Differentiable Tier Assignment for Timing and Congestion-Aware Routing in 3D ICs

Yuan-Hsiang Lu[1], Hao-Hsiang Hsiao[1], Yi-Chen Lu[2], Haoxing Ren[2], and Sung Kyu Lim[1]
[1]School of ECE, Georgia Institute of Technology, Atlanta, GA
[2]NVIDIA Research;
{yuan-hsiang.lu, thsiao, limsk}@gatech.edu; {yilu, haoxingr}@nvidia.com;

***Abstract*—State-of-the-art (SOTA) 3D physical design (PD) flows extend commercial 2D place-and-route (P&R) tools to enable signoff-quality 3D IC implementation through double metal stacking and inter-die metal layer sharing. While metal layer sharing introduces additional routing resources, the substantially higher manufacturing cost of face-to-face (F2F) inter-die vias compared to intra-die vias necessitates 3D-aware routing strategies to manage routability–cost trade-offs. To address this, we propose differentiable routing guidance for 3D ICs (DRG-3D), a GPU-accelerated differentiable optimization framework that provides routing guidance for 3D ICs. DRG-3D formulates a fully differentiable objective that simultaneously optimizes key 3D design metrics: routing congestion, wirelength, via cost, and F2F-via cost, which enables efficient and scalable gradient-based optimization over large-scale netlists. Experimental results show that DRG-3D outperforms the SOTA Pin-3D flow, achieving up to 8.37% reduction in routing overflow, 23.99% reduction in total negative slack (TNS), and 18.05% reduction in post-route timing violations.**

I. Introduction

With the diminishing returns of 2D scaling and the deceleration of Moore's Law, 3D integrated circuits (ICs) have emerged as a promising path to extend performance, power efficiency, and area (PPA) scaling through vertical stacking. As full-stack commercial 3D IC solutions remain under development, pseudo-3D approaches have become the state of the art (SOTA) by leveraging well-established 2D place-and-route (P&R) tools for practical 3D integration. Among these, Pin-3D offers significant advantages over traditional die-by-die optimization flows by incorporating full 3D-context routing and optimization. While 3D metal layer sharing routing in pseudo-3D flows introduces additional routing resources, the substantially higher manufacturing cost of face-to-face (F2F) inter-die vias relative to intra-die vias introduces a critical trade-off between routability and cost. However, the lack of global routing guidance specific to 3D integration often leads to poor via distribution, routing congestion, and post-route timing violations.

In parallel, global routing in 2D ICs has been extensively studied over the past decades. State-of-the-art routers [1], [2], [3], [4], [5], [6], [7], [8] typically project the routing problem onto a 2D plane consisting of 2-pin sub-nets, and subsequently restore the solution to the multi-layer routing space through a separate layer assignment [9], [10], [11], [12], [13], [14], [15] process. However, due to their inherently sequential nature, these methods are highly sensitive to net ordering and perform routing on a per-net basis without a holistic view of the design. This often results in suboptimal solutions driven by local decisions. Concurrent routing techniques—such as those based on integer linear programming (ILP)[16], [17], [18], [19]—offer improved solution quality by jointly considering multiple nets, but suffer from scalability limitations in large designs. [20] presents a promising alternative by relaxing the problem into a differentiable form [21], [22], [23], [24], [25], [26], [27], enabling scalable, GPU-accelerated [28], [29], [30], [31], [32], [33], [34], [35], [36] optimization via gradient-based methods. Recently, learning-based routing [37], [38], [39], [40] has demonstrated the potential

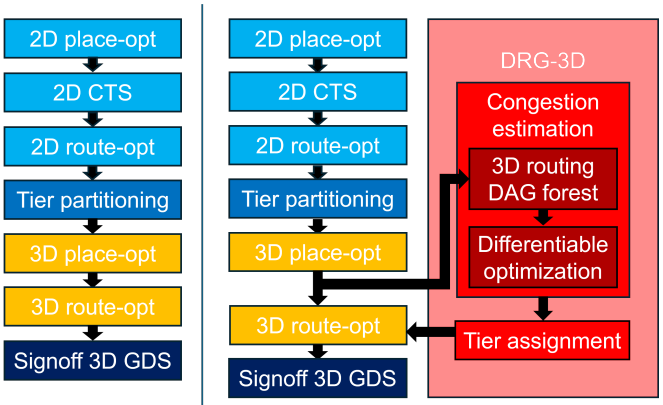

Fig. 1: SOTA Pin-3D [41] vs. our proposed DRG-3D

to model complex routing heuristics through data-driven approaches. Nevertheless, existing techniques are not directly applicable to 3D ICs, where routing must additionally consider vertical design constraints such as tier assignment, inter-die via minimization, and metal layer sharing. These challenges call for a new approach that effectively handles global routing in the 3D context.

In this work, we propose a novel framework, differentiable routing guidance for 3D ICs (DRG-3D), that provides routing guidance for 3D IC design flows, as illustrated in Figure 1. The flow begins by constructing a 3D routing DAG forest that models the complete 3D pattern routing space—comprising conventional 2D routing space along with candidate tier assignments in the vertical dimension. For each net, we first generate a rectilinear Steiner minimum tree (RSMT) to approximate the minimal wirelength topology in 2D. To extend this topology into the 3D context, we augment each Steiner point by duplicating it across both tiers—creating mirrored nodes on the top and bottom tier candidates. This tier duplication allows the DAG to encode all feasible topologies of the net within the 3D routing space, as illustrated in Figure 4. Each edge in the DAG represents a candidate 3D routing solution for a 2-pin sub-net, defined by a selected 2D L-shaped path and its associated tier assignment. This representation allows us to formulate the 3D global routing problem as the task of selecting one 3D routing candidate per 2-pin sub-net, aiming to minimize congestion overflow, total wirelength, via count, and face-to-face (F2F) via count. By probabilistically modeling the candidate selection, we construct a differentiable multi-objective loss function that can be directly optimized using gradient descent, with GPU acceleration enabled by deep learning toolkits. Following optimization, we extract net-level tier assignment information to serve as early-stage routing guidance for the subsequent stages of the flow.

