

Automated Curvy Waveguide Routing for Large-Scale Photonic Integrated Circuits

Hongjian Zhou¹, Keren Zhu², Jiaqi Gu^{1†}
¹Arizona State University, ²Fudan University
†jiaqigu@asu.edu

ABSTRACT

As photonic integrated circuit (PIC) designs advance and grow in complexity, largely driven by innovations in photonic computing and interconnects, traditional manual physical design processes have become increasingly cumbersome. Available PIC layout automation tools are mostly schematic-driven, which has not alleviated the burden of manual waveguide planning and layout drawing for engineers. Previous research in automated PIC routing largely relies on off-the-shelf algorithms designed for electrical circuits, which only support high-level route planning to minimize waveguide crossings. It is not customized to handle unique photonics-specific routing constraints and metrics, such as curvy waveguides, bending, port alignment, and insertion loss. These approaches struggle with large-scale PICs and cannot produce real layout geometries without design-rule violations (DRVs). This highlights the pressing need for electronic-photonic design automation (EPDA) tools that can streamline the physical design of modern PICs. In this paper, *for the first time*, we propose an open-source automated PIC detailed routing tool, dubbed APR, to generate DRV-free PIC layout for large-scale real-world PICs. APR features a grid-based curvy-aware A* engine with adaptive crossing insertion, congestion-aware net ordering and objective, and crossing-waveguide optimization scheme, all tailored to the unique property of PIC. On large-scale real-world photonic computing cores and interconnects, APR generates a DRV-free layout with 14% lower insertion loss and 6.25× speedup than prior methods, paving the way for future advancements in the EPDA toolchain. Our codes are open-sourced at [link](#).

1 INTRODUCTION

In recent years, as silicon photonics advances, photonic integrated circuits (PICs) have received significant attention among researchers due to the characteristics of high-speed and low-power dissipation. There are various designs and demonstrations on photonic tensor cores (PTCs) for optical neural networks (ONNs), and photonic network-on-chips (NoCs) for high-bandwidth chip communications. Due to those main driving research areas, PICs exhibit an exponential increase in complexity. As shown in Fig. 1, the number of photonic components on a single chip is rapidly approaching the order of 1000 components per chip and is expected to double every 2.5 years [1]. There is an increasing demand for electronic-photonic design automation (EPDA) toolkits to automate layouts, improving both productivity and solution quality.

Traditionally, PIC physical design is schematic-driven [2]. In this approach, components are placed and interconnected according to the circuit topology and signal paths in the schematic, aiming to minimize crossings, detours, and bending, which helps reduce insertion loss and improves signal integrity. In certain cases, routing

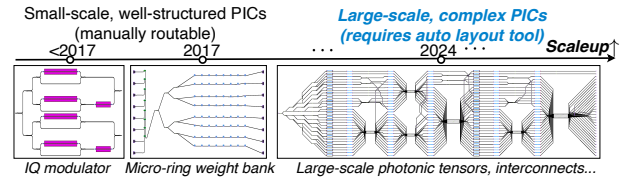


Figure 1: Modern PIC scale and complexity require EPDA.

can be **manually managed**, particularly when the circuits are *highly structured with a well-designed no-crossing topology*, such as in crossbar arrays [3] or triangular/rectangular meshes [4], binary tree structure [5]. When these designs are *optimally placed with large spacing and perfectly aligned ports*, device abutment or simple straight waveguides can automatically connect ports, similar to the standard cell-based layout used in SRAM arrays.

However, significant routing challenges arise, requiring PIC routing automation when: ❶ the circuit **scale exceeds manual capabilities**, such as with hundreds or thousands of instances/nets; ❷ the circuits have **complicated topology or are not perfectly placed**, leading to issues like port offsets, numerous crossings, and routing congestion; ❸ the design needs to **adapt to different fabrication processes or device designs**, each with varying component sizes and properties, necessitating adjustments in waveguide routing; ❹ frequent updates or design iterations make manual modifications inefficient, particularly when schematic designers **lack full visibility into waveguide routing and where crossings need to be inserted**. This often leads to repeated back-and-forth between schematic and layout design, especially during layout design space exploration with iterative placement and routing.

Most of the existing work focuses on PIC global route planning. Optical routing algorithms [6–8] are proposed for on-chip 3D system-on-package designs, primarily aiming at optimizing the signal loss and total power. PROTON [9] and PLATON [10] are automatic place-and-route tools where a modified Lee’s algorithm is used for optical waveguide routing. In [11], the insertion loss is further reduced by optimizing device flipping and rotation to minimize crossings. Those existing global routing approaches primarily **focus on planning/finding paths** that minimize path loss, but often **overlook the physical implementation** of these paths. This can result in issues like *path congestion, failure to insert crossings or bends*, ultimately leading to an *invalid routing solution*.

There is also work focused on completing the detailed routing stage. Prior work [12] solved global routing using mixed integer programming, followed by a Manhattan grid-based detailed routing using a left-edge algorithm, with crossings treated as design constraints. However, this grid-based approach limits waveguide bends to 90°. A subsequent approach [13] introduced non-Manhattan

channel routing to handle optical waveguide curves. Yet, since current PICs typically use only a single optical waveguide layer, *crossings are inevitable*, leaving little room for optimization during detailed channel routing. To address these limitations, it is essential to have an automated PIC router that is **fully aware of the physical instantiation and design rule** of waveguides and components with smart crossing insertion.

In this work, we propose an automated PIC detailed routing tool featuring non-Manhattan curvy waveguide handling and adaptive crossing insertion. Our framework addresses key limitations of existing methods by not only optimizing path insertion loss but also considering waveguide geometry and layout constraints during routing. By **adaptively inserting crossings**, rather than manually pre-inserting them in the schematic, and **considering the actual geometry** of waveguides, crossings, and bends, APR can generate **complete and design-rule violation (DRV)-free layout in minutes**, minimizing the need for extensive post-routing adjustments or iterative schematic/layout modification.

Highlights of this work are summarized as follows.

- We devise a fully automated PIC detailed routing tool APR that delivers DRC-free, low insertion loss routed PIC layout for large-scale circuits in minutes, supporting curvy waveguide geometry and automatic crossing insertion.
- **Curvy-Aware Non-Manhattan A* Router**: A customized curvy-aware A* search with adaptive neighbors to support different types of curvy structures.
- **Accessibility-Enhanced Port Assignment**: We introduce synergistic strategies to improve PIC routability by enabling orientation-aware port access and reserving space in port-congested regions.
- **Congestion-Penalized RR with Grouped Net Order**: We propose group-based net order, group congestion penalty, and local rip-up & reroute (RR) strategies to minimize waveguide crossings for better routability.
- On large-scale PTC and oNoC benchmarks, our proposed APR generates DRV-free layouts with 14% lower insertion loss (dB) and 6.25× faster runtime than prior methods.

2 PRELIMINARIES

In this session, we will first give a brief background on related VLSI routing methods, and the differences between PIC routing and VLSI routing by discussing PIC design rules. Following that, we will outline the traditional manual PIC routing flow. Finally, we will present the evaluation metrics for PIC routing and define the specific challenges associated with it. The notations used in this paper is summarized in Table 1.

