2026 Design, Automation & Test in Europe Conference (DATE 2026)

No Pixel, More Efficient: Multimodal Framework for Sub-nm Mask Process Correction

Kai Ma[1,3*], Tianyi Li[1,3*], Jiaqi Liu[2], Jingyi Yu[1†], Hao Geng[1,3†]
[1]ShanghaiTech University [2]Shanghai Optoelectronics Science and Technology innovation Center [3]ZeroShot Co., Ltd.

***Abstract*—Mask Process Correction (MPC) is a critical step in advanced semiconductor manufacturing to mitigate pattern errors from e-beam writing and etching. However, conventional CPU-based commercial tools create a severe computational bottleneck as layout sizes and polygon complexity increase, resulting in prohibitively long MPC correction times. While academic efforts have explored using GPUs and advanced image-based machine learning algorithms on mask images, the inherent inefficiency of pixel mask representation results in an intractably large parameter space at the full-tile scale, limiting the practicality of these methods for industrial production. In this paper, we introduce a novel multimodal framework that processes point-cloud representations of GDS masks alongside E-beam Lithography (EBL) modeling information. This multimodal approach enables massively parallel processing, preserves pattern fidelity, and ensures adaptability to variable e-beam lithography conditions. Experimental results show our method significantly reduces runtime while achieving edge placement error performance comparable to commercial tools.**

I. Introduction

As integrated circuit (IC) technology nodes continue to shrink, the error budgets for all manufacturing components are drastically reduced to meet tightening wafer specifications [1]. Among these, the photomask is of paramount importance, as it serves as the master template for wafer-level patterning [2], [3]. This critical role demands unprecedented fabrication precision, especially at advanced nodes where even sub-nanometer (sub-nm) deviations on the mask can lead to catastrophic failures in device performance and yield. However, achieving this fidelity is profoundly challenged by physical distortions such as e-beam blurring and proximity effects. To counteract these problems, Mask Process Correction (MPC) has emerged as an indispensable technique to compensate for systematic errors introduced during electron beam lithography (EBL) and subsequent etching processes [4]–[9].

Existing MPC methodologies fall into two main categories, each with inherent limitations. The commercial tool approach, illustrated in Fig. 1(a), relies on data-efficient vector representations. However, its core algorithms are fundamentally designed for CPU architectures and are not amenable to migration to parallel hardware like GPUs. This creates a significant performance bottleneck, as the CPUs struggle with the massive parallelization required for full-mask correction, leading to prohibitive turnaround times [10]. Conversely, academic research, shown in Fig. 1(b), has leveraged GPU acceleration for pixel-based machine learning (ML) models

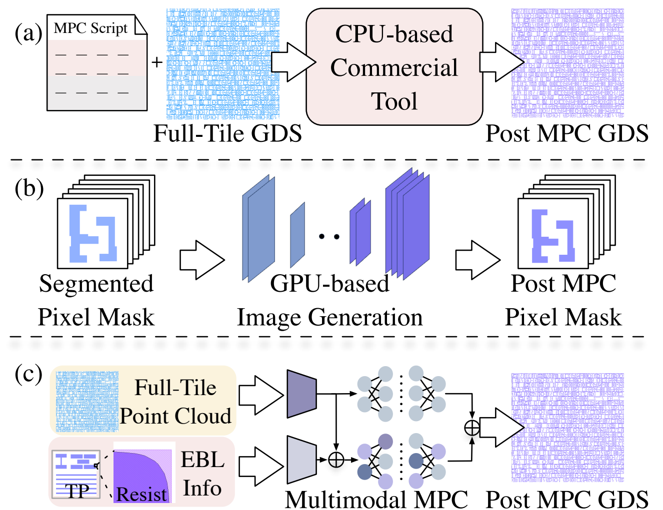


Fig. 1 Mask process correction approaches: (a) Commercial tools use model-based vector correction for high precision but suffer from slow CPU computation. (b) Academic GPU-accelerated image methods face excessive computational costs from pixel representations and limited adaptability. (c) Our Multimodal MPC combines efficient vector geometry with process-aware conditioning to match commercial accuracy at much higher speed.

[11]–[16]. While using faster hardware, the pixel-based format creates an infeasible computational and memory load to achieve sub-nm accuracy. Moreover, these ML models are inflexible, requiring costly retraining for new optical conditions. This fundamental misalignment—**efficient expression on slow hardware versus inefficient expression on fast hardware**—has prevented a breakthrough solution from being adopted in industrial workflows.

To resolve this fundamental misalignment, we introduce **a foundational feature extractor** purpose-built for GPU-accelerated deep learning architectures, capable of retaining the data efficiency of vector representations while comprehending physical lithography information. This extractor, illustrated in Fig. 1(c), is designed to handle two distinct modalities. For mask comprehension, it leverages an efficient point cloud input—a representation that preserves the data efficiency of vector formats. Simultaneously, it accepts lithography information, such as a simulated resist image from a test pattern, to capture the specific physical characteristics of

*Equal contributors.
†Corresponding author.