Our contributions are summarized as follows:

- We propose DRG-3D, the first framework to provide timing- and

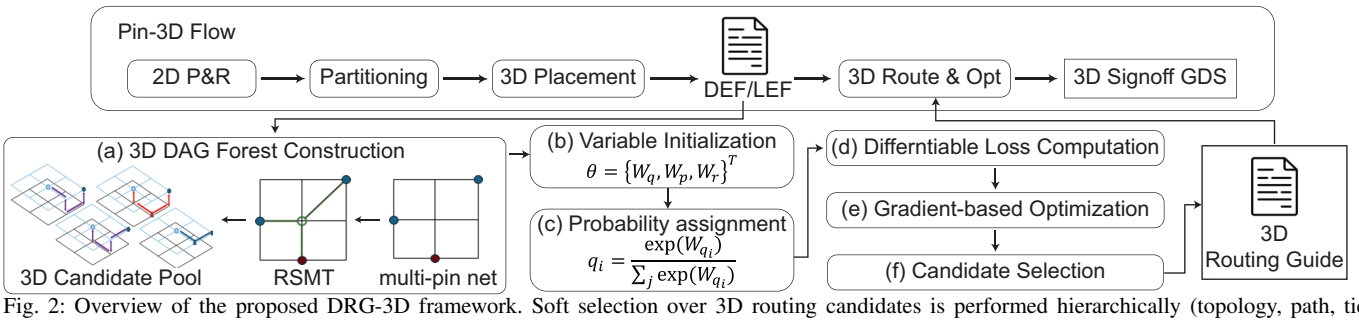

Fig. 2: Overview of the proposed DRG-3D framework. Soft selection over 3D routing candidates is performed hierarchically (topology, path, tier assignment), guided by softmax probabilities and differentiable loss minimization. Final routing decisions are derived from the highest-probability candidates as routing guidance to commercial tool.

congestion-aware routing guidance for SOTA 3DIC flows.

- We present the first differentiable multi-objective optimization framework for 3D ICs that enables concurrent exploration of routing solutions across all nets with GPU acceleration.
- Our method outperforms the SOTA 3D IC flow, reducing routing overflow by 8.37%, post-route total negative slack (TNS) by 23.99%, and timing violations by 18.05%, without increasing wirelength or F2F via count.

II. Preliminaries

A. Probabilistic Routing Resource Model

A gcell is a routing bin used in global routing, where the chip layout is partitioned into a grid to estimate routing demand, capacity, and congestion. Each gcell corresponds to a vertex v in the routing graph, and each edge e between adjacent gcells is assigned a *capacity* cap_e, defined as the number of available tracks minus pin density and local net congestion:

$$\text{cap}_e = \text{track}_e - \beta_v \cdot \text{pin\_density}_v - \text{local\_net}_e \tag{1}$$

The weight β_v [4] is a scaling factor derived from standard cell layout information. The *demand* d_e on edge e is calculated as the aggregated contribution from all 2-pin path candidates traversing that edge. Each candidate is weighted by its soft selection probability p_i and the probability of its associated routing tree $q_{tree(i)}$. To account for the via usage overhead, an additional term is included based on the number of turning points.

$$d_e = \sum_{i \in P_e} q_{\text{tree}(i)} p_i + \beta_v \sum_{k \in P_v} q_{\text{tree}(k)} p_k \tag{2}$$

Here P_e denotes the set of 2-pin path candidates that traverse edge e, and P_v denotes those with a turning point at vertex v.

To model the overflow cost in a differentiable form, [20] introduces a smooth overflow cost function $f(\text{cap}_e - d_e)$, where f is typically implemented as a $sigmoid$ or $ReLU$. This continuous formulation enables the estimation of expected routing cost—including wirelength and via count—under soft selection, allowing the routing problem to be optimized via gradient descent.

B. Routing DAG Forest for 2D IC

To compactly represent the routing solution space of a net, most global routing frameworks begin by constructing a Routing Directed Acyclic Graph (DAG), as illustrated in Figure 3(a–c). Each DAG corresponds to a distinct rectilinear Steiner minimum tree (RSMT) topology [42], [43], [44], typically generated using FLUTE [44], and is decomposed into a set of two-pin sub-nets. Within each DAG, vertices represent pins, Steiner points, or turning points, while edges denote feasible two-pin routing candidates, as shown in Figure 3(d).

III. Methodologies

A. DRG-3D Overview

We propose DRG-3D, a differentiable framework for routing guidance in 3D IC design flows, aimed at reducing congestion and improving post-route quality on top of the state-of-the-art pseudo-3D flow, Pin-3D. As shown in Figure 5, the framework performs iterative, gradient-based optimization over the full 3D routing candidate space. The overall process is illustrated in Figure 2 and detailed below.

1) **3D Routing Space Construction:** The flow begins by hierarchically constructing a 3D routing DAG forest, as described in earlier sections and shown in Figure 2(a). For each multi-pin net, we generate several 3D topology candidates by combining Steiner tree construction with mirrored duplication of Steiner points across tiers. Each topology is then decomposed into two-pin subnets, for which we derive 3D path candidates consisting of different combinations of 2D L-shaped paths and tier assignments.
2) **Variable Initialization:** As shown in Figure 2(b), for every candidate in the three-layer structure (topology, path, and tier assignment), we associate a trainable real-valued variable, randomly initialized.
3) **Probability Assignment via Softmax:** Each candidate group is normalized through a softmax function (as defined in Equations 4 and 5) to obtain a probability distribution over all routing candidates (Figure 2(c)). These probabilities are used to compute the expected contribution of each candidate to routing demand and cost.
4) **Loss Computation:** Using the selection probabilities P_i (as defined in Equations 6) of each candidate, we compute the expected cost values—OverflowCost, F2FViaCountCost, WireLengthCost, and ViaCountCost (Figure 2(d)). The overall differentiable loss function is then computed as shown in Equation 10.
5) **Gradient-Based Optimization:** As shown in Figure 2(e), we use the back-propagation algorithm to compute gradients of the total loss with respect to all parameters. The parameters are then updated using gradient descent.
6) **Convergence Check:** As shown in Figure 2(f), we repeat Steps 3 to 5 iteratively. The optimization process terminates when either the loss improvement falls below a predefined threshold δ or the maximum number of iterations $T_{\max}$ is reached:

$$|\mathcal{L}^{(t)} - \mathcal{L}^{(t-1)}| < \delta \quad \text{or} \quad t = T_{\max} \tag{3}$$

7) **Routing Guidance Deployment:** After convergence, we extract the most probable routing candidates to generate net-level guidance, including preferred tiers and routing paths. This guidance is integrated into the 3D P&R flow to improve routing quality.