2.1 VLSI Detailed Routing

Detailed routing faces challenges such as complex design rules, pin access, and limited routing resources [14]. Common VLSI routing methods, for example [15–18], utilize path-finding algorithms such as A* search or maze routing, supported by a DRC engine. A classic and still popular paradigm for resolving competition over routing resources between different nets is *negotiation-based routing* [19]. In these approaches, a rip-up and reroute scheme is utilized to clear routing failures.

Table 1: Notations used in this paper.

Symbol	Description
N	The set of nets specified in the circuit netlist.
n_i	The i^{th} net in N , $1 \leq i \leq N $.
P	The set of all paths.
p_i	The i^{th} path in P , $1 \leq i \leq P $.
$IL(p_i)$	The insertion loss of the p_i .
IL_{max}	The maximum insertion loss over all paths.
$IL_{wg}(p_i)$	The propagation loss of the path.
$IL_{cr}(p_i)$	The crossing loss of the path.
$IL_{bn}(p_i)$	The bending loss of the path.
$\alpha_w, \alpha_c, \alpha_b$	Coefficient of $IL_{wg}(p_i)$, $IL_{cr}(p_i)$, and $IL_{bn}(p_i)$.
g_i	The i^{th} port group.
w_{g_i}	Check region of group-based congestion penalty.
λ_c	The coefficient of group-based congestion penalty.
s	Routing grid size.

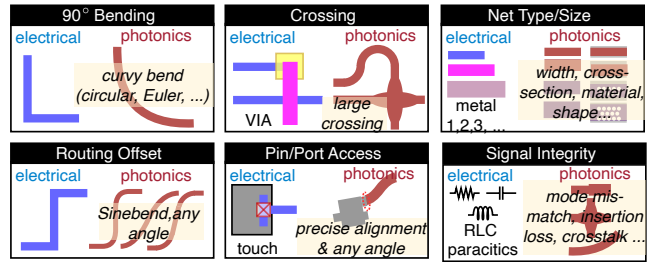


Figure 2: Compare properties/rules of EIC and PIC routing.

A key distinction between VLSI and PIC routing is the routing direction. VLSI routing is usually Manhattan or even unidirectional, while PIC routing necessitates curvy waveguides. Diverse routing directions, including octagonal routing, are employed in analog [20, 21], PCB [22–24], and package routing [25, 26]. Nevertheless, research on curvy path routing remains limited.

2.2 Photonic Design Rules

In the PIC routing problem, we typically operate with a single silicon waveguide routing layer, where photonic devices are considered as obstacles. The waveguides form port-to-port optical paths, resulting in all nets being 2-pin nets. Here, we provide a brief overview of PIC routing design rules and highlight the unique considerations specific to photonic circuits.

2.2.1 Waveguide Spacing. Waveguides need proper spacing with each other and photonic device structures to avoid crosstalk from unwanted coupling. Due to the diverse types/sizes of waveguides, as shown in Fig. 2, the minimum spacing rule between two nets depends on many factors, e.g., wavelength, polarization mode, refractive index contrast, substrate type, waveguide cross-sections. For example, for high-index contrast systems (such as silicon-on-insulator), small spacings (e.g., 1-3 μm) are sufficient.

2.2.2 Bend Radius. In photonic circuits, the bend radius is a key parameter that has a huge difference from the 90° metal wire bend in VLSI. Sharp bends in photonic waveguides can cause significant mode mismatch and radiation losses. To mitigate these losses, the waveguide bend structure typically forms a smooth curve, such as

a circular or Euler bend for 90° turns and a sine bend for routing offset, with sufficient curvature to ensure proper light confinement and minimize loss, as shown in Fig. 2.

The bend radius in photonic circuits can vary widely, typically ranging from a few microns to millimeters, depending on factors like material properties, bending structure, and refractive index contrast. Silicon waveguides with high refractive index contrast can support small bend radii, typically around $5\text{-}10\ \mu\text{m}$. Silicon nitride waveguides, with lower refractive index contrast, require larger bend radii, generally $20\text{-}100\ \mu\text{m}$, depending on waveguide geometry and application. While a larger bend radius minimizes insertion loss, it also consumes more chip area and routing resources, which can potentially cause routability issues.

2.2.3 Waveguide Crossing. Unlike VLSI routing that forbids wire crossings and uses vias for layer transitions, photonic circuits enable waveguide crossings (CRs) on the same layer. CRs are often essential especially for dense circuits. However, each CR introduces insertion loss, typically ranging from $0.1\ \text{dB}$ to $1\ \text{dB}$, and occupies a footprint of about $\sim 5 \times 5\ \mu\text{m}^2$. Moreover, the angle at which waveguides intersect is crucial in minimizing crosstalk. CRs require *perpendicular waveguide intersections to minimize crosstalk*, posing *challenges for routing dense PICs*, especially when parallel waveguides need sufficient space to adjust their relative orientation through curvy bending, as shown in Fig. 2.

2.2.4 Port Connection and Alignment. PIC connects waveguides via precise port abutment, which requires exact face-to-face alignment (180° orientation). Fig. 2 shows an example. Misalignment or offset between waveguides can lead to signal path failure, making *precise alignment* a critical requirement during PIC routing.

2.2.5 Signal Integrity. One of the most important metrics for PIC is insertion loss, which impacts the laser power budget and signal integrity (signal-to-noise ratio, crosstalk). The major evaluation metric for PIC routing is the maximum insertion loss on the critical path. Long waveguides and CRs introduces disturbance in signal integrity and are preferred to be avoided.

2.3 Schematic-Driven PIC Layout

Traditional PIC physical design workflows, including manual design and current available EPDA tools, are schematic-driven [27]. In this approach, all structures, including crossings and even each segment of a waveguide, are treated as separate instances in the netlist. Designers need to plan the routing ahead during schematic stage and manually insert crossings as instances to the netlist. Then, the nets in the schematic represent only port connectivity, eliminating the need for physical instantiation of nets, as all ports of waveguides are connected through abutment.

One significant drawback of the schematic-driven layout approach is that waveguide routing and crossings must be predetermined by design experts at the schematic stage, relying on empirical predictions of physical design solutions. Once established, these elements cannot be easily added or removed during routing, resulting in a **rigid, manually-defined routing topology**. This rigidity often causes back-and-forth modifications between the physical design and schematic stages, which can be inefficient. Moreover, it is **not scalable** to manually handle the routing of large-scale

PICs. To address this issue, **a new formulation of instances and nets is needed to decouple the schematic and physical design stages** while incorporating automated crossing insertion. This would allow for **greater flexibility and efficiency for scalable PIC auto-routing**.

