3D Chiplet Partitioning and Floorplanning Interaction with Vertical Bonding Consideration

Tong Shen[1], Mengen Chen[1], Xu He[1†], Yao Wang[2], Yang Guo[3]

[1]Hunan University, Changsha, China
[2]Independent Researcher, Changsha, China
[3]National University of Defense Technology, Changsha, China

***Abstract*—The emerging technologies of 3D chiplet integration offer a promising path to increase functional density and communication efficiency beyond the limitations of traditional 2D designs. Among various methods, hybrid bonding enables high-density vertical connections with reduced parasitics and latency. However, few existing approaches explicitly support this vertical interconnect scenario, and most treat partitioning and floorplanning as separate stages. To fully exploit the benefits of vertical integration, partitioning and floorplanning need to be considered jointly to balance bond demand and bond supply. In this paper, we propose a unified framework for 3D chiplet partitioning and floorplanning under fine-pitch bonding technologies. Experiments on industry-standard benchmarks demonstrate a 10%∼40% reduction in HPWL and over 60% decrease in inter-die vertical connection overflow, confirming the effectiveness and superiority of our approach over current methods.**

***Index Terms*—Chiplet, Partitioning, Functional Module Floorplanning**

I. Introduction

The chiplet integration is a key "More-than-Moore" approach that stacks dies or chiplets to achieve higher functional integration in a limited space. This technology flexibly consolidates homogeneous or heterogeneous chips like logic, memory, analog and RF, boosting performance and communication efficiency for applications like in-memory computing [1]. Traditional chiplet technologies with silicon/RDL interposers mainly enable planar interconnects only. TSVs in passive interposers provide only vertical conduction without active circuitry, essentially function as horizontal links and fail to meet the integration and communication demands of hundred-billion-transistor systems. These limitations have driven the emergence of 3D chiplet integration, which delivers true vertical connectivity to shorten interconnect paths, reduce latency, and increase bandwidth, ultimately enabling higher logic density and greater communication energy efficiency [2].

Due to the size and pitch limitations of current TSV+uBump technologies, hybrid bonding has thus emerged as a critical method for realizing high-density vertical connections in 3D integration [1]. Unlike traditional micro-bump bonding, its direct copper-to-copper and dielectric-to-dielectric contact eliminates the intermediate solder joints, reducing interconnect size and pitch from tens of microns to the sub-10 µm level [3]. This also lowers parasitic resistance and capacitance, thereby improving signal delay and energy efficiency. The AMD MI300A APU [4] is a prime example, featuring multiple top-die chiplets vertically stacked on an active interposer and integrated with base-die, showcasing the full potential of 3D chiplet architectures.

†Xu He is the corresponding author, email: dawn.hx@gmail.com. This work is supported by the National Natural Science Foundation of China, under grant 92473115.

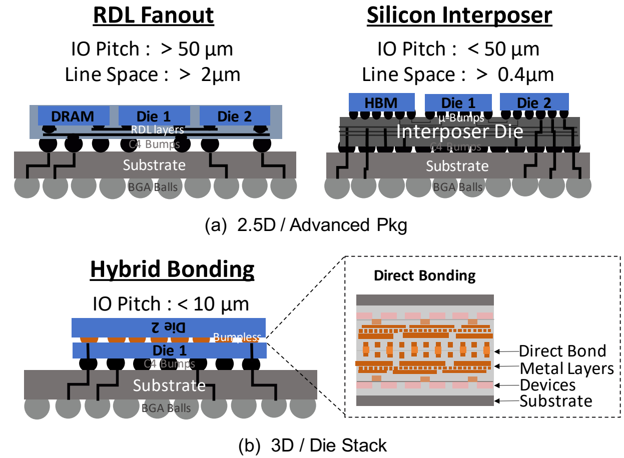

Fig. 1. Evolution of chiplet integration technologies: from 2.5D interposer-based interconnects to 3D die stacking with hybrid bonding, enabling sub-10 µm pitch and high-density vertical integration.