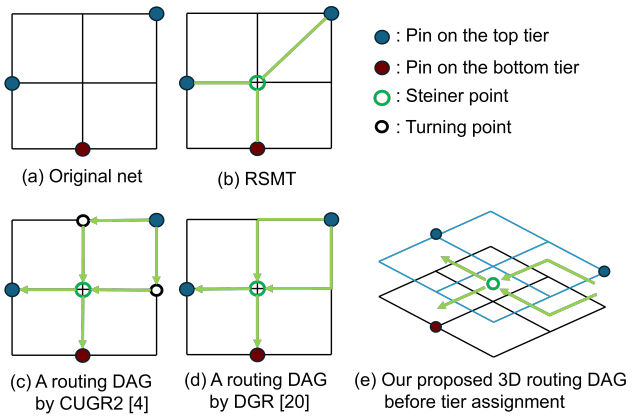

Fig. 3: Example of 3D routing DAG construction. Starting from FLUTE-generated RSMTs, multiple DAGs are formed. Unlike [45], [4], which splits edges at turns, we treat each edge as a full 2-pin path, extended with tier assignments in our 3D framework.

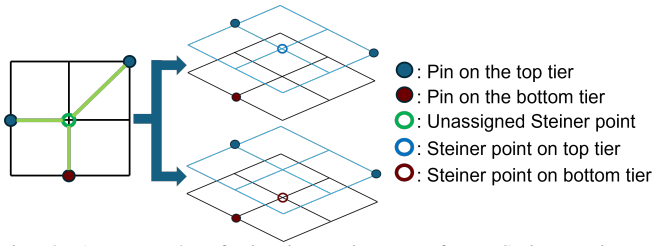

Fig. 4: An example of pin tier assignment for a Steiner point. By exhaustively enumerating feasible tier assignments for all Steiner points, the initial RSMT is expanded into multiple 3D topology candidates, each representing a unique configuration in the tier assignment domain.

B. 3D Routing DAG Forest Constructon

We construct a *3D routing DAG forest*, a data structure that extends the traditional 2D routing DAG forest by incorporating 3D tier assignments, as illustrated in Figure 3(e). For each multi-pin net, we first generate a 2D rectilinear Steiner minimum tree (RSMT) using FLUTE. To extend the topology into 3D, we introduce *mirror nodes* on both tiers for each Steiner point, which expands the initial RSMT into a set of *3D topology candidates*, as shown in Figure 4. Each 3D topology candidate is then decomposed into a set of two-pin subnets, which serve as the fundamental units for downstream routing.

For each 2-pin net, we adopt L-shape pattern to generate a set of 2D *path candidates*, as illustrated in Figure 5(b). Each 2D *path candidate* is then combined with a set of possible tier assignment scenarios for vias and trunks to produce corresponding 3D routing candidates, as shown in Figure 5(c). These candidates collectively define the 3D routing space. As a result, each net is associated with multiple 3D DAGs that represent distinct RSMT topologies and tier assignment combinations, forming a DAG forest that compactly encodes diverse routing alternatives.

C. Pattern Routing for 3D IC

With the 3D routing DAG forest in place, we formulate the 3D global routing problem as a hierarchical candidate selection: topology, path, and tier assignment. This structure enables comprehensive exploration of routing solutions across the entire netlist.

The first stage, referred to as the topology candidate stage (q-selection stage), involves choosing the global topology for each multi-pin net. Starting from RSMTs, we generate multiple 3D topology candidates by enumerating feasible tier assignments for Steiner points. Each candidate defines a distinct interconnection structure and is assigned a trainable weight. These weights are normalized via a softmax function to produce a probability distribution over the topology candidates, enabling differentiable selection. As shown in Figure 5(a), this stage determines how pins are globally connected in 3D space and forms the structural foundation for subsequent path and tier-level decisions. The second stage, referred to as the path candidate stage (p-selection stage), refines routing for each two-pin subnet derived from the selected topology. For each subnet, we generate multiple legal 2D L-shape path candidates capturing different geometric patterns. A trainable weight is assigned to each path, and softmax is applied to compute a probabilistic selection, as shown in Figure 5(b). The third stage, referred to as the tier assignment candidate stage (r-selection stage), determines how each selected 2D path is mapped to specific metal tiers in the 3D stack. Each path candidate is paired with multiple tier assignment options, specifying how to distribute its horizontal and vertical segments across tiers, as shown in Figure 5(c)

Together, the three levels—topology, path, and tier assignment—form a comprehensive candidate space for each net. The 3D pattern routing task is defined as selecting: (1) one topology candidate per net, (2) one path candidate per two-pin subnet within that topology, and (3) one tier assignment for each selected path. These steps correspond to the q-, p-, and r-selection stages, respectively—each associated with a trainable weight and softmax-normalized probability. This hierarchical selection defines a complete 3D routing realization for each net. By applying this selection process in parallel across all nets, we obtain a full-chip 3D global routing solution. This formulation enables a differentiable optimization process, where each candidate is treated as a soft (probabilistic) variable. This allows the use of gradient-based methods to achieve globally optimzed routing decisions across the entire design.

Figure 6 illustrates the adaptability of our framework for different routing scenarios. In (a), multiple equivalent topology candidates are available, and the framework selects the one with lower routing cost. In (b), when congestion arises on a specific tier, the optimizer reroutes through alternative paths on less congested tiers. These examples demonstrate the framework's flexibility in resolving redundancy and mitigating congestion.

D. Probabilistic Modeling over 3D Routing Candidates

To enable differentiable optimization across the full 3D routing space, each candidate in the three-level hierarchy—topology, path, and tier assignment—is assigned a trainable variable. A softmax operation is applied within each mutually exclusive candidate set to produce a probability distribution over selections.

Let $W_q = \{w_{q_1}, w_{q_2}, \ldots, w_{q_m}\}$ denote the trainable weights assigned to the m topology candidates for a given net. The corresponding softmax selection probabilities are computed as:

$$q_i = \frac{\exp(w_{q_i})}{\sum_{j=1}^{m} \exp(w_{q_j})}, \quad \text{for } i = 1, \ldots, m \tag{4}$$

Similarly, for a given 2-pin subnet, let $W_p = \{w_{p_1}, w_{p_2}, \ldots\}$ and $W_r = \{w_{r_1}, w_{r_2}, \ldots\}$ represent the sets of trainable weights associated with path and tier assignment candidates, respectively. The corresponding selection probabilities are computed as:

$$p_j = \frac{\exp(w_{p_j})}{\sum_k \exp(w_{p_k})}, \qquad r_l = \frac{\exp(w_{r_l})}{\sum_h \exp(w_{r_h})} \tag{5}$$

Each final routing candidate corresponds to a unique combination of one topology, one path, and one tier assignment. The probability of selecting the i-th candidate is given by:

$$P_i = q_{x(i)} \cdot p_{y(i)} \cdot r_{z(i)} \tag{6}$$

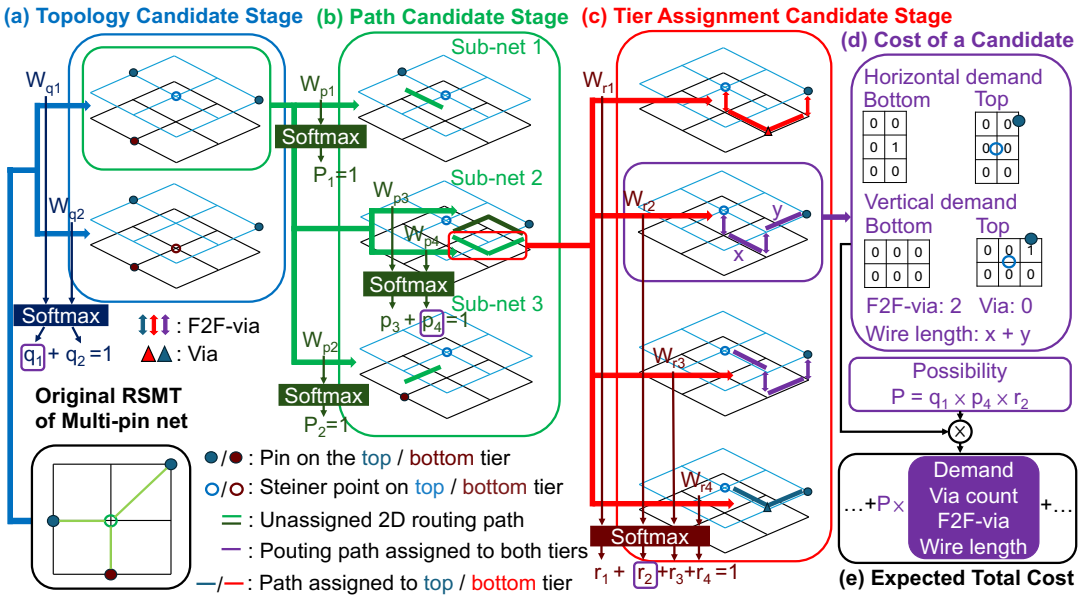

Fig. 5: Candidate-rich 3D pattern routing framework. Each net explores multiple topology, path, and tier assignment candidates. Expected routing costs are computed via probability-weighted summation. The process can be divided into three stages: (a) the topology candidate stage, also referred to as the q-selection stage; (b) the path candidate stage, or the p-selection stage; and (c) the tier assignment candidate stage, or the r-selection stage. Subfigures (d) and (e) illustrate how the routing cost is computed.

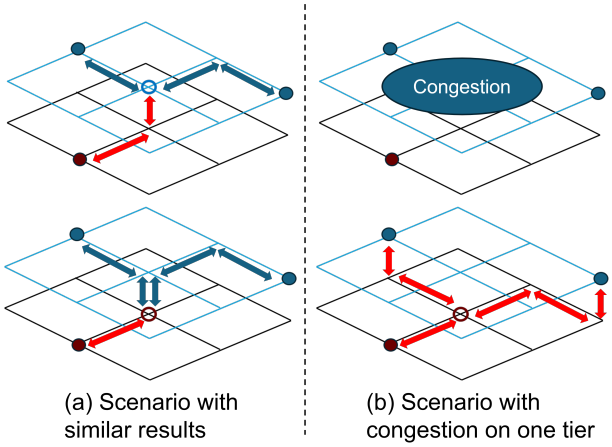

Fig. 6: Figure 6: Examples showing our framework's ability to resolve redundancy and detour around congestion.

where $q_{x(i)}$, $p_{y(i)}$, and $r_{z(i)}$ denote the softmax probabilities of the selected topology, path, and tier assignment components associated with candidate i, respectively. Each final routing candidate i contributes a deterministic routing demand matrix D_i, determined by its associated 2-pin sub-net, path geometry, and tier assignment. We define two separate 3D demand matrices—one for horizontal tracks and one for vertical tracks—each indexed by (x, y, t), where t denotes the tier (e.g., top or bottom die), as illustrated in Figure 5(d).

The expected demand matrix $E(D)$ is then computed as the weighted sum of all candidate demands:

$$E(D) = \sum_i P_i \cdot D_i \tag{7}$$

Given a capacity matrix C that encodes the routing capacity available per grid location (x, y, t), the element-wise overflow matrix is calculated as follows:

$$O = ReLU(E(D) - C) = \max(0, E(D) - C) \tag{8}$$

Finally, the total expected congestion overflow cost is obtained by summing over all entries in the overflow matrix:

$$\text{OverflowCost} = \sum_{x,y,t} O(x, y, t) \tag{9}$$

This differentiable formulation enables global coordination of routing decisions across all nets, while jointly considering topology, path, and tier constraints specific to 3D ICs.

IV. Differentiable Optimization

A. Differentiable Loss Function

In the previous section, we formulated the expected congestion overflow cost by computing a probability-weighted sum over all routing candidates. Using the same probabilistic framework, we compute the expected values for other routing-related objectives, enabling a unified differentiable loss formulation.