2.4 PIC Routing Quality Metrics

In addition to regular routing metrics, such as wirelength, design rule violation, and runtime, one of the most important photonic-specific metrics is **critical path insertion loss (IL)** that impacts link power budget and signal-to-ratio ratio. IL is calculated based on the optical path which refers to the light propagation path through all cascaded components from the laser source to the photodetector. Assume a path p_i consists of instances and nets ($m_0 \rightarrow n_0 \rightarrow m_1 \rightarrow n_1 \rightarrow \dots$). Some nets and instances are shared across different paths. Note that for multi-port photonic devices, we assume the same IL from any input port to any output port given lack of accurate IL information from available free PDKs. Port-specific ILs can be easily considered in the same formulation. The insertion loss $IL(p_i)$ is defined as the sum of ILs of all devices $IL(m_j)$ and waveguide routes $IL(n_j)$ along the path in the decibel unit (dB) as a convention. For net IL, we will consider the crossing IL_{cr} , bending IL_{bn} , and propagation IL_{wg} losses in the instantiated waveguide routes. Therefore, we have:

$$IL(p_i) = \sum_{m_j \in p_i} IL(m_j) + \sum_{n_j \in p_i} IL(n_j)$$

$$\sum_{n_j \in p_i} IL(n_j) = IL_{wg}(p_i) + IL_{cr}(p_i) + IL_{bn}(p_i) \quad (1)$$

$$IL_{wg}(p_i) = \alpha_w WL_{p_i}, \quad IL_{cr}(p_i) = \alpha_c \#CR_{p_i}, \quad IL_{bn}(p_i) = \alpha_b \angle BN_{p_i},$$

where WL_{p_i} , $\#CR_{p_i}$, and $\angle BN_{p_i}$ are the total straight waveguide length, the number of crossings and total degree of bending along the path p_i , and coefficients α_w , α_c , α_b are the insertion loss per unit length/CR/angle for the specific photonic component structures. To achieve the desired optical functionality and signal-to-noise ratio for switching, modulation, or multiplexing, the insertion loss should be minimized.

The maximum insertion loss IL_{max} among all paths P determines how much extra power is required from the laser to ensure that enough light reaches the output photodetectors or subsequent stages in the circuit. Thus, IL_{max} is the main qualifier of PIC routing, and the objective function is given as:

$$IL_{max} = \max_{p_i \in P} IL(p_i) \quad (2)$$

2.5 Problem Formulation

We formally define the PIC detailed routing problem as follows.

PIC Detailed Routing. *Given a set of nets $N = \{n_i | 1 \leq i \leq |N|\}$, a set of placed devices $M = \{m_i | 1 \leq i \leq |M|\}$, generate a routing solution for each net $n_i \in N$ such that n_i is connected without design rule violations and minimize the IL_{max} .*

3 APR: AUTOMATED PIC DETAILED ROUTING

In this section, we present the details of our proposed APR framework, built on a customized grid-based A^* search algorithm. It efficiently finds curvy waveguide paths and inserts crossings automatically to minimize maximum insertion loss while honoring

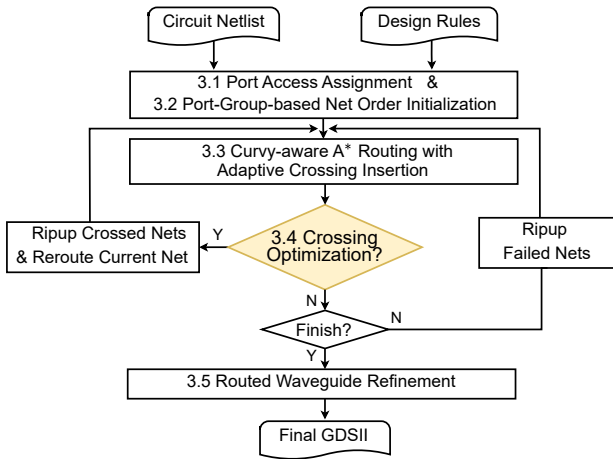


Figure 3: Algorithm flow of our APR framework.

design rules. The overall flow of our proposed framework is shown in Fig. 3. The core of our routing framework includes three main phases: ❶ *Port Access Assignment*: This phase assigns ports, considering orientation and density, to ensure smooth routing and minimize congestion; ❷ *Iterative Curvy-Aware Waveguide Routing*: This phase connects all nets with curvy-aware A* search following group-based net ordering, with a local rip-up and reroute (LRR) check to optimize crossings and comply with the design rules in Section 2; and ❸ *Route Refinement*: At the end of the routing stage, we refine the routing solution and generate a DRV-free GDS layout.

3.1 Accessibility-Enhanced Port Assignment

The port access problem is one of the most challenging subroutines in PIC detailed routing. Unlike the VLSI routing problem, where metal pins are unidirectional, PICs use *directional waveguide ports*, which have strict **access orientation and precise alignment requirements**. Ports must be accessed with waveguides in a specific face-to-face orientation (180°) and exact cross-section alignment, as shown in Fig. 4a. Accessing the target port with a wrongly-oriented waveguide fails to find a legal connection, as there may not be enough space near the port to adjust direction using curvy bends. When a waveguide passes near ports of other nets, accessing the *blocked* port becomes even more difficult. The primary reason for this port access challenge is the large area required to accommodate curvy waveguide bends. To solve the above challenges, we propose the following port access assignment techniques that account for both port orientation and port density, enhancing overall port accessibility.

Port Propagation. In PICs, some ports are located within the device bounding box. Since devices are treated as obstacles, we propagate these internal ports to the boundary of the device bounding box according to their orientation, as shown in Fig. 4a.

Bending-Aware Port Access Region Reservation. To prevent other waveguides from blocking port regions, grids in front of each port, along the port orientation, are reserved for the corresponding net, ensuring they cannot be crossed by other nets, as

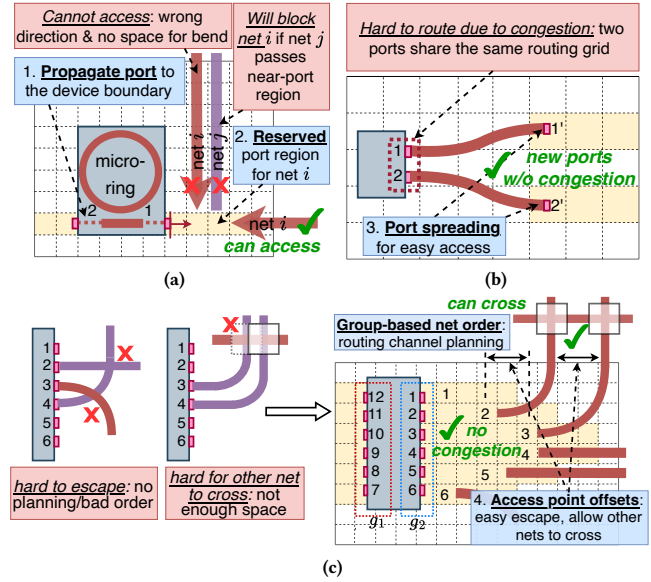


Figure 4: (a) Port propagation and reserved port region help port access. (b) Port spreading removes congested ports in the same grid. (c) group-based net order with access point offset enables channel planning and allows potential crossing.

shown in Fig. 4a. The size of the reserved region is adaptive to the waveguide’s bending radius, ensuring enough space for potential bends to maximize port access success while minimizing area.

Congested Port Spreading. In some PIC devices, high-density ports may occupy the same routing grid, causing port access difficulty. To address this, we *symmetrically* spread these access ports with a predefined extension length and spacing, as illustrated in Fig. 4b. The newly arranged ports will connect to the original ports using sine bends, ensuring they **occupy distinct routing grids to reduce congestion**. The reserved port access region will be updated to reflect the new port locations.

Channel Planning via Staggered Access Point Offsets. To enhance accessibility, we propose staggered access point regions for densely placed ports, as depicted in Fig. 4c. For instance, in multi-mode interference (MMI) devices with numerous ports on the same side, high port density can lead to access ports being obstructed by nearby waveguides. Parallel waveguides with narrow spacings prevent other nets from crossing over them, as inserting crossings requires sufficient space. We **progressively extend the access region length** for inner ports with an offset larger than a waveguide crossing size. This approach not only leaves enough bending space for inner ports to navigate out of congested regions but also facilitates the **placement of consecutive crossings**, allowing other waveguides to pass through parallel waveguides. This significantly decreases the chance of infeasible routing or excessive detours.

3.2 Port-Group-based Net Order

APR is a sequential router that processes nets one at a time. The order of net routing impacts the final routing quality and feasibility. We propose a port-group-based net ordering strategy that organizes ports on the same device based on their direction. Ports facing the

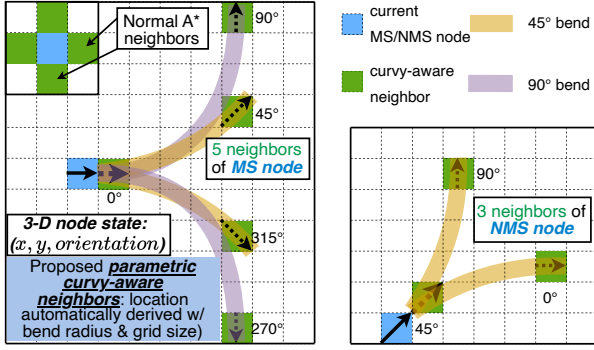


Figure 5: Parametric curvy-aware neighbors allow non-Manhattan curvy waveguide routing. Neighbors are automatically derived based on bending radius and grid size.

same direction are clustered together into groups (e.g., g_i). For example, as shown in Fig. 4c, the 0° and 180° ports in a device are divided into two port groups g_1 and g_2 . The **routing process is completed one group at a time**, ensuring that all nets within a group are routed before proceeding to the next group. The key insight behind this method is the observation that *most congestion and routing conflicts arise between nets within the same group*. By employing a group-wise routing approach, **nets are routed with awareness of others in the same group, minimizing intra-group conflicts** and improving overall routing quality. The routing order of a net n_i is given by the following priority score PR_{n_i} when a minimum-priority queue is used to manage all unrouted nets:

$$PR_{n_i} = dist_g + o_{n_i} = \min_{n_i \in g} dist_{n_i} + o_{n_i}, n_i \in g \quad (3)$$

where $dist_{n_i}$ is the Euclidean distance between two end ports of net n_i , and $dist_g$ denotes the smallest Euclidean distance among the nets within group g . A smaller max net distance of a group will result in a higher routing order for that group. The term o_{n_i} refers to the local order of net n_i within its group. This ensures that nets are routed based on their relative position in the group, helping to reduce conflicts between nets, particularly in multiport devices.

3.3 Non-Manhattan Waveguide Routing with Curvy-Aware A^* Search

In contrast to typical Manhattan VLSI routing, PIC designs typically employ non-Manhattan routing methods. Smooth curves decrease bending angles and waveguide lengths, thus reducing insertion loss. In this section, we present our iterative waveguide routing algorithm, designed to generate smooth waveguides with both 45° and 90° turns, while supporting adaptive crossing insertion.

3.3.1 Spacing-Ensured A^* Routing Grid Size Setting. The APR grid size s is set to be larger than the waveguide width. In typical PIC designs, waveguides are generally wider than ports. Setting s larger than the waveguide width maximizes pathfinding efficiency while facilitating easy port access.

3.3.2 Parametric Curvy-Aware Neighbor Candidate Generation. To efficiently enable curvy-aware A^* search, we propose parametric curvy-aware methods to generate neighbor candidates and perform comprehensive DRC check to select legal neighbors for exploration.

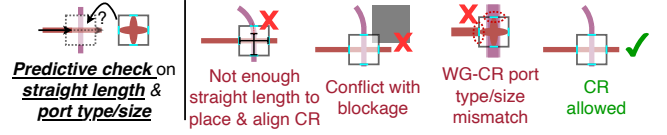


Figure 6: Proposed adaptive waveguide crossing insertion.

PIC uses curves instead of 90° or 45° turns in VLSI and PCB routing. We develop a customized curvy-aware neighbor generation scheme for the node based on parametric bending geometry. Each routing node is defined by its spatial location and orientation, which is crucial for accessing ports in the correct direction. We represent this as a **directional node** using $(x, y, orientation)$. As shown in Fig. 5, we derive neighbor candidates based on the current node's orientation and a user-defined bending radius. Based on their orientation, current nodes are categorized into two states: the Manhattan State (MS) and the Non-Manhattan State (NMS).

The MS nodes align with the x/y -axis and have five neighbors: one adjacent neighbor at 0° and four non-adjacent neighbors at $\pm 45^\circ$ and $\pm 90^\circ$. The NMS nodes are routed along the diagonal line with three neighbors. The **location of neighbor candidates is adaptively derived** based on the bend radius (r) and grid size (s). Larger radii and smaller grids result in larger step sizes in the grid. For instance, for an MS node in 0° , its adjacent neighbor is simply 1 grid away, and the steps of 90° and 45° neighbors are given by

$$\begin{aligned} step_{90,x} &= step_{90,y} = \lceil r/s \rceil, \\ step_{45,x} &= \lceil (\sqrt{2} - 1) \cdot r/s \rceil; \quad step_{45,y} = \lceil (1 - \frac{\sqrt{2}}{2}) \cdot r/s \rceil. \end{aligned} \quad (4)$$

In the neighbor generation process, we apply the ceiling function, $\lceil \cdot \rceil$, to ensure enough space for bending. Unlike 45° diagonal neighbors in traditional 8-way A^* , where $step_{45,x}$ and $step_{45,y}$ are equal, our approach **intentionally sets $step_{45,x}$ to be larger than $step_{45,y}$** , which prevents direct diagonal turns. This is motivated by the fact that, at the corner grid, the 45° bend will indent inward toward the center, occupying the inner grid and disrupting the straight part before the corner. A larger $step_{45,x}$ ensures that it does not rely on the previously established search path and leaves enough space for the 45° bend.

3.3.3 Geometry-Aware Neighbor Legality Check. To ensure that only feasible neighbors are considered for exploration, a legality check is necessary before adding them to the priority queue. A neighbor is legal only when the real geometry of the corresponding waveguide does not violate any design rules.

Hit No Obstacle: Geometry-Aware Spacing Check. If the neighbor does not hit an obstacle, we instantiate the real geometry of the connecting waveguide and perform a spacing check to ensure the route has no DRV.

Hit Routed Nets: Predictive Crossing Insertion. If a neighbor candidate hits a previously routed waveguide (marked as an obstacle), we need to check whether it is feasible to insert a waveguide crossing to pass through it.

As illustrated in Fig. 6, several critical constraints must be considered for crossing insertion: 1. **Enough straight waveguide length:** Waveguide crossings occupy specific chip areas, requiring adequate

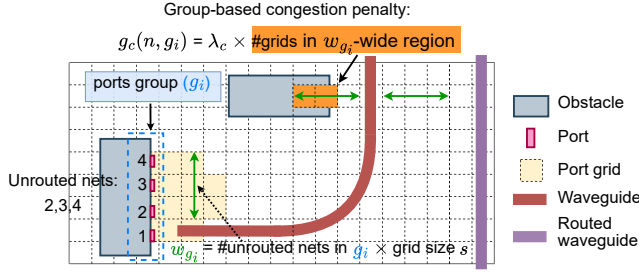


Figure 7: Group-based congestion penalty in Eq. (5).

spacing, and perpendicular orientations. Therefore, we check and ensure sufficient straight length and correct port orientation by checking the orientation state at each routing grid. ② *No conflict with blockages*: We will check whether the bounding box of the CR overlaps with any obstacles to honor design rules. ③ *Port matching*: For successful connectivity, the waveguides must align precisely with the four ports of the crossing. This includes matching properties such as cross-section, width, etc. By predictively checking all those legality conditions, we can adaptively incorporate crossing insertion during the routing process. This approach **reduces the need for long detours and avoids the complications associated with manually defined crossings** in the schematic.

3.3.4 Insertion Loss-Aware A* Search Cost. APR uses a customized A* search cost to consider insertion loss and optimize the algorithm efficiency. An A* search cost function $f(n)$ representing the cost of a path can be defined as $f(n) = g(n) + h(n)$, where $g(n)$ is the cost from the source (s) to the current node n , and $h(n)$ is the estimated cost from the current node to the target t . The formulation of $g(n)$ is divided into two parts, the insertion loss of current node $g_{IL}(n)$ which follow the calculation of Eq. (1) and the **group-based congestion penalty (GCP)** $g_c(n, g_i)$:

$$\begin{aligned} g(n) &= g_{IL}(n) + g_c(n, g_i), \\ g_c(n, g_i) &= \lambda_c \cdot \#grids(w_{g_i}), \end{aligned} \quad (5)$$

where λ_c is a penalty coefficient to **prevent the net from routing too close to the blockage or previously routed waveguides**, and $\#grids(w_{g_i})$ is the number of grids that occupied by others in the check region w_{g_i} as shown in Fig. 7. w_{g_i} is determined by the number of unrouted nets in its port group. As more nets are routed, w_{g_i} decreases to avoid consuming extra space. Empirically, w_{g_i} aids the routing process by **reserving resources for each port group**, thereby preventing other nets from entering the port area.

We further customize the heuristic cost function $h(n)$ to better estimate 45° and 90° bends, as shown in Eq. (6).

$$\begin{aligned} d_{min} &= \min(|n_x - t_x|, |n_y - t_y|), \\ d_{max} &= \max(|n_x - t_x|, |n_y - t_y|), \\ h(n) &= d_{max} - d_{min} + \sqrt{2} * d_{min} + \alpha \cdot IL_{bd,45}, \\ \alpha &= \begin{cases} 1, & \text{if } d_{min} > 0, d_{max} > 0 \\ 0, & \text{others} \end{cases}, \end{aligned} \quad (6)$$

where d_{min} is the minimum difference between the current node n and target node t along either the x-axis or y-axis, and d_{max} is the maximum one. The **insertion loss of 45° bend is added as a**

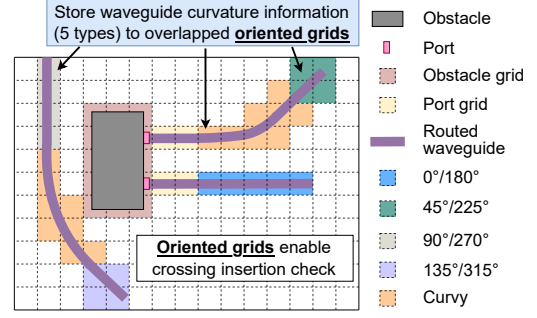


Figure 8: Represent routed waveguides in oriented grid map.

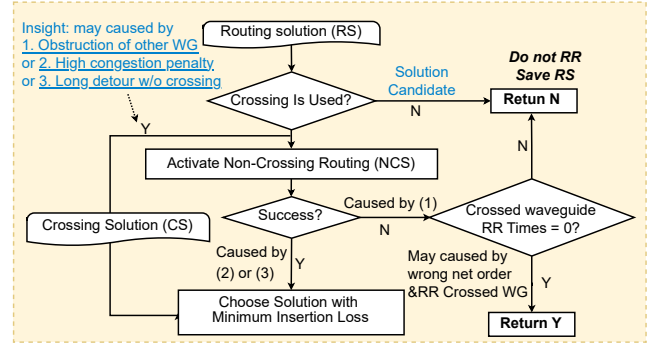


Figure 9: LRR check after finding a routing solution.

penalty since a non-zero d_{min} or d_{max} means there will be an orientation misalignment in the end of the path and makes it hard to connect to the target port.

3.3.5 Waveguide Instantiation. One of the largest differences of APR from prior global routing methods is **geometry awareness**. Once we obtain a path, we instantiate the curvy waveguide's real geometry with extrude function from GDSFactory [28] and store it on the **overlapped oriented routing grid map** accordingly as shown in Fig. 8. Later, the A* search engine can thereby treat the existing routed waveguides as obstacles and consider waveguide spacing check and crossing insertion conveniently.

3.3.6 Violated Net Removal. When accessing to the oriented target port is failed, APR apply a rip-up-and-reroute (RR) scheme. We relax DRC checking and record nets that conflict with the established paths. These nets are subsequently ripped up and rerouted in subsequent iterations. To avoid repeating the same routing results and to mitigate congestion, a history cost [29] is updated in the history map prior to net removal. Empirically, this history map-based negotiation process successfully resolves routing failures by balancing the demands of various nets.

3.4 Crossing-Waveguide Optimization

We propose a local ripup-and-reroute (LRR) scheme to further balance the waveguide length and CRs.

APR adopts group-wise net routing order and incorporates group-based congestion penalties. However, this strategy may cause longer-path nets within a group to be routed first, thereby blocking nets

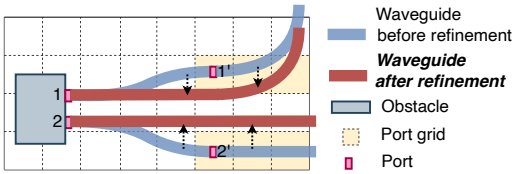


Figure 10: Waveguide refinement to remove bending.

in other port groups. Additionally, routing through congested areas and utilizing CRs often leads to divergent solutions, causing pathfinding to miss optimal routes. To address these issues, our LRR scheme performs both *crossing-enabled* and *crossing-disabled* routing attempts and selects the solution with lower IL.

The LRR evaluation is activated if a solution is found as shown in Fig. 9. If the current routing solution (RS) does not involve CRs, we will directly use it as the optimal path. Otherwise, it will be ripped up in a later stage. If CRs occur, possible reasons are (1) a blockage caused by another waveguide requiring a crossing, (2) a crossing chosen to bypass congestion, or (3) high propagation loss for non-crossing paths. To verify these three possibilities, a crossing-disabled routing (NCS) is then activated. If NCS finds a path without using CRs, the insertion losses of CS and NCS are compared, and the lower-loss path is selected. If NCS fails, it indicates the net is blocked. In this case, the blocking waveguide is assessed. If it has never been ripped up before, the blockage is likely caused by the group-based net order, and this blocking net will be ripped up, as it will not affect CR re-insertions in subsequent iteration. Our LRR strategy empirically optimizes the routing by **balancing long waveguides and CRs**.

3.5 Routed Waveguide Refinement

Since our grid-based routing method often results in the port center not aligning perfectly with the grid center, a slight offset can occur between the final path and the access port, as shown in Fig. 10. To resolve this, we adjust the initial and final segments of the waveguide path to align with the target device port, ensuring that the bend radius along the path remains unaffected. If this adjustment is not feasible, the waveguide will be connected to the port using a sine bend to maintain proper alignment.

4 EXPERIMENTAL RESULTS

4.1 Experimental Setting

The proposed photonic detailed routing framework is implemented in Python based on GDSFactory [28] libraries. All experiments are conducted on a personal workstation with an Intel i5-12560KF 3.7GHz CPU with 32GB memory.

Benchmarks. To assess the scalability of our proposed framework, we conduct experiments on different types of benchmarks: Photonic Tensor Cores (PTC) and Wavelength-routed Optical Network-on-Chip (WRONoC).

PTCs and WRONoCs have very different characteristics. Table 3 shows the benchmark statistics. PTC circuits have a more structured topology but have limited routing resources and high port density. For PTCs, we evaluate APR on Clements-style Mach-Zehnder interferometer (MZI) array [30] and auto-searched PTC ADEPT [31] with

Table 2: Device IL parameters used in IL_{max} evaluation.

Propagation α_w	Bending α_b	CR α_c	Y-branch	MZI	MMI
1.5 dB/cm [33]	0.005 dB [33]	0.52 dB [9]	0.3 dB [34]	1.2 dB [35]	0.1 dB [36]

Table 3: Benchmark PIC information.

Benchmark	#Devices	#Nets	Die Size	Waveguide Width	Grid Size
Clements_8x8 [30]	52	79	4800x1600 μm^2	0.5 μm	0.2 μm
Clements_16x16 [30]	168	287	8000x3200 μm^2	0.5 μm	0.2 μm
ADEPT_8x8 [31]	82	111	4400x1600 μm^2	0.5 μm	0.2 μm
ADEPT_16x16 [31]	162	223	6900x3200 μm^2	0.5 μm	0.2 μm
ADEPT_32x32 [31]	318	446	13000x6400 μm^2	0.5 μm	0.2 μm
Routers [37]	9	16	10000x10000 μm^2	2 μm	50 μm

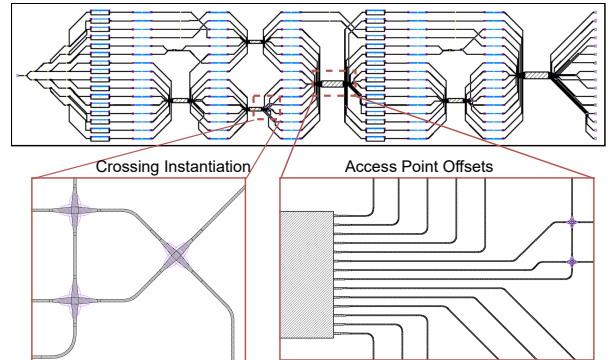


Figure 11: Layout of ADEPT_16x16 [31] routed by our APR.

different scales. The bend radius is set by 5 μm for single-mode Si waveguides (width=500 nm). WRONoC circuits, on the other hand, occupy a large die area and have unstructured interconnection topology. For WRONoCs, we conduct experiments on optical router benchmarks [32] with all the optical switches centered in the layout. Based on the positions of the memory controllers, we have four cases for this benchmark. The bend radius is set by 60 μm for its huge routing resource. Placement solutions of all benchmark circuits are designed manually by an experienced designer and verified with simulation using GDSFactory and KLayout. To evaluate the critical path IL, we summarize the device IL used in Table 2.

Baselines. We compare our APR with a prior method PROTON [9]. Note that the original PROTON mainly focuses on path planning and crossing optimization with adaptive crossing penalty, which cannot generate real waveguide geometry. For a fair comparison, we adapt PROTON by adding reserved port regions and a global ripup and reroute scheme to make it applicable to PIC detailed routing problems. Two variants of the adapted PROTON are: (1) the original implementation with global RR scheme (**Base-1**) and (2) additional 45-degree bend neighbors with more rip-up and re-route iterations to address accessing problem (**Base-2**).

4.2 PIC Routing Quality Evaluation

We compare APR with PROTON [9] in terms of critical path insertion loss IL_{max} , the critical path length, the number of crossings on the critical path, design rule violations (DRV), and wall-clock runtime. Table 4 shows that our APR can generate **DRV-free layouts** on all benchmarks with an average of **14% lower critical path IL** and **6.25x speedup**.

Table 4: Comparisons of the maximum insertion loss value IL_{max} (dB), the path length with IL_{max} (WL (μm)), the number of crossings passed by the signal with IL_{max} , total design rule violations (DRV), and runtime (s). \downarrow : lower is better.

Benchmark	Base-1 (Adaptive crossing penalty) [9]					Base-2 (w/ Diagonal neighbors) [9]					APR				
	#CR	WL (mm)	IL_{max} \downarrow (dB)	DRV \downarrow	Time \downarrow (s)	#CR	WL (mm)	IL_{max} \downarrow (dB)	DRV \downarrow	Time \downarrow (s)	#CR	WL (mm)	IL_{max} \downarrow (dB)	DRV \downarrow	Time \downarrow (s)
Clements_8x8	0	3.39	16.99	0	113	1	3.01	16.82	0	258	0	2.94	16.38	0	32
Clements_16x16	5	5.06	29.32	12	580	2	4.34	27.52	4	274	0	4.38	26.74	0	164
ADEPT_8x8	16	4.7	17.12	26	179	18	4.16	17.46	17	249	18	4.1	18	0	98
ADEPT_16x16	28	7.84	24.07	98	1306	17	7.66	18.36	26	2627	16	7.38	17.8	0	243
ADEPT_32x32	66	16.13	44.57	355	9981	52	13.97	37.19	181	27140	50	15.04	36.34	0	1204
router_north	6	32.98	11.09	0	36	6	21.63	9.37	0	66	0	31.11	7.78	0	81
router_oneside	0	18.71	5.89	0	5	4	20.96	8.26	0	33	0	21.55	6.31	0	37
router_corner	8	20.81	10.23	1	54	9	16.7	10.1	0	75	0	35.29	8.4	0	59
router_pairwise	7	28.49	10.94	1	48	8	19.52	10.05	0	66	0	33.52	8.14	0	65
Geo-mean	-	15.34	18.91	-	1367	-	12.44	17.24	-	3421	-	17.26	16.21	-	220
Ratio	-	1	1	-	1	-	0.81	0.91	-	2.5	-	1.12	0.86	-	0.16

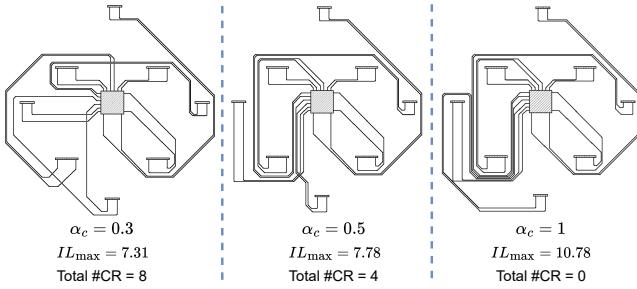


Figure 12: Layout of router_north of different crossing loss.

Analysis of PTC Results. The PTC benchmarks, featuring limited routing resources and high port density, provide a strong validation for a router’s ability to place bends and crossings while successfully accessing the target ports. (1) The Clements-style MZI array features a highly structured mesh topology with no inherent topological crossings, but suffers from non-ideal placement issues such as misalignments, flipped devices, and limited routing space. Due to the stringent routing spaces to access ports, baselines introduce extra waveguide CRs and lead to DRVs. In contrast, our APR can find crossing-optimal ($\#CR=0$), DRV-free paths in much shorter runtime. (2) ADEPT PTC is even more challenging due to the high port density in multi-port MMI devices and numerous topological crossings. As the size of the PTC increases, baselines exhibit a sharp rise in DRV and runtime. APR shows superior **scalability**, consistently producing **DRV-free low-IL** layouts for large circuits with **2-22.5 \times faster runtime**. Figure 11 visualizes the DRV-free ADEPT_16 \times 16 layout generated by APR with real curvy waveguide geometry and instantiated crossings.

Analysis of WRONoC Results. WRONoC features a large chip area and unstructured interconnection topology, which makes it challenging for a router to explore the large search space. It is important to note the counter-intuitive trade-off between $\#CR$ and WL. In NoC benchmarks, where the die size is large, fewer CRs do not necessarily result in lower IL. Reducing CRs may cause considerably longer detours, increasing propagation loss and ultimately leading to a higher overall IL_{max} . Aside from the case Router_oneside, our APR exhibits the minimum IL_{max} across the remaining cases with crossing-optimal ($\#CR=0$), DRV-free layout.

Table 5: Ablation study of GCP with different crossing cost.

Metrics	High Crossing Cost $\alpha_c = 1$		Low Crossing Cost $\alpha_c = 0.3$	
	w/o GCP	APR	w/o GCP	APR
#CR	6	0	5	5
WL (mm)	20.72	31.11	25.11	26.04
IL_{max} \downarrow	15.21	10.78	7.18	7.31
DRV	0	0	1	0
Time (s)	129	73	261	197

4.3 Discussion

Non-Manhattan 45-Degree Routing. Compared to **Base-1**, **Base-2** achieves an average of 19% shorter critical path WL by introducing the 45-degree bend (diagonal neighbors), which validates the effectiveness of a non-Manhattan routing style in PICs.

Crossing-Disabled Routing (NCS). As shown in Table 4 (**Base-2** vs. APR), our proposed additional crossing-disabled routing trial (NCS) introduces an extra runtime penalty, but it reduces the overall runtime and leads to higher solution quality as it *mitigates the port access issue and leads to much fewer total RR iterations*.

Port-Group-based Congestion Penalty (GCP). We evaluate the benefits of our proposed group-based congestion penalty in optimizing crossings using CRs with different IL: high crossing IL with $\alpha_c = 1$ and low crossing cost with $\alpha_c = 0.3$ as shown in Table 5. When crossings have high IL, our method effectively avoids crossings, minimizing the maximum insertion loss (IL_{max}). For low-IL crossings, it opts for paths with shorter WL with more CRs to optimize IL_{max} . Without the group-based penalty, however, the algorithm turns out to increase CR usage as α_c rises, as it struggles to find a low- $\#CR$ path due to congestion from other nets. By applying our group-based penalty, **net conflicts are largely reduced**, enabling more efficient routing decisions with fewer crossings and lower IL. As shown in Fig. 12, our α_c factor serves as a flexible control knob, enabling users to adjust crossing insertion according to their preferences and specific PIC performance requirements, such as phase balancing and reduced crosstalk.

5 CONCLUSION

We introduce APR, an open-source automated detailed routing tool specifically designed for photonic integrated circuits (PICs). APR features a non-Manhattan curvy-aware A* search engine with

accessibility-enhanced port assignment, adaptive crossing insertion, congestion-aware group-based net ordering and objective, and crossing-waveguide optimization scheme to handle unique PIC routing constraints while optimizing critical path insertion loss. On large-scale PIC benchmarks, APR demonstrates its capability to generate DRV-free layouts with 14% lower insertion loss and a 6.25× speedup compared to prior approaches, which highlight APR’s potential to significantly advance EPDA for complex photonic systems, paving the way for more efficient, scalable PIC designs.

REFERENCES

- [1] Lars Thylén, Sailing He, Lech Wosinski, and Daoxin Dai. The moore’s law for photonic integrated circuits. *Journal of Zhejiang University-SCIENCE A*, 7: 1961–1967, 2006.
- [2] Twan Korthorst, Wim Bogaerts, Duane Boning, Mitch Heins, and Barton Bergman. Photonic integrated circuit design methods and tools. In *Integrated Photonics for Data Communication Applications*, pages 335–367. Elsevier, 2023.
- [3] Johannes Feldmann, Nathan Youngblood, Maxim Karpov, Helge Gehring, Xuan Li, Maik Stappers, Manuel Le Gallo, Xin Fu, Anton Lukashchuk, Arslan Raja, Junqiu Liu, David Wright, Abu Sebastian, Tobias Kippenberg, Wolfram Pernice, and Harish Bhaskaran. Parallel convolutional processing using an integrated photonic tensor core. *Nature*, 2021.
- [4] Yichen Shen, Nicholas C. Harris, Scott Skirlov, et al. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 2017.
- [5] Hossein Hashemi. A review of silicon photonics lidar. In *2022 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–8. IEEE, 2022.
- [6] Jacob Rajkumar Minz, Somaskanda Thyagara, and Sung Kyu Lim. Optical routing for 3-d system-on-package. *IEEE Transactions on Components and Packaging Technologies*, 30(4):805–812, 2007.
- [7] Duo Ding and David Z Pan. Oil: A nano-photonics optical interconnect library for a new photonic networks-on-chip architecture. In *Proceedings of the 11th international workshop on System level interconnect prediction*, pages 11–18, 2009.
- [8] Duo Ding, Yilin Zhang, Haiyu Huang, Ray T Chen, and David Z Pan. O-router: an optical routing framework for low power on-chip silicon nano-photonics integration. In *Proceedings of the 46th annual design automation conference*, pages 264–269, 2009.
- [9] Anja Boos, Luca Ramini, Ulf Schlichtmann, and Davide Bertozzi. PROTON: An automatic place-and-route tool for optical networks-on-chip. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 138–145. IEEE, 2013.
- [10] Anja von Beuningen and Ulf Schlichtmann. Platon: A force-directed placement algorithm for 3d optical networks-on-chip. In *Proceedings of the 2016 on International Symposium on Physical Design*, pages 27–34, 2016.
- [11] Yu-Kai Chuang, Kuan-Jung Chen, Kun-Lin Lin, Shao-Yun Fang, Bing Li, and Ulf Schlichtmann. Planaronoc: concurrent placement and routing considering crossing minimization for optical networks-on-chip. In *Proceedings of the 55th Annual Design Automation Conference*, pages 1–6, 2018.
- [12] Christopher Condrat, Priyank Kalla, and Steve Blair. A methodology for physical design automation for integrated optics. In *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 598–601. IEEE, 2012.
- [13] Christopher Condrat, Priyank Kalla, and Steve Blair. Channel routing for integrated optics. In *2013 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*, pages 1–8. IEEE, 2013.
- [14] Gracieli Posser, Evangeline F.Y. Young, Stephan Held, Yih-Lang Li, and David Z. Pan. Challenges and approaches in vlsi routing. In *Proceedings of the 2022 International Symposium on Physical Design, ISPD ’22*, page 185–192, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392105. doi: 10.1145/3505170.3511477. URL <https://doi.org/10.1145/3505170.3511477>.
- [15] Andrew B. Kahng, Lutong Wang, and Bangqi Xu. Tritonroute-wxl: The open-source router with integrated drc engine. *IEEE TCAD*, 41(4):1076–1089, 2022. doi: 10.1109/TCAD.2021.3079268.
- [16] Michael Gester, Dirk Müller, Tim Nieberg, Christian Panten, Christian Schulte, and Jens Vygen. Algorithms and data structures for fast and good vlsi routing. In *Proc. DAC*, pages 459–464, 2012. doi: 10.1145/2228360.2228441.
- [17] Haocheng Li, Gengjie Chen, Bentian Jiang, Jingsong Chen, and Evangeline F. Y. Young. Dr. cu 2.0: A scalable detailed routing framework with correct-by-construction design rule satisfaction. In *Proc. ICCAD*, 2019. doi: 10.1109/ICCAD45719.2019.8942074.
- [18] Fan-Keng Sun, Hao Chen, Ching-Yu Chen, Chen-Hao Hsu, and Yao-Wen Chang. A multithreaded initial detailed routing algorithm considering global routing guides. In *Proc. ICCAD*, 2018. doi: 10.1145/3240765.3240777.
- [19] L. McMurchie and C. Ebeling. Pathfinder: A negotiation-based performance-driven router for fpgas. In *International ACM Symposium on Field-Programmable Gate Arrays*, pages 111–117, 1995. doi: 10.1109/FPGA.1995.242049.
- [20] Ricardo M. F. Martins and Nuno C. C. Lourenço. Analog integrated circuit routing techniques: An extensive review. *IEEE Access*, 11:35965–35983, 2023. doi: 10.1109/ACCESS.2023.3265481.
- [21] Hung-Chih Ou, Hsing-Chih Chang Chien, and Yao-Wen Chang. Nonuniform multilevel analog routing with matching constraints. *IEEE TCAD*, 33(12):1942–1954, 2014. doi: 10.1109/TCAD.2014.2363394.
- [22] Tan Yan and Martin D. F. Wong. Recent research development in pcb layout. In *Proc. ICCAD*, pages 398–403, 2010. doi: 10.1109/ICCAD.2010.5654190.
- [23] Qinghai Liu, Qinfei Tang, Jiarui Chen, Chuandong Chen, Ziran Zhu, Huan He, Jianli Chen, and Yao-Wen Chang. Disjoint-path and golden-pin based irregular pcb routing with complex constraints. In *Proc. DAC*, 2023. doi: 10.1109/DAC56929.2023.10247728.
- [24] Shih-Ting Lin, Hung-Hsiao Wang, Chia-Yu Kuo, Yolo Chen, and Yih-Lang Li. A complete pcb routing methodology with concurrent hierarchical routing. In *Proc. DAC*, pages 1141–1146, 2021. doi: 10.1109/DAC18074.2021.9586143.
- [25] Min-Hsuan Chung, Je-Wei Chuang, and Yao-Wen Chang. Any-angle routing for redistribution layers in 2.5d ic packages. In *Proc. DAC*, 2023. doi: 10.1109/DAC56929.2023.10247899.
- [26] Chun-An Lee, Wen-Hao Liu, Gary Lin, and Tsung-Yi Ho. Delay-matching routing for advanced packages. In *Proc. ICCAD*, 2023. doi: 10.1109/ICCAD57390.2023.10323679.
- [27] Lukas Chrostowski, Zeqin Lu, Jonas Flückiger, James Pond, Jackson Klein, Xu Wang, Sarah Li, Wei Tai, En Yao Hsu, Chan Kim, et al. Schematic driven silicon photonics design. In *Smart Photonic and Optoelectronic Integrated Circuits XVIII*, volume 9751, pages 9–22. SPIE, 2016.
- [28] Joaquin Matres et al. Gdsfactory. <https://github.com/gdsfactory/gdsfactory>, 2024.
- [29] Wen-Hao Liu, Wei-Chun Kao, Yih-Lang Li, and Kai-Yuan Chao. Nctu-gr 2.0: Multithreaded collision-aware global routing with bounded-length maze routing. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 32(5):709–722, 2013.
- [30] William R. Clements, Peter C. Humphreys, Benjamin J. Metcalf, et al. Optimal Design for Universal Multiport Interferometers. *Optica*, 2018.
- [31] Jiaqi Gu, Hanqing Zhu, Chenghao Feng, Zixuan Jiang, Mingjie Liu, Shuhan Zhang, Ray T. Chen, and David Z. Pan. Adept: Automatic differentiable design of photonic tensor cores. In *Proc. DAC*, 2022.
- [32] Alexandre Truppel, Tsun-Ming Tseng, Davide Bertozzi, José Carlos Alves, and Ulf Schlichtmann. Psion: Combining logical topology and physical layout optimization for wavelength-routed onocs. In *Proceedings of the 2019 International Symposium on Physical Design*, pages 49–56, 2019.
- [33] Johnnie Chan, Gilbert Hendry, Aleksandr Biberman, and Keren Bergman. Architectural exploration of chip-scale photonic interconnection network designs using physical-layer analysis. *Journal of Lightwave Technology*, 28(9):1305–1315, 2010.
- [34] Devika Padmakumar Nair and Michaël Ménard. A compact low-loss broadband polarization independent silicon 50/50 splitter. *IEEE Photonics Journal*, 13(4):1–7, 2021.
- [35] Suguru Akiyama, Takeshi Baba, Masahiko Imai, Takeshi Akagawa, Masashi Takahashi, Naoki Hirayama, Hiroyuki Takahashi, Yoshiji Noguchi, Hideaki Okayama, Tsuyoshi Horikawa, et al. 12.5-gb/s operation with 0.29- ν cm v π 1 using silicon mach-zehnder modulator based-on forward-biased pin diode. *Optics express*, 20(3):2911–2923, 2012.
- [36] Michal Rakowski, Colleen Meagher, Karen Nummy, Abdelsalam Aboketaf, Javier Ayala, Yusheng Bian, Brendan Harris, Kate Mclean, Kevin McStay, Asli Sahin, et al. 45nm cmos-silicon photonics monolithic technology (45clo) for next-generation, low power and high speed optical interconnects. In *Optical Fiber Communication Conference*, pages T3H–3. Optica Publishing Group, 2020.
- [37] Zhidan Zheng, Mengchu Li, Tsun-Ming Tseng, and Ulf Schlichtmann. Topro: A topology projector and waveguide router for wavelength-routed optical networks-on-chip. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–9. IEEE, 2021.