978-3-9826741-1-7/DATE26/© 2026 EDAA

the process conditions. The key advantage of this multimodal design is its versatility. Once trained, this powerful feature extractor, which fuses geometric awareness with lithography condition awareness, can be readily adapted to a wide range of downstream tasks in the mask process domain.

In its application to MPC, the framework first implements a fast and lossless transformation of GDSII layout data into a sparse point cloud. This method preserves sub-nm vertex information and inherently maintains Manhattan geometries, preventing the introduction of Mask Rule Check (MRC) violations. This point cloud is then processed by a mask encoder to extract geometric features [17]. To embed the lithography conditions, a separate image encoder is utilized, and its output is integrated with the mask information through a conditioning architecture inspired by ControlNet [18]. The fusion of these branches yields a powerful model capable of delivering fast and highly accurate corrections for the specific task of MPC. The main contributions of our work can be summarized as follows:

- **Sub-nm Efficient Point-Cloud Representation**. We introduce a sparse point cloud representation for mask layouts, created via a fast, lossless GDS transformation. This approach bypasses the immense computational overhead of pixel-based methods, enabling efficient, large-scale correction with sub-nanometer precision.
- **Multimodal, Process-Aware Framework**. We propose a multimodal architecture that conditions the correction model on the physical fingerprint of the manufacturing process. By using a resist image of the test pattern, our model adapts to diverse EBL writer conditions, making it a flexible solution applicable across multiple process configurations without retraining.
- **Highly Scalable, Full-Tile Correction Engine**. The proposed framework is designed for end-to-end, large-scale correction. Its efficient handling of geometric data allows it to model global effects across a full tile, overcoming the locality limitations of conventional methods and significantly reducing mask synthesis run time.

Extensive experiments validate the effectiveness of our proposed framework. The results demonstrate that our approach achieves sub-nanometer accuracy comparable to state-of-the-art commercial tools while delivering a scalable speedup of over 200× on large-scale layouts.

II. Preliminaries

A. Mask Process Correction

MPC is a critical step in the post tape-out stage of semiconductor manufacturing, where design intent is transformed into a physical photomask ready for fabrication. During this phase, the idealized mask layout generated by OPC or ILT must be adapted to account for systematic distortions introduced by mask writing and etching processes. Such distortions primarily arise from electron scattering in resist during e-beam exposure, forward and backward proximity effects, as well as etch bias during pattern transfer. Without correction, these effects cause edge placement errors (EPEs) and critical dimension (CD) deviations that propagate to the wafer, ultimately degrading device yield and reliability.

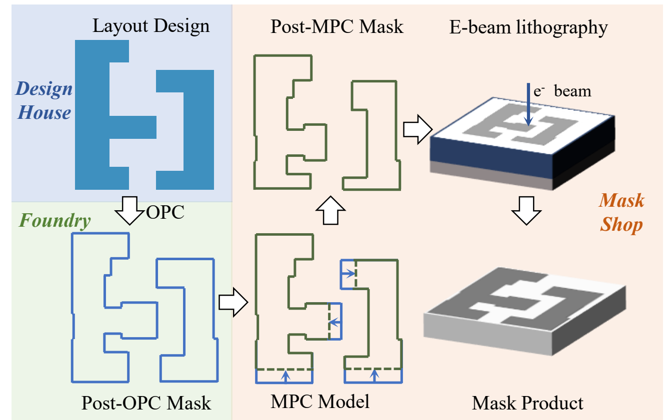


Fig. 2 Post tape-out design-to-mask flow.

As shown in Fig. 2, within the overall tape-out flow, MPC serves as a bridge between OPC mask design and the physical mask writing process. After the OPC stage, MPC refines this OPC mask design by modeling the mask process and applying corrective adjustments so that the printed mask contour faithfully reproduces the intended design. The corrected mask data then undergoes MRC, verification, and finally rasterization for mask writing. In this way, MPC ensures that the lithographic benefits promised by advanced RET (Resolution Enhancement Techniques) are not compromised during mask manufacturing.

B. Point-Cloud Representation for MPC

While the GDSII format is highly efficient for design representation, its vertex-based structure is ill-suited for deep learning-based correction. A model trained to predict vertex coordinates cannot guarantee the preservation of Manhattan geometry, risking non-orthogonal artifacts that compromise manufacturability. To overcome this, we introduce a point-cloud representation using edge midpoints instead of vertices. Our model predicts a normal displacement for each midpoint, which translates to a perpendicular shift of the entire edge segment. This edge-based approach ensures manufacturability by preserving orthogonality and providing jog-free corrections. This maintains the layout's topological integrity—preventing key MRC violations—while serving as a lossless representation for the original design. This combination of geometric fidelity and rule-awareness makes our method uniquely robust and scalable for large-scale, full-tile MPC.

C. Problem Formulation

Definition 1 (Edge Placement Error (EPE)). *Edge Placement Error (EPE) measures the geometric accuracy of the lithography simulation relative to the target design. For each sampled edge point i, the error is computed as the Euclidean distance between the simulated contour point $E_{sim,i}$ and its corresponding target point $E_{target,i}$:*

$$EPE_i = \|E_{sim,i} - E_{target,i}\|_2. \quad (1)$$

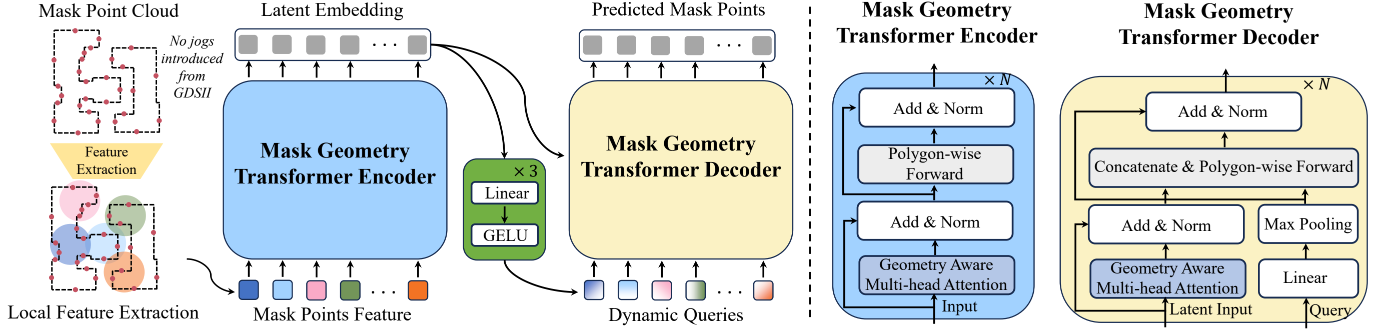


Fig. 3 The backbone architecture of our method.

Definition 2 (Turn around time (TAT)). *Inference time of the MPC model. The goal is to have fast inference.*

Problem 1. *Given a target mask layout in GDSII format and process-dependent parameters from the e-beam writing system, the objective of mask process correction is to generate a corrected mask such that the printed contour x_{print} aligns with the intended target contour x_{target}.*

III. The Proposed Mask Process Correction Flow

As illustrated in Fig. 3, the proposed **MPC backbone** begins by encoding the GDS layout into a mask point cloud in a *jog-free* manner, ensuring that the transformation preserves geometric fidelity without introducing jogs. This point cloud is then processed by the *point cloud correction model*, which formulates mask process correction as a control-point optimization problem. The optimized control points are finally mapped back into GDS format, producing a corrected mask.

A. Point Cloud Correction Model

1) Geometry Extraction via Mask Geometry Transformer

To enable geometry-aware learning for MPC, we propose representing mask layouts as *point clouds*. Unlike conventional MPC preprocessing, this point-based representation is a core component of our framework, designed to provide geometric detail while maintaining computational tractability.

The extracted point cloud is encoded using the **Mask Geometry Transformer (MGT)**, a transformer-based module that captures both local edge variations and global shape correlations. The key mechanism is *multi-head self-attention (MHSA)*, which enables efficient modeling of pairwise spatial relations across control points.

Formally, for a set of point embeddings $H = \{h_i\}_{i=1}^{N}$, attention is defined as

$$\mathrm{Attn}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V, \tag{2}$$

where $Q = HW_Q$, $K = HW_K$, and $V = HW_V$ are learned projections of H. Multi-head attention aggregates multiple such relations in parallel:

$$\mathrm{MHSA}(H) = \mathrm{Concat}\big(\mathrm{Attn}_1(H), \ldots, \mathrm{Attn}_M(H)\big)W_O. \tag{3}$$

By leveraging MHSA, MGT efficiently balances local geometric precision with global contextual awareness, allowing it to capture subtle variations such as corner rounding and edge curvature while preserving overall mask connectivity. The final global embedding provides a compact yet expressive feature representation. These features are subsequently utilized for bias correction within the multimodal MPC controller.

2) Coordinate Bias Prediction via Geometry-Aware Offset Modeling

In mask process correction, systematic deviations in e-beam writing are not purely random noise but exhibit *process-dependent bias*, often arising from electron scattering, energy deposition variations, and proximity effects. Such deviations manifest as correlated shifts in the positions of control points along the mask contour. To explicitly capture these effects, we design a **geometry-aware coordinate bias predictor** that estimates localized offsets for each control point.

The predictor fuses three sources of information: (i) *global geometric context* encoded by the Mask Geometry Transformer (MGT), (ii) *local neighborhood features* extracted through convolutional layers, and (iii) the *original spatial coordinates*. These inputs are processed by a residual deep neural network with bounded outputs, ensuring stable and physically interpretable corrections.

Physical Interpretation. The predicted offsets can be interpreted as a data-driven approximation of the e-beam process transfer function, mapping from ideal design intent to printed geometry. Instead of relying on explicit resist/beam simulations, the predictor directly learns systematic corrections induced by scattering kernels and dose blur, thereby embedding process-awareness into the MPC loop.

Efficiency and Industrial Relevance. Compared with pixel- or grid-based correction methods, the proposed predictor operates directly on *control points* derived from the GDS layout. This design avoids costly image-space convolutions and scales linearly with the number of control points, making it well-suited for large-scale industrial layouts. Moreover, the integration with MGT ensures that both local curvatures and global layout contexts are utilized, enabling sub-nanometer correction accuracy while maintaining compatibility with standard EDA data formats and e-beam writing systems.

B. Condition Network

To incorporate process variability into mask process correction, we replace conventional numerical parameter lists with a

vision-based encoder that learns e-beam writer characteristics directly from *resist images*. This design reflects the fact that many process effects (e.g., beam blur, focus drift, dose variation, etch effects) manifest nonlinearly in printed resist contours and cannot be fully captured by a limited set of scalar parameters.

Concretely, each resist image $I_{\text{resist}} \in \mathbb{R}^{8192\times8192}$ is partitioned into 16×16 non-overlapping patches. Each patch is linearly embedded and passed into a Vision Transformer (ViT) encoder comprising 12 transformer blocks with 8 attention heads. The encoder produces a latent feature sequence F_{resist}, which is aggregated and projected into a compact embedding vector. This representation V_{ebeam} encodes the implicit transfer function of the e-beam system, linking input parameters such as current, voltage, and focus to their lithographic outcomes. By operating in image space, the encoder learns nonlinear correlations among process conditions, resist responses, and geometric distortions, providing a physics-aware condition vector for guiding geometry correction in MPC.

C. Training and Inference Flow

As illustrated in Fig. 4, the overall pipeline consists of three stages: pretraining, finetuning, and inference.

In the **pretraining stage**, the backbone is trained on multi-scale MPCed GDS layouts to capture robust geometric representations. This large-scale pretraining enables the model to learn generic correction patterns without overfitting to specific process conditions.

In the **finetuning stage**, the pretrained backbone is frozen, and additional trainable layers are optimized under varying e-beam lithography conditions. A condition network is employed to encode these variations, allowing the model to adapt its correction strategy to specific process settings.

Finally, in the **inference stage**, a mask layout together with its condition specifications is fed into the finetuned model, which outputs a corrected, MPCed GDS. This three-stage design ensures both generalization across layouts and adaptability to diverse manufacturing conditions.

D. Physics-Informed Multimodal MPC Framework

We propose a **multimodal MPC framework** that integrates geometry correction and process adaptation in a closed loop. The MGT backbone provides geometry-based refinements, the condition network injects e-beam–dependent constraints, and a fusion layer reconciles the two. This unified system ensures 0.1 nm EPE tolerance and enhanced image contrast, while scaling efficiently to full-chip correction tasks.

IV. Experiments

A. Experimental Settings and Benchmarks

All experiments were conducted on a workstation equipped with an NVIDIA H100 GPU (80GB). The software stack consists of Python 3.9.18, PyTorch 2.1.1, and CUDA 12.3. To ensure a comprehensive and fair evaluation, we curated a diverse suite of benchmarks derived from both public and private datasets.

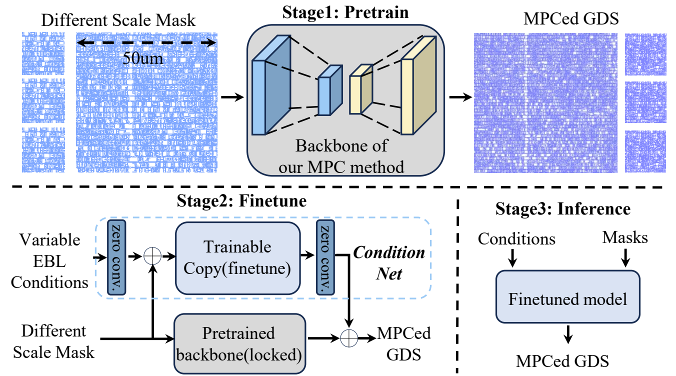


Fig. 4 Illustration of training and inference flow.