Together with the total expected congestion overflow cost OverflowCost previously defined, we construct our differentiable loss function as a weighted sum of the four objectives:

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \lambda_o \cdot \text{OverflowCost} + \lambda_f \cdot \text{F2FViaCountCost} \\ &\quad + \lambda_w \cdot \text{WireLengthCost} + \lambda_v \cdot \text{ViaCountCost} \\ &= \lambda_o \cdot \text{OverflowCost} + \sum_i P_i \cdot (\lambda_f F_i + \lambda_w W_i + \lambda_v V_i)\end{aligned} \tag{10}$$

In this formulation, P_i denotes the soft selection probability assigned to routing candidate i, as defined in the previous sections. Each candidate is associated with three cost metrics: wirelength W_i, via count V_i, and F2F via count F_i. The total loss $\mathcal{L}_{\text{total}}$ is expressed as a weighted sum of four objectives: expected overflow cost, wirelength, via count, and F2F via count. The weights λ_o, λ_f, λ_w, and λ_v represent the relative importance of each objective. In our setting, overflow minimization is prioritized as the dominant goal ($\lambda_o = 500$), followed by F2F via count due to its high manufacturing cost in 3D

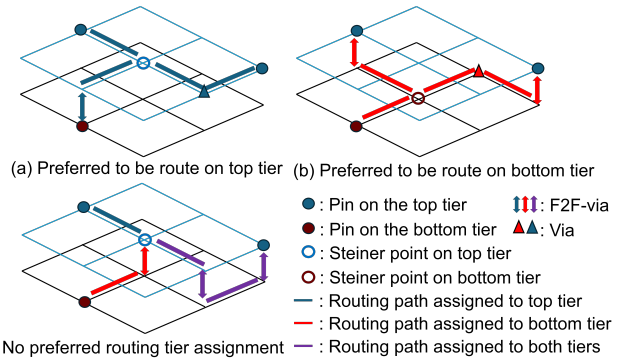

Fig. 7: Illustration of tier exclusivity and tier sharing in net-wise routing. (a)–(b) Nets fully routed within a single tier are assigned to that corresponding die. (c) A net spanning both tiers is considered tier-sharing and left unassigned.

integration ($\lambda_f = 50$). Wirelength and via count are weighted equally ($\lambda_w = \lambda_v = 1$) to promote routing efficiency without overwhelming the optimization. This differentiable formulation enables end-to-end gradient-based optimization over the 3D routing space and design objectives.

V. Routing Tier Assignment Guidance

A. Candidate Selection and Discretization

After the optimization converges, we extract a discrete routing solution from the final soft probability distributions (Figure 2(g)). For each multi-pin net, we select the topology candidate with the highest softmax probability q_i as the final topology choice. Then, for each two-pin sub-net belonging to that topology, we select the path candidate with the maximum p_j and its corresponding tier assignment candidate with the maximum r_k, as shown in Figure 5 purple boxes. By performing this selection for every net in the design, we obtain a complete 3D global routing solution consisting of one deterministic topology, path, and tier assignment per sub-net. This final decision reflects the outcome of the differentiable optimization process and can be used to guide subsequent physical design stages.

B. Routing Tier Assigner

Following the candidate selection and discretization stage, we obtain a complete 3D global routing solution in which each multi-pin net has a concrete topology, path, and tier assignment for all its two-pin sub-nets. This rich solution space contains embedded tier-wise information that can be further leveraged to enhance commercial router routing result. To extract tier assignment information at the net level, we begin by aggregating the tier assignments originally assigned to each two-pin sub-net. Specifically, for each net, we collect the tier information associated with its routed trunks from all sub-nets and combine them into a net-wise routing tier set. Our proposed *Routing Tier Assigner* module analyzes the net-wise set of routing trunks to determine whether a net exhibits tier locality (Figure 2(h)). If all routing trunks of a net are assigned to the same tier—i.e., routed entirely within a single die—the net is said to exhibit tier exclusivity and is directly assigned to that tier. For instance, Figure 7(a) shows a case where the net is fully routed in the top tier, leading to an assignment to the top die. Similarly, Figure 7(b) depicts a net routed exclusively in the bottom tier.

However, if the routing trunks of a net span multiple tiers—indicating that its paths traverse more than one die—the net is considered non-exclusive in tier assignment. In this case, it is not assigned to any specific tier, as illustrated in Figure 7(c). Note that the Routing Tier Assigner considers only the routing trunk tier assignments, and does not account for the tiers where pins or Steiner points are located.

C. Preferred Routing Tier Guide

All nets that are assigned to a specific tier by the Routing Tier Assigner are compiled into a structured list that we refer to as the *Preferred Routing Tier Guide* (Figure 2(i)). This guide can be fed back into commercial tools to influence subsequent routing decisions. For each net in the guide, we specify the *bottom_preferred_routing_layer* and *top_preferred_routing_layer* to align with the metal layer range associated with its assigned tier. This encourages the router to route the majority of that net's wire segments within the designated tier. Although many 3D nets naturally span multiple tiers due to their pin locations, which are distributed across dies, it is not possible for the entire net to be confined to a single tier. However, by guiding the router to prefer the routing layers associated with the assigned tier, our Preferred Routing Tier Guide effectively ensures that most of the net's wirelength remains within the intended tier.

VI. Experimental Results

A. Experimental Setting

We evaluated our framework on three industrial F2F 3D IC designs—LDPC, DMA, and AES—using TSMC's 28 nm Process Design Kit (PDK). Each design comprises a two-tier stack, with six metal layers per tier forming a 3D back-end-of-line (BEOL) structure, interconnected via F2F hybrid bonding at a 1 μm pitch [46]. To reflect realistic congestion conditions and stress the global router, we introduced controlled routing blockages across selected layers. These emulate practical constraints such as macro placement, IP hardening, and reserved routing regions, which enables a more meaningful evaluation of our framework's ability to manage congestion and improve timing closure. Blockages were set to 75% on M5–M6 for LDPC, 12% on M3–M6 for DMA, and 12% on M1–M6 for AES.

B. Naive 2D Tier Assignment is Insufficient

To assess the effectiveness of simple tier confinement, we conduct a baseline experiment (w/ naive 2D PRTG) by applying tier-specific Routing Guides (PRTGs) to all 2D nets without any optimization. In this setup, each 2D net is restricted to route only within the tier containing all its pins, effectively disabling metal layer sharing. As shown in Table I, this naive PRTG strategy significantly degrades routing quality, increasing early global routing overflow by 48.37% and slightly worsening total negative slack (TNS). Additionally, directly applying all extracted PRTG information leads to suboptimal outcomes. This is likely because many low fan-out 2D nets—which do not benefit from tier-level guidance—are unnecessarily constrained, thereby reducing routing flexibility and increasing congestion. These results clearly demonstrate that effective tier assignment in 3D ICs cannot be achieved through simple rules or static confinement. The complex interactions between nets and the spatially localized nature of congestion require more adaptive, net-specific strategies that consider both topology and routing context.

C. Targeting High-Impact Nets for Routing Guidance

Rather than applying routing guidance (PRTG) indiscriminately to all nets, we adopt a selective strategy that targets high fan-out nets, which are more likely to contribute to congestion and timing violations. In contrast, low fan-out nets-particularly those with fan-out of 2 or 3—tend to have simpler routing requirements and limited impact on timing and global congestion. Over-constraining these nets may reduce routing flexibility and inadvertently worsen overall routing quality. Table I summarizes the impact of different fan-out thresholds on design outcomes. Experimental results show that applying PRTG to only the top 3% of high fan-out nets provides an effective trade-off between targeted guidance and overall routing flexibility. This selective strategy focuses on the nets most likely to affect congestion

TABLE I: Comparison of routing and timing metrics for LDPC under various Routing Guide (PRTG) strategies. Selectively applying PRTG to the top 3% high fan-out nets provides the best overall improvements in congestion and timing without increasing wirelength and F2F-via count.

	# overflow	TNS (*ps*)	# timing violation	Wire length (*m*)	# F2F-via
LDPC (2.5*GHz*) (# nets: 56527)					
Pin-3D [41]	920	-7.683	1003	1.145	14594
w/ naive 2D PRTG	1365 (+48.37%)	-7.738 (+0.72%)	999 (-0.40%)	1.149	14661
w/ PRTG entire	1626(+76.74%)	-9.395(+22.28%)	1131(+12.76%)	1.148	14801
w/ PRTG fan-out $\geq$ 3 (Top 7.6%)	895 (-2.72%)	-7.629 (-0.70%)	1038 (+3.49%)	1.145	14697
w/ PRTG Top 5% high fan-out net	852 (-7.39%)	-7.624 (-0.77%)	1014 (+1.10%)	1.145	14735
w/ PRTG Top 3% high fan-out net	843 (-8.37%)	-5.84 (-23.99%)	822 (-18.05%)	1.144	14708
w/ PRTG Top 1% high fan-out net	906 (-1.52%)	-5.877 (-23.51%)	830 (-17.25%)	1.144	14718
w/ PRTG fan-out $\geq$ 10 (Top 0.18%)	914 (-0.65%)	-7.361 (-4.19%)	955 (-4.79%)	1.145	14731

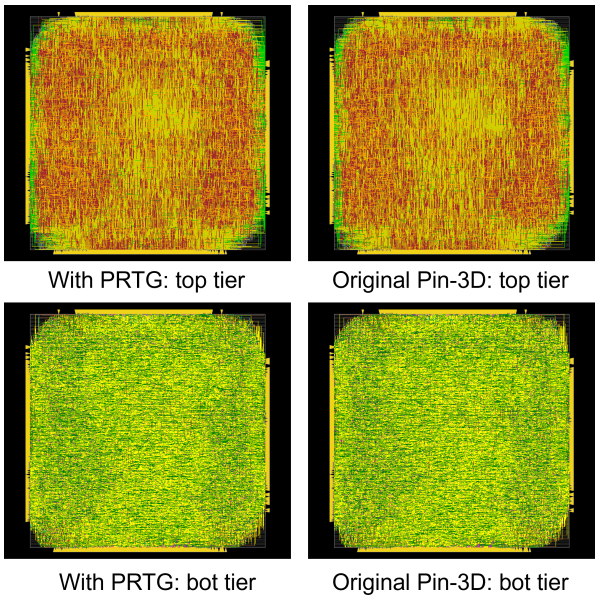

Fig. 8: Post-route layouts of LDPC with our Routing Guide (PRTG) vs. original Pin-3D [41] flow. Our method achieves up to 8.37% reduction in early global routing overflow, 23.99% reduction in post-route TNS, and 18.05% reduction in post-route number of timing violations.

and timing, while avoiding unnecessary constraints on lower-impact nets. Compared to the Pin-3D baseline, this configuration achieves a reduction of **8.37%** in global routing overflow, **23.99%** in post-route total negative slack (TNS), and **18.05%** in post-route timing violations, without increasing wirelength or F2F-via count. These results demonstrate the effectiveness of our high-impact targeted strategy for routing guidance in improving overall design quality.

D. Generalization Across Multiple Designs

To assess the generality of our PRTG strategy, we evaluate it on two additional designs, AES and DMA. As shown in Table II, PRTG consistently improves routing and timing quality across benchmarks. For instance, in the DMA design, PRTG reduces post-route total negative slack (TNS) by **14.37%** and timing violations by **17.05%**, with minimal impact on wirelength and F2F-via count, which demonstrate the effectiveness of net-wise tier assignment across diverse designs.

E. PRTG Compliance and Routing Behavior Analysis

We further analyze whether our PRTG constraints are respected by the commercial router. For this, we compute the percentage of each constrained net's post-route wirelength that resides within the specified preferred tier. Among constrained 3D nets, we observe a substantial increase in compliance rate: For 3D net assign to bottom tier, the percentage of routing within the assigned tier increased from 76.45% (baseline) to 98.43% with our PRTG; For 3D net assign to top tier, the same metric improved from 83.19% to 99.04%, as shown in Table III. These results confirm that our PRTG strategy is highly effective in steering net routing into desired tiers, validating its utility as a practical guidance mechanism for tier-aware metal layer sharing.

TABLE II: Comparison of routing and timing metrics before and after applying PRTG across designs. Results show that PRTG improves congestion and timing while maintaining wirelength and via usage.

Metrics	**Pin-3D [41]**	**w/ PRTG**	**improv**
LDPC (2.5*GHz*) (# nets: 56527) Top 3% high fan-out nets			
# overflow	920	843	-8.37%
TNS (*ps*)	-7.683	-5.84	-23.99%
# timing violation	1003	822	-18.05%
Wire length (*m*)	1.145	1.144	-0.12%
# F2F-via	14594	14708	0.78%
DMA (2.5*GHz*) (# nets: 11082) Top 3% high fan-out nets			
# overflow	28058	28914	3.05%
TNS (*ps*)	-24.177	-20.703	-14.37%
# timing violation	1056	876	-17.05%
Wire length (*m*)	0.187	0.187	-0.13%
# F2F-via	4847	4859	0.25%
AES (4.5*GHz*) (# nets: 129454) Top 1% high fan-out nets			
# overflow	309068	309115	0.02%
TNS (*ps*)	-36.551	-34.721	-5.01%
# timing violation	2050	2023	-1.32%
Wire length (*m*)	1.260	1.260	0.03%
# F2F-via	43580	43728	0.34%

TABLE III: Post-route wirelength compliance with assigned tiers under Pin-3D baseline and after applying Preferred Routing Tier Guide (PRTG). Significant improvements are observed for 3D nets.

Wire length distribution	**Pin-3D [41]**	**w/ PRTG**	
2D net in top tier	99.07%	99.53%	+0.46%
2D net in bottom tier	99.76%	99.90%	+0.14%
3D net assign to top	83.19%	99.04%	+15.85%
3D net assign to bottom	76.45%	98.43%	+21.98%

VII. Conclusion

In this work, we proposed a differentiable routing guidance framework for 3D ICs that generates tier-specific guidance compatible with commercial design flows. By constructing a candidate-rich 3D routing DAG forest and formulating a differentiable loss over multiple objectives: congestion overflow, wirelength, via count, and face-to-face (F2F) via usage, our method enables concurrent optimization across the full 3D routing space. Experiments on multiple F2F 3D IC benchmarks demonstrate improvements in both congestion and timing. DRG-3D reduces overflow by up to 8.37%, improves total negative slack (TNS) by 23.99%, and reduces timing violations by 18.05%, with minimal wirelength and via overhead. In future work, we plan to extend DRG-3D to larger industrial scale 3D designs and explore its integration with parameter optimization, graph neural network or reinforcement Learning framework [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63] to further improve design quality.

REFERENCES

[1] Y. Xu *et al.*, "Fastroute 4.0: Global router with efficient via minimization," in *2009 ASP-DAC*, IEEE, 2009.

[2] W.-H. Liu *et al.*, "Nctu-gr 2.0: Multithreaded collision-aware global routing with bounded-length maze routing," *IEEE TCAD*, 2013.

[3] Y.-J. Chang *et al.*, "Nthu-route 2.0: a robust global router for modern designs," *IEEE TCAD*, 2010.

[4] J. Liu *et al.*, "Edge: Efficient dag-based global routing engine," in *2023 60th ACM/IEEE DAC*, IEEE, 2023.

[5] J. Hu *et al.*, "Sidewinder: a scalable ilp-based router," in *Proceedings of the SLIP*, 2008.

[6] M. Cho *et al.*, "Boxrouter 2.0: A hybrid and robust global router with layer assignment for routability," *ACM TODAES*, 2009.

[7] M. Pan *et al.*, "Fastroute: A step to integrate global routing into placement," in *Proceedings of the 2006 IEEE/ACM ICCAD*, 2006.

[8] Y. Xu *et al.*, "Mgr: Multi-level global router," in *2011 IEEE/ACM ICCAD*, IEEE, 2011.

[9] D. Wu *et al.*, "Layer assignment for crosstalk risk minimization," in *ASP-DAC 2004*, IEEE, 2004.

[10] G. Xu *et al.*, "Redundant-via enhanced maze routing for yield improvement," in *Proceedings of the 2005 ASP-DAC*, 2005.

[11] K.-Y. Lee *et al.*, "Post-routing redundant via insertion for yield/reliability improvement," in *Proceedings of the 2006 ASP-DAC*, 2006.

[12] T.-R. Lin *et al.*, "qgdr: A via-minimization-oriented routing tool for large-scale superconductive single-flux-quantum circuits," *IEEE Transactions on Applied Superconductivity*, 2019.

[13] T.-H. Lee and T.-C. Wang, "Congestion-constrained layer assignment for via minimization in global routing," *IEEE TCAD*, 2008.

[14] T.-H. Lee and T.-C. Wang, "Simultaneous antenna avoidance and via optimization in layer assignment of multi-layer global routing," in *2010 IEEE/ACM ICCAD*, IEEE, 2010.

[15] J. He, U. Agarwal, Y. Yang, R. Manohar, and K. Pingali, "Sproute 2.0: A detailed-routability-driven deterministic parallel global router with soft capacity," in *2022 27th ASP-DAC*, IEEE, 2022.

[16] A. Youssef *et al.*, "A power-efficient multipin ilp-based routing technique," *IEEE T Circuits-I*, 2009.

[17] C.-J. Chang *et al.*, "Ilp-based inter-die routing for 3d ics," in *16th ASP-DAC 2011*, IEEE, 2011.

[18] H. Kong *et al.*, "Automatic bus planner for dense pcbs," in *Proceedings of the 46th Annual DAC*, 2009.

[19] T.-H. Wu *et al.*, "Grip: Scalable 3d global routing using integer programming," in *Proceedings of the 46th Annual DAC*, 2009.

[20] W. Li *et al.*, "Dgr: Differentiable global router," in *Proceedings of the 61st ACM/IEEE DAC*, pp. 1–6, 2024.

[21] Y. Lin *et al.*, "Dreamplace: Deep learning toolkit-enabled gpu acceleration for modern vlsi placement," in *ACM/IEEE DAC*, 2019.

[22] P. Liao *et al.*, "Dreamplace 4.0: Timing-driven global placement with momentum-based net weighting," in *2022 DATE*, IEEE, 2022.

[23] Y.-C. Lu *et al.*, "Insta: An ultra-fast, differentiable, statistical static timing analysis engine for industrial physical design applications," in *62th ACM/IEEE Design Automation Conference (DAC)*, 2025.

[24] H.-H. Hsiao, Y.-C. Lu, P. Vanna-Iampikul, A. Agnesina, R. Liang, Y.-H. Lu, H. Ren, and S. K. Lim, "DCO-3D: Differentiable Congestion Optimization in 3D ICs," in *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pp. 1–7, IEEE, 2025.

[25] Y.-C. Lu *et al.*, "GAN-Place: Advancing Open Source Placers to Commercial-quality Using Generative Adversarial Networks and Transfer Learning," *ACM Transactions on Design Automation of Electronic Systems*, 2024.

[26] Y.-C. Lu *et al.*, "Dream-gan: Advancing dreamplace towards commercial-quality using generative adversarial learning," in *Proceedings of the 2023 International Symposium on Physical Design*, 2023.

[27] Y.-C. Lu *et al.*, "C3po: Commercial-quality global placement via coherent, concurrent timing, routability, and wirelength optimization," in *Proceedings of the 31st Asia and South Pacific Design Automation Conference*, 2026.

[28] S. Lin *et al.*, "Superfast full-scale cpu-accelerated global routing," in *Proceedings of the 41st IEEE/ACM ICCAD*, pp. 1–8, 2022.

[29] S. Lin *et al.*, "Gamer: Gpu-accelerated maze routing," *IEEE TCAD*.

[30] Y.-C. Lu, H.-H. Hsiao, and H. Ren, "LLM-Enhanced GPU-Optimized Physical Design at Scale,"

[31] S. Lin *et al.*, "Instantgr: Scalable gpu parallelization for global routing," in *Proceedings of the 43rd IEEE/ACM ICCAD*, 2024.

[32] R. Liang *et al.*, "Gpu/ml-enhanced large scale global routing contest," in *Proceedings of the 2024 ISPD*, 2024.

[33] Y.-H. Chung *et al.*, "Simpart: A simple yet effective replication-aided partitioning algorithm for logic simulation on gpu," in *European Conference on Parallel Processing*, pp. 197–210, Springer, 2025.

[34] Y.-H. Chung *et al.*, "Accelerating gate sizing using gpu," in *European Conference on Parallel Processing*, 2025.

[35] B. Zhang *et al.*, "iTAP: An Incremental Task Graph Partitioner for Task-parallel Static Timing Analysis," in *Proceedings of the 30th ASP-DAC*, 2025.

[36] C. Chang *et al.*, "PathGen: An Efficient Parallel Critical Path Generation Algorithm," in *Proceedings of the 30th ASP-DAC*, 2025.

[37] Y.-R. Chen *et al.*, "Rl-routing: An sdn routing algorithm based on deep reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, 2020.

[38] H. Liao *et al.*, "A deep reinforcement learning approach for global routing," *Journal of Mechanical Design*, 2020.

[39] L. Yang *et al.*, "Towards timing-driven routing: An efficient learning based geometric approach," in *IEEE/ACM ICCAD*, IEEE, 2023.

[40] Q. Wang *et al.*, "A multi-agent generative model for collaborative global routing refinement," in *Proceedings of the GLS-VLSI 2024*, 2024.

[41] S. S. K. Pentapati *et al.*, "Pin-3d: A physical synthesis and post-layout optimization flow for heterogeneous monolithic 3d ics," in *Proceedings of the 39th ICCAD*, pp. 1–9, 2020.

[42] A. B. Kahng *et al.*, "NN-Steiner: A mixed neural-algorithmic approach for the rectilinear Steiner minimum tree problem," in *AAAI*, 2024.

[43] G. Chen and E. F. Young, "Salt: provably good routing topology by a novel steiner shallow-light tree algorithm," *IEEE TCAD*, 2019.

[44] C. Chu *et al.*, "Flute: Fast lookup table based rectilinear steiner minimal tree algorithm for vlsi design," *IEEE TCAD*, 2007.

[45] J. Liu *et al.*, "Cugr: Detailed-routability-driven 3d global routing with probabilistic resource model," in *57th ACM/IEEE DAC*, IEEE, 2020.

[46] S. E. Kim *et al.*, "Wafer level cu–cu direct bonding for 3d integration," *Microelectronic Engineering*, 2015.

[47] H.-H. Hsiao *et al.*, "FastTuner: Transferable Physical Design Parameter Optimization using Fast Reinforcement Learning," in *Proceedings of the 2024 International Symposium on Physical Design*, pp. 93–101, 2024.

[48] H.-H. Hsiao *et al.*, "Insightalign: A transferable physical design recipe recommender based on design insights," in *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pp. 1–7, IEEE, 2025.

[49] H.-H. Hsiao *et al.*, "Ml-based physical design parameter optimization for 3d ics: From parameter selection to optimization," in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pp. 1–6, 2024.

[50] H.-H. Hsiao, Y.-C. Lu, S. K. Lim, and H. Ren, "BUFFALO: PPA-Configurable, LLM-based Buffer Tree Generation via Group Relative Policy Optimization," in *Proceedings of the 44th IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2025.

[51] H.-H. Hsiao *et al.*, "A Hybrid Reinforcement Learning Framework for Efficient Physical Design Parameter Tuning," in *ACM Transactions on Design Automation of Electronic Systems*, 2025.

[52] J. Pan *et al.*, "Crop: Circuit retrieval and optimization with parameter guidance using llms," 2025.

[53] R. Zhang *et al.*, "Automated physical design watermarking leveraging graph neural networks," in *ACM/IEEE MLCAD*, 2024.

[54] R. Zhang *et al.*, "Automarks: A gnn-based automated physical design watermarking framework," *ACM TODAES*, 2025.

[55] J. Pan, *Intelligent Electronic Design Automation Through Machine Learning Methods*. PhD thesis, Duke University, 2025.

[56] Y.-C. Lu *et al.*, "Rl-sizer: Vlsi gate sizing for timing optimization using deep reinforcement learning," in *2021 58th ACM/IEEE DAC*, 2021.

[57] Y.-C. Lu *et al.*, "Rl-ccd: Concurrent clock and data optimization using attention-based self-supervised reinforcement learning," in *60th ACM/IEEE DAC*, 2023.

[58] Y.-C. Lu *et al.*, "Doomed run prediction in physical design by exploiting sequential flow and graph learning," in *IEEE/ACM ICCAD*, IEEE, 2021.

[59] Y.-C. Lu *et al.*, "A fast learningdriven signoff power optimization framework. In 2020 IEEE," in *ACM ICCAD*, 2020.

[60] Y.-C. Lu *et al.*, "Vlsi placement optimization using graph neural networks," in *34th NeurIPS Workshop on ML for Systems*, 2020.

[61] Z. Yang *et al.*, "Ml-based fine-grained modeling of dc current crowding in power delivery tsvs for face-to-face 3d ics," in *Proceedings of the ISPD*, 2025.

[62] Z. Yang *et al.*, "Graph attention-based current crowding analysis at tsv interfaces in 3d power delivery networks," in *Proceedings of the 31st ASP-DAC*, 2026.

[63] J. Hu *et al.*, "Gnn-mls: Signal routing in mixed-node 3d ics through gnn-assisted metal layer sharing," in *62nd ACM/IEEE DAC*, IEEE, 2025.