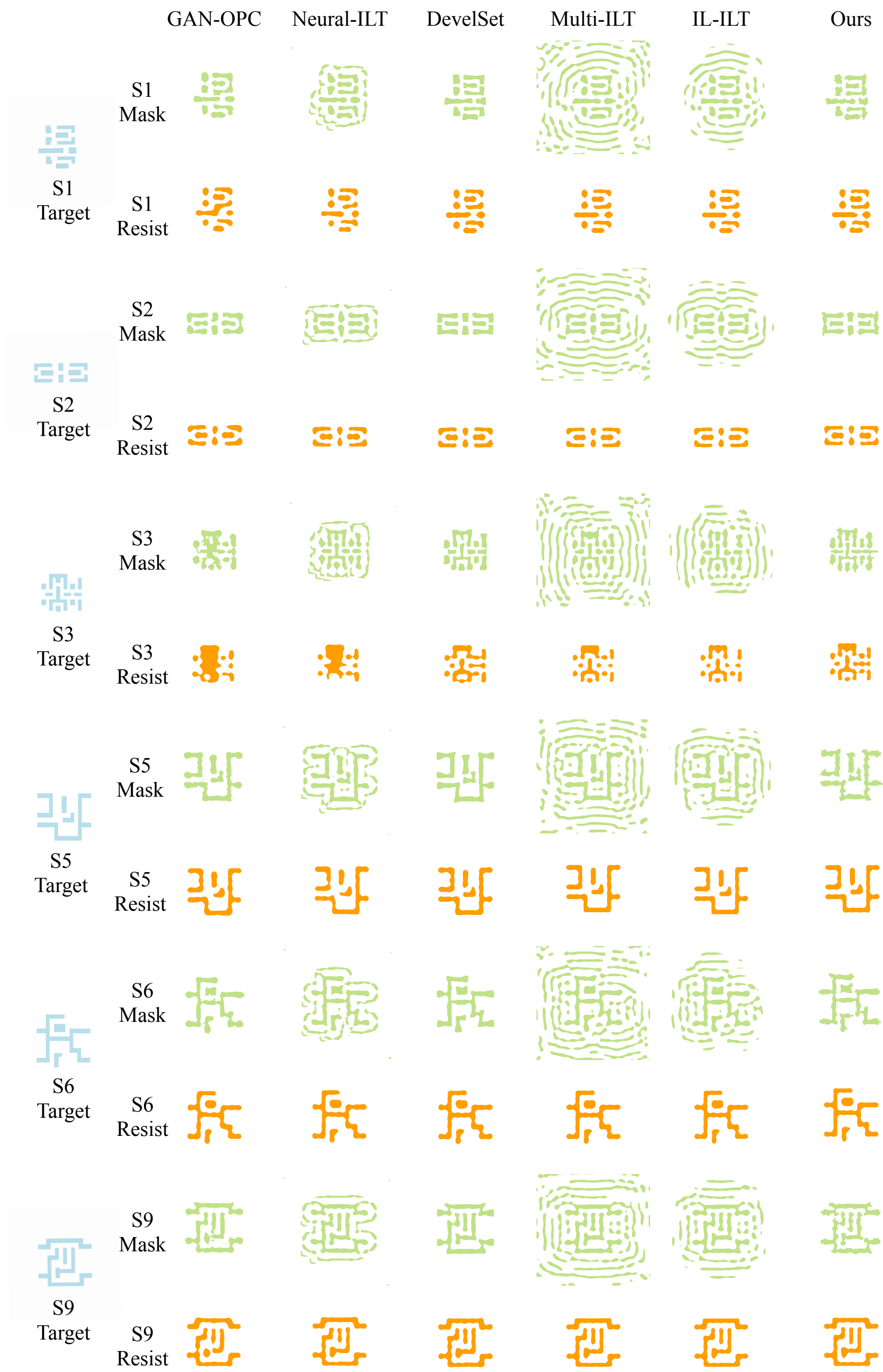

Figure 8: Visualization Comparison with SOTA on selected critical layouts of ICCAD13-S.

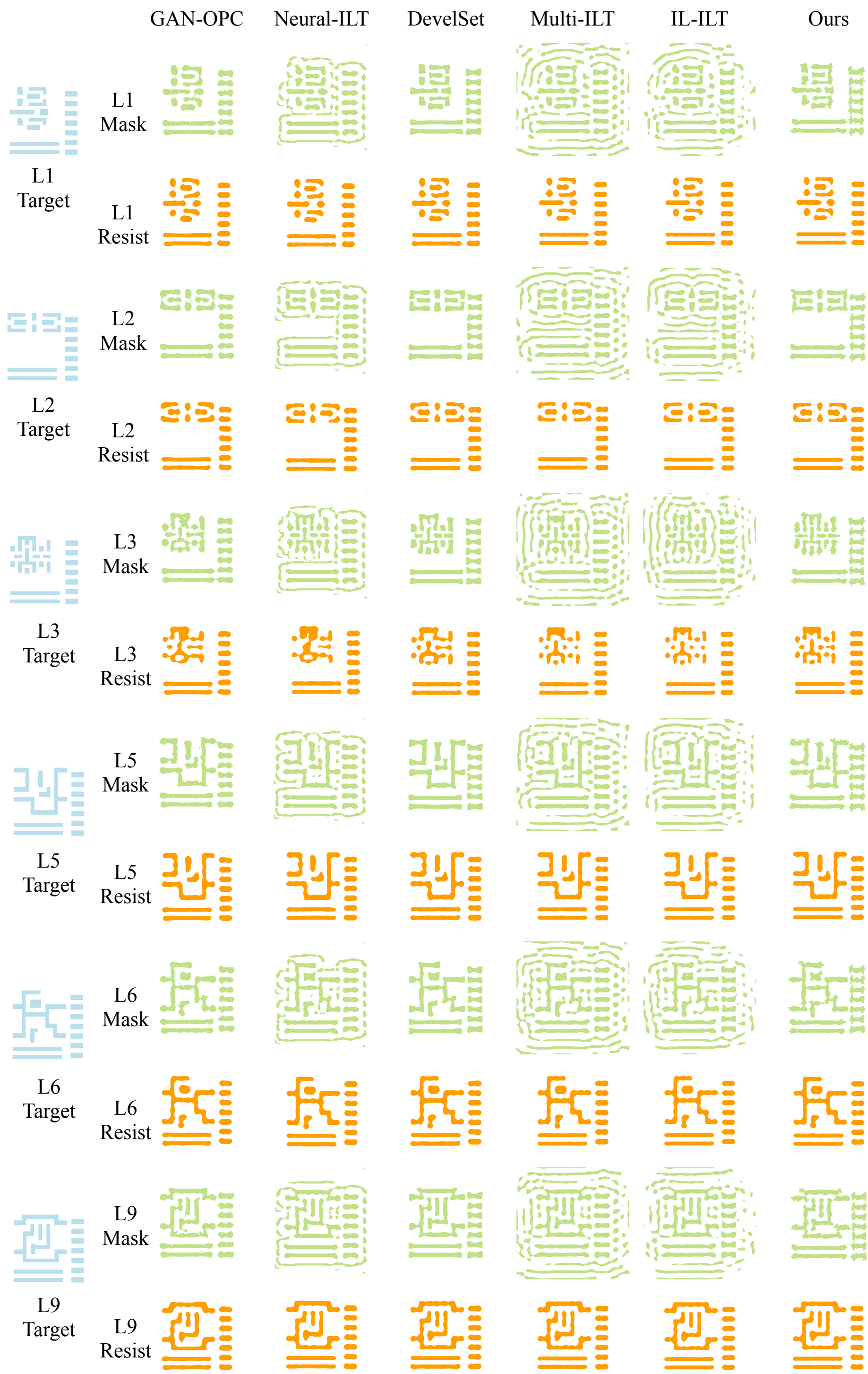

Figure 9: Visualization Comparison with SOTA on selected critical layouts of ICCAD13-L.

References

[1] Xiaohu Lu, Jian Yaoy, Haoang Li, Yahui Liu, and Xiaofeng Zhang. 2-line exhaustive searching for real-time vanishing point estimation in manhattan world. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 345–353. IEEE, 2017.

[2] Mahdi Yazdanpour, Guoliang Fan, and Weihua Sheng. Online reconstruction of indoor scenes with local manhattan frame growing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[3] Eunju Jeong, Jina Lee, Suyoung Kang, and Pyojin Kim. Linear four-point lidar slam for manhattan world environments. *IEEE Robotics and Automation Letters*, 8(11):7392–7399, 2023.

[4] Miriam Schönbein and Andreas Geiger. Omnidirectional 3d reconstruction in augmented manhattan worlds. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 716–723. IEEE, 2014.

[5] Cadence Design Systems. PCB Manhattan routing techniques, 2020.

[6] Photomask, 2025. Revision dated 12 May 2025.

[7] S. et al. Lin. Projection optical lithography. *Materials Today*, 8(6):28–35, 2005.

[8] Guojin Chen, Ziyang Yu, Hongduo Liu, Yuzhe Ma, and Bei Yu. DevelSet: Deep neural level set for instant mask optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 42(12):5020–5033, 2023.

[9] Haoyu Yang, Shuhe Li, Zihao Deng, Yuzhe Ma, Bei Yu, and Evangeline F. Y. Young. GAN-OPC: mask optimization with lithography-guided generative adversarial nets. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2020.

[10] Bentian Jiang, Lixin Liu, Yuzhe Ma, Bei Yu, and Evangeline F. Y. Young. Neural-ILT 2.0: Migrating ILT to domain-specific and multitask-enabled neural network. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2022.

[11] Shuyuan Sun, Fan Yang, Bei Yu, Li Shang, and Xuan Zeng. Efficient ILT via multi-level lithography simulation. In *ACM/IEEE Design Automation Conference (DAC)*, 2023.

[12] Haoyu Yang and Haoxing Ren. ILILT: implicit learning of inverse lithography technologies. In *International Conference on Machine Learning (ICML)*, 2024.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[14] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020.

[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[16] Marko Sarstedt, Susanne J Adler, Lea Rau, and Bernd Schmitt. Using large language models to generate silicon samples in consumer and marketing research: Challenges, opportunities, and guidelines. *Psychology & Marketing*, 41(6):1254–1270, 2024.

[17] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision (ECCV)*, pages 580–597. Springer, 2022.

[18] Norman Mu, Jingwei Ji, Zhenpei Yang, Nate Harada, Haotian Tang, Kan Chen, Charles R Qi, Runzhou Ge, Kratarth Goel, Zoey Yang, et al. MoST: Multi-modality scene tokenization for motion prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14988–14999, 2024.

[19] Limei Wang, Kaveh Hassani, Si Zhang, Dongqi Fu, Baichuan Yuan, Weilin Cong, Zhigang Hua, Hao Wu, Ning Yao, and Bo Long. Learning graph quantized tokenizers. In *International Conference on Learning Representations (ICLR)*, 2024.

[20] Nihal Siddeek et al. MeshGPT: Generating triangle meshes with decoder-only transformers. *arXiv preprint arXiv:2302.04181*, 2023.

[21] Jiaman Li, Karen Liu, and Jiajun Wu. Ego-body pose estimation via ego-head pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17142–17151, 2023.

[22] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4195–4205, 2023.

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

[24] Su Zheng, Haoyu Yang, Binwu Zhu, Bei Yu, and Martin D.F. Wong. LithoBench: Benchmarking AI computational lithography for semiconductor manufacturing. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[25] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020.

[26] Xiaoguo Zhang, Guo Wang, Ye Gao, Huiqing Wang, and Qing Wang. An improved building reconstruction algorithm based on manhattan world assumption and line-restricted hypothetical plane fitting. *Mathematical Problems in Engineering*, 2020(1):9267854, 2020.

[27] James M Coughlan and Alan L Yuille. The Manhattan world assumption: Regularities in scene statistics which enable Bayesian inference. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 13, 2000.

[28] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the Manhattan-world assumption. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5521–5530, 2022.

[29] Nobuhiko Wakai, Satoshi Sato, Yasunori Ishii, and Takayoshi Yamashita. Deep single image camera calibration by heatmap regression to recover fisheye images under manhattan world assumption. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12345–12354, 2023.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 4171–4186, 2019.

[31] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 35:27730–27744, 2022.

[32] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1439–1449, 2021.

[33] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*, pages 4651–4664. PMLR, 2021.

[34] Zhimin Chen, Liang Yang, Yingwei Li, Longlong Jing, and Bing Li. SAM-guided masked token prediction for 3d scene understanding. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[35] Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024.

[36] 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 *ACM International Symposium on Physical Design (ISPD)*, 2024.

[37] Chris A Mack. PROLITH: a comprehensive optical lithography model. In *Optical Microlithography IV*, volume 538, pages 207–220. SPIE, 1985.

[38] Chris A Mack. Thirty years of lithography simulation. In *Optical Microlithography XVIII*, volume 5754, pages 1–12. SPIE, 2005.

[39] Chris Mack. *Fundamental principles of optical lithography: the science of microfabrication*. John Wiley & Sons, 2008.

[40] Linyong Pang. Inverse lithography technology: 30 years from concept to practical, full-chip reality. *Journal of Micro/Nanopatterning, Materials, and Metrology*, 20(3):030901–030901, 2021.

[41] Daniel S Abrams and Linyong Pang. Fast inverse lithography technology. In *Optical Microlithography XIX*, volume 6154, pages 534–542. SPIE, 2006.

[42] Yuri Granik. Fast pixel-based mask optimization for inverse lithography. *Journal of Micro/Nanolithography, MEMS and MOEMS*, 5(4):043002–043002, 2006.

[43] Guojin Chen, Haoyu Yang, Haoxing Mark Ren, Bei Yu, and David Z Pan. Differentiable Edge-based OPC. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–9, 2024.

[44] Guojin Chen, Hao Geng, Bei Yu, and David Z Pan. Open-source differentiable lithography imaging framework. In *DTCO and Computational Patterning III*, volume 12954, pages 118–127. SPIE, 2024.

[45] Shayak Banerjee, Zhuo Li, and Sani R. Nassif. Iccad-2013 cad contest in mask optimization and benchmark suite. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 271–274, 2013.

[46] Yoshishige Sato, Shang-Chieh Huang, Kotaro Maruyama, and Yuichiro Yamazaki. Edge placement error measurement in lithography process with die to database algorithm. In *Metrology, Inspection, and Process Control for Microlithography XXXIII*, volume 10959, pages 61–70. SPIE, 2019.

[47] 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, pages 830–840. SPIE, 2011.