The framework's reliability and performance were validated using the ICCAD 2025 Post-OPC Mask Dataset [19], which contains diverse test patterns and complex CPU layouts. To establish a golden reference, a state-of-the-art commercial MPC tool was employed to generate ground-truth data. A primary dataset of 10,000 pairs of $10\mu m \times 10\mu m$ sized masks were generated for training the core point cloud model, with each pair consisting of a pre-MPC layout and its corresponding post-MPC version. To enable the multi-modal conditioning, this process was extended. Three distinct EBL process conditions were defined. For each condition, a unique resist image was simulated from an industry-standard test pattern, serving as the process fingerprint. The commercial tool was then used to create three separate, process-specific datasets. Each of these datasets contains 10,000 training instances, where the input is the resist image of the test pattern and the pre-MPC mask, while the label is the post-MPC mask.

For evaluation, a representative benchmark suite was randomly selected from the ICCAD 2025 dataset, consisting of 15 simple test patterns and 15 challenging CPU mask layouts. The 15 test patterns comprise three instances each of the following five critical test patterns, including 1D lines (M1-M3), line-end-to-line (M4-M6), line-end-to-line-end (M7-M9), U-shaped corners (M10-M12), and Z-shaped corners (M13-M15).

B. Comparisons with SOTA works

The proposed algorithm's performance was benchmarked against a baseline without MPC, a pixel-level optimization MPC method based on the pixel EBL simulator in [12], and a state-of-the-art commercial tool. The evaluation, detailed in TABLE I, focused on two key metrics: EPE for accuracy and turnaround time for performance.

In terms of accuracy, our model demonstrates EPE performance that is highly competitive with the commercial tool. On the test pattern suite, our framework achieves an average EPE of 0.16 nm, slightly outperforming the 0.17 nm of the commercial tool. For the more complex CPU layouts, our model achieves an average EPE of 0.31 nm, identical to the commercial reference. In contrast, while the pixel-based method offers a 35% EPE improvement over the baseline, its 1 nm minimum resolution makes achieving sub-nanometer

TABLE I Comparison of different methods on representative $10\mu m \times 10\mu m$ benchmarks in terms of EPE (nm) and TAT (s).

Testcase	w/o MPC	Pixel MPC [12]		C. T.		Ours	
	EPE	EPE	TAT	EPE	TAT	EPE	TAT
M1	1.54	0.53	124.88	0.10	8.79	0.12	0.086
M2	2.56	0.37	124.87	0.21	8.69	0.03	0.087
M3	1.22	0.43	124.88	0.13	9.64	0.08	0.087
M4	2.07	0.62	125.34	0.24	8.63	0.34	0.069
M5	2.29	0.84	124.79	0.18	8.67	0.13	0.087
M6	2.18	0.69	124.88	0.22	8.57	0.13	0.087
M7	2.48	1.24	124.83	0.23	8.68	0.16	0.086
M8	2.59	0.85	124.87	0.26	8.65	0.19	0.087
M9	2.10	0.68	124.88	0.25	8.59	0.15	0.083
M10	1.01	0.51	124.88	0.07	8.58	0.29	0.078
M11	0.99	0.64	124.80	0.04	8.71	0.07	0.086
M12	0.87	0.45	124.86	0.05	8.67	0.20	0.087
M13	1.85	0.58	124.80	0.24	8.64	0.17	0.087
M14	1.82	0.64	124.88	0.30	8.65	0.22	0.086
M15	0.79	0.73	124.92	0.06	8.64	0.16	0.085
Average	1.76	0.65	124.89	0.17	8.72	**0.16**	**0.085**
Ratio	10.35	3.82	14.32	1.00	1.00	**0.94**	**0.01**

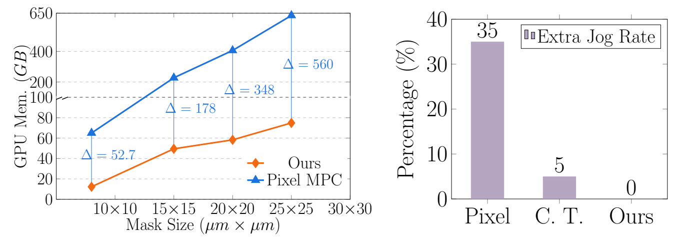
(a) Test results of test pattern benchmarks.

Testcase	w/o MPC	Pixel MPC [12]		C. T.		Ours	
	EPE	EPE	TAT	EPE	TAT	EPE	TAT
CPU1	1.70	0.46	124.82	0.31	8.58	0.14	0.078
CPU2	1.49	0.43	126.12	0.19	8.57	0.12	0.090
CPU3	1.86	0.56	125.43	0.41	8.61	0.34	0.089
CPU4	1.23	0.41	125.21	0.30	8.59	0.26	0.088
CPU5	1.22	0.52	125.16	0.31	8.57	0.33	0.086
CPU6	1.48	0.54	125.05	0.24	8.64	0.42	0.086
CPU7	1.79	0.72	125.39	0.25	8.54	0.43	0.086
CPU8	1.53	0.64	125.33	0.21	8.59	0.46	0.084
CPU9	1.97	0.58	125.02	0.37	8.65	0.18	0.087
CPU10	1.63	0.71	125.57	0.48	8.58	0.50	0.087
CPU11	1.84	0.71	125.97	0.34	8.60	0.10	0.087
CPU12	1.89	0.48	125.23	0.42	8.54	0.24	0.087
CPU13	1.62	0.70	125.88	0.23	8.69	0.47	0.084
CPU14	1.47	0.57	125.78	0.27	8.59	0.33	0.085
CPU15	1.50	0.73	125.09	0.30	8.63	0.34	0.084
Average	1.62	0.58	125.53	0.31	8.60	**0.31**	**0.086**
Ratio	5.23	1.87	14.60	1.00	1.00	**1.00**	**0.01**

(b) Test results of CPU pattern benchmarks.

TABLE II Larger and full-tile MPC results comparison on Turn around time and speedup

Testcase	Points	TAT (s)		Speedup
		Commercial Tool	Ours	
$50\mu m \times 50\mu m$	0.3M	586.45	2.66	220×
$100\mu m \times 100\mu m$	1.3M	1327.87	3.49	380×
1mm × 1mm	0.12B	79081.89	99.76	792×
Avg	37.5M	26998.74	**35.30**	**764×**



(a) Comparison of GPU memory between our method and pixel-based methods during training.

(b) Comparison of Extra Jog rate of Pixel method, C. T. and our method.

Fig. 5 Efficiency and Manufacturability comparison results.

accuracy computationally prohibitive and ineffective for this task.

Regarding performance, our model provides a significant advantage, achieving over a 100x speedup compared to the commercial tool. As shown in TABLE II, this advantage scales with the size of the mask. For a $50\mu m \times 50\mu m$ layout, our model is 220x faster. This speedup becomes even more pronounced on larger layouts, reaching 792x for a $1mm \times 1mm$ mask. This scaling occurs because for smaller layouts, model initialization overhead constitutes a larger portion of the total runtime. On larger layouts, the massive parallel processing capability of the GPU for batch inference far outpaces the multi-core processing of the CPU-based commercial tool.

Beyond runtime advantages, our multimodal approach demonstrates substantial memory efficiency compared to pixel-based methods. As illustrated in Fig. 5(a), GPU memory consumption exhibits dramatically different scaling behaviors between the two approaches. The pixel-based MPC method suffers from quadratic memory growth with mask size, consuming 52.7 GB for a $10 \times 10\mu m$ mask and escalating rapidly to an impractical 650 GB for a $25 \times 25\mu m$ mask—far exceeding the capacity of even high-end GPUs like the H100 (80GB). In stark contrast, our method maintains remarkably efficient memory utilization with near-linear scaling. The incremental memory increases of approximately 15-18 GB between sizes reflect the sparse nature of our point cloud representation. At the largest tested size, our method achieves an 8.7× memory reduction (75 GB vs. 650 GB), while still comfortably fitting within standard GPU memory constraints.

This dramatic difference in memory efficiency stems from our sparse point cloud representation, which captures only the geometrically relevant boundary information rather than storing the entire pixel grid. While the pixel-based method becomes computationally infeasible beyond $15 \times 15\mu m$ masks on available hardware, our approach can readily scale to production-size masks. This memory efficiency, combined with the previously demonstrated 100× speedup, establishes our framework as a practical and scalable solution for production-scale MPC deployment.

In stark contrast, our method maintains nearly linear memory scaling, requiring only 15 GB for the $10 \times 10\mu m$ mask and increasing modestly to 75 GB for the $25 \times 25\mu m$ mask. This 8.7× memory reduction at the largest tested size en-

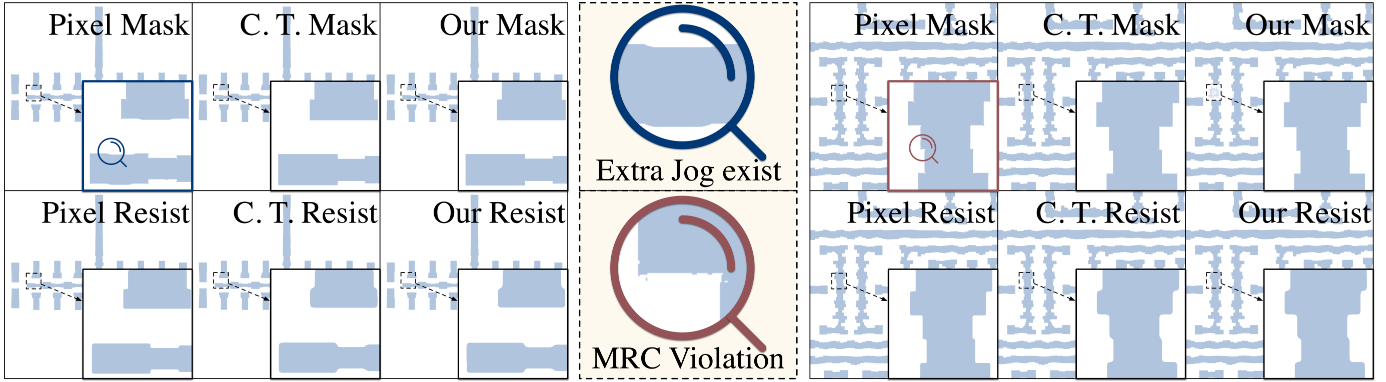


Fig. 6 Post-MPC results comparison: Mask fidelity and resist pattern quality for pixel-based, commercial, and multimodal MPC approaches.

ables processing of production-scale masks on standard GPU. The efficient memory footprint stems from the sparse point cloud representation, which captures only the geometrically relevant boundary information rather than the entire pixel grid. This memory efficiency, combined with the previously demonstrated $100\times$ speedup, establishes our framework as a practical solution for production-scale MPC deployment.

It is noteworthy that, as shown in Fig. 5(b), our method offers a significant advantage in terms of mask manufacturability, specifically concerning the creation of jogs. By strictly preventing the generation of new breaking points during the point cloud encoding stage, our approach is entirely jog-free. In contrast, a leading commercial tool, while aiming for superior correction, still introduced new jogs on 5% of the mask features. Our experiments indicate that these additional jogs yield no significant improvement in EBL quality. Moreover, they can potentially lead to Mask Rule Checking violations or require extra shots during the manufacturing process. Pixel-based methods, on the other hand, lack any mechanism to control jog generation, resulting in the introduction of new jogs on 35% of all edges. As visualized in Fig. 6, with results for the test pattern and CPU examples shown on the left and right respectively, the pixel-based mask introduces significant manufacturability issues. For the test pattern, it creates new, unwanted jogs in the post-MPC mask. In the CPU example, the corners are marred by rough edges and fine, fragmented pixels. Although these fragments can reduce EPE in simulation, their dimensions are far below the minimum manufacturable size. In contrast, our multimodal MPC achieves correction quality nearly identical to that of commercial software, ensuring manufacturability by producing clean patterns without introducing new jogs.

C. Ablation studies

An ablation study was conducted to validate the effectiveness and robustness of the multi-modal conditioning mechanism against process variations. For this analysis, the same benchmark suite of test patterns and CPU layouts as TABLE I was used. The model's average Edge Placement Error (EPE) was evaluated across three distinct EBL conditions, with the corresponding resist image provided as the conditional input for each run.

TABLE III Condition comparison results

Testcase	Condition 1 EPE	Condition 2 EPE	Condition 3 EPE
Test Pattern	0.16	0.15	0.16
CPU	0.31	0.31	0.33

As summarized in TABLE III, the framework delivers highly consistent EPE results across all tested conditions. The final correction accuracy shows negligible variation between the conditions for both simple and complex patterns. This result demonstrates that the model successfully generalizes from the process fingerprint, indicating its robustness to common manufacturing variations and the effectiveness of the proposed conditioning architecture.

V. Discuss and Conclusion

In this paper, we addressed the critical trade-off between accuracy and computational efficiency in mask process correction for advanced technology nodes. We introduced a novel multi-modal framework that, for the first time, successfully marries the precision of vector-based representations with the speed of GPU-accelerated deep learning. By representing mask geometry as a sparse point cloud and conditioning the model with a process fingerprint from a resist image, our method resolves the core limitations of both slow commercial tools and inaccurate pixel-based academic approaches. Experimental results confirm that our framework achieves accuracy comparable to state-of-the-art commercial tools while delivering a scalable speedup of over two orders of magnitude, significantly reducing mask synthesis turnaround time.

Promising future work includes adapting this multi-modal approach for other tasks like OPC, exploring semi-supervised methods to reduce data dependency, and integrating wafer-level feedback to further enhance fidelity.

Acknowledgment

This work is sponsored by Natural Science Foundation of Shanghai (Project No.25JD1403000)

References

[1] W. Y. Kwok, J. Yeap, S. Munoz, S. Chou, T. Sekiya, and H. Konnanur, "Model-driven rule-based mask process correction," in *Photomask Japan 2021: XXVII Symposium on Photomask and Next-Generation Lithography Mask Technology*, A. Ando, Ed., vol. 11908, International Society for Optics and Photonics. SPIE, 2021, p. 1190806.

[2] L. Pang and A. Fujimura, "Why the mask world is moving to curvilinear," *Journal of Micro/Nanopatterning, Materials, and Metrology*, vol. 23, no. 4, p. 041503, 2024.

[3] L. L. Pang, P. J. Ungar, A. Bouaricha, L. Sha, M. Pomerantsev, M. Niewczas, K. Wang, B. Su, R. Pearman, and A. Fujimura, "True-Mask ILT MWCO: full-chip curvilinear ILT in a day and full mask multi-beam and VSB writing in 12 hrs for 193i," in *Optical Microlithography XXXIII*, S. Owa, Ed., vol. 11327, International Society for Optics and Photonics. SPIE, 2020, p. 113270K.

[4] T. Lin, T. Donnelly, and S. Schulze, "Model based mask process correction and verification for advanced process nodes," in *Optical Microlithography XXII*, H. J. Levinson and M. V. Dusa, Eds., vol. 7274, International Society for Optics and Photonics. SPIE, 2009, p. 72742A.

[5] E. Sahouria, A. Bowhill, and S. Schulze, "Pattern based mask process correction: impact on data quality and mask writing time," in *Photomask and Next-Generation Lithography Mask Technology XII*, M. Komuro, Ed., vol. 5853, International Society for Optics and Photonics. SPIE, 2005, pp. 564 – 573.

[6] E. Tejnil, Y. Hu, E. Sahouria, S. Schulze, M. J. Tian, and E. Guo, "Advanced mask process modeling for 45-nm and 32-nm nodes," in *Optical Microlithography XXI*, H. J. Levinson and M. V. Dusa, Eds., vol. 6924, International Society for Optics and Photonics. SPIE, 2008, p. 69243H.

[7] W. Yao, H. Zhao, C. Hou, W. Liu, H. Xu, X. Zhang, J. Xiao, and J. Liu, "Efficient Proximity Effect Correction Using Fast Multipole Method With Unequally Spaced Grid for Electron Beam Lithography," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 1, pp. 218–228, 2023.

[8] I. Bork, P. Buck, B. Durvasula, V. Liubich, N. Rao, R. Sharma, and M. Zuo, "Curvature based fragmentation for curvilinear mask process correction," in *Photomask Technology 2021*, S. P. Renwick, Ed., vol. 11855, International Society for Optics and Photonics. SPIE, 2021, p. 118550R.

[9] G. S. Chua, W. L. Wang, B. I. Choi, Y. Zou, C. Tabery, I. Bork, T. Nguyen, and A. Fujimura, "Optimization of mask shot count using MB-MDP and lithography simulation," in *Photomask Technology 2011*, W. Maurer and F. E. Abboud, Eds., vol. 8166, International Society for Optics and Photonics. SPIE, 2011, p. 816632.

[10] P. Liao, Y. Zhao, D. Guo, Y. Lin, and B. Yu, "Analytical die-to-die 3-d placement with bistratal wirelength model and gpu acceleration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 6, pp. 1624–1637, 2024.

[11] E. H. Eriksen, A. Nazir, P. Balling, J. Vester-Petersen, R. E. Christiansen, O. Sigmund, and S. P. Madsen, "Dose regularization via filtering and projection: An open-source code for optimization-based proximity-effect-correction for nanoscale lithography," *Microelectron. Eng.*, vol. 199, no. C, pp. 52–57, Nov. 2018.

[12] A. Awad, C. Behroozi, and A. Erdmann, "Integrated mask process modeling for better yield predictions," in *Photomask Technology 2024*, S.-S. Kim and L. S. M. III, Eds., vol. 13216, International Society for Optics and Photonics. SPIE, 2024, p. 132161W.

[13] G. Chen, W. Chen, Q. Sun, Y. Ma, H. Yang, and B. Yu, "DAMO: Deep agile mask optimization for full-chip scale," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 9, pp. 3118–3131, 2022.

[14] H. Yang, S. Li, Z. Deng, Y. Ma, B. Yu, and E. F. Y. Young, "Gan-opc: Mask optimization with lithography-guided generative adversarial nets," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2822–2834, 2020.

[15] S. Sun, F. Yang, B. Yu, L. Shang, and X. Zeng, "Efficient ilt via multi-level lithography simulation," in *2023 60th ACM/IEEE Design Automation Conference (DAC)*, 2023, pp. 1–6.

[16] Z. Yu, G. Chen, Y. Ma, and B. Yu, "A gpu-enabled level-set method for mask optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 2, pp. 594–605, 2023.

[17] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point transformer v3: Simpler, faster, stronger," in *CVPR*, 2024.

[18] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," 2023.

[19] "Lmlitho," 2024. [Online]. Available: https://github.com/LMLitho/LMLitho