However, hybrid bonding demands extremely high alignment precision, requiring sufficient vertical overlap for 3D bonding between chiplets across different dies. Nevertheless, most existing approaches overlook explicit optimization of overlap and treat partitioning and floorplanning as separate stages. As illustrated in Fig. 1, the number of inter-die nets defines the demand for vertical interconnects, while the overlap area between chiplets represents the supply constrained by the density limits of hybrid bonding technologies. Overflow arises when the inter-die demand exceeds the available overlap supply. Partitioning determines demand, whereas floorplanning governs supply. Ideally, supply should exceed demand, which can be achieved either by reducing inter-chiplet connections through partitioning or by enlarging overlap regions through floorplanning. Once the overlap reaches a local optimum under a fixed partition, further improvement becomes infeasible. This tight coupling between demand and supply highlights the necessity of a unified framework that co-optimizes partitioning and floorplanning.

Despite this need, most prior works treat partitioning and location computation in isolation, only approximating final quality and thus resulting in degradation. Recent advances in chiplet design include Chipletizer [5], which offers a framework for cost-aware SoC partitioning into reusable chiplets.

978-3-9826741-1-7/DATE26/© 2026 EDAA

Floorplet [6], a performance-aware floorplanning framework that optimizes interconnect delay and overall system performance. ChipletPart [7], which proposes cost-driven partitioning for 2.5D systems. However, these studies remain limited to 2D/2.5D chiplet design, making them difficult to extend directly to 3D chiplet scenarios. In parallel, many 3D IC works have attempted co-optimization of partitioning and placement by stacking or folding 2D layouts [8], applying FM-based die partitioning [9], extending 2D commercial tools [10], or modeling electrostatic fields for direct 3D placement [11]. Nevertheless, since chiplet partitioning is inherently performed at the module level, existing cell-level placements are not directly applicable. This gap highlights the need for new algorithms tailored to module-level constraints in 3D chiplet integration.

Unlike 3D partitioning in monolithic chips, 3D chiplet design introduces greater complexity. Beyond assigning modules and chiplets to top-die and base-die, a single chiplet may integrate multiple functional modules, making partitioning subject to both interconnect and area constraints. Consequently, traditional 3D IC partitioning methods are often inadequate for large-scale 3D chiplet systems. To address this challenge, we formulate a unified problem that jointly solves chiplet partitioning and floorplanning. Our approach explicitly considers the unique characteristics of chiplet integration, including partitioning area constraints, chiplet upper/lower utilization limits, and inter-die vertical connection resource constraints. We further introduce an iterative process in which repartitioning and floorplanning are performed simultaneously during the chiplet floorplanning stage. This process ensures that all constraints are satisfied while optimizing for shorter overall wirelength and reducing overflow of inter-die connections that cannot be directly realized through vertical bonding. Our main contributions are summarized as follows:

- Present a solution framework that transforms 3D chiplet vertical bonding requirements into inter-die chiplet partitioning and location overlap optimization.
- Introduce a vertical overlap-aware floorplanning method that balances inter-die connection demand and supply, while iteratively co-optimizing with chiplet repartitioning.
- Propose a theoretically justified, overlap-aware allocation strategy that addresses extreme imbalances between vertical bonding supply and demand.
- Experiments on industry benchmarks demonstrate that our method outperforms state-of-the-art chiplet design and partitioning approaches, achieving 10%∼40% reduction in HPWL and over 60% decrease in inter-die vertical connection overflow.

The remainder of this paper is organized as follows. Section II presents the problem formulation. Section III demonstrates our framework. The details of 3D chiplet partitioning and floorplanning are given in Section IV. Experimental results are presented and discussed in Section V, followed by the conclusions in Section VI.

II. Problem Formulation

Under face-to-face 3D chiplet structure, given a circuit netlist, we model it as a weighted undirected graph $G = (V, E, c, w)$, where the vertex set V represents modules and the edge set E represents interconnections between them. Each vertex $v \in V$ is assigned a weight $c(v)$ corresponding to its area, and each edge $e \in E$ has a weight $w(e)$ representing the number of nets connecting the modules.

For a given total chiplet count $k = k_{top} + k_{base}$ with the k_{top} chiplets in the top-die and the remaining k_{base} corresponding to chiplets in the base-die. A k-way partitioning solution is defined as $P = \{p_1, p_2, \ldots, p_k\}$, where P forms a disjoint partition of the vertex set V. Each subset p_i contains the modules assigned to the i-th chiplet. The total module area assigned to chiplet i is given by $c(p_i) = \sum_{v \in p_i} c(v)$.

The top-die and base-die are typically aligned vertically. The parameter $\rho_{\max}$ sets the maximum allowable vertical connection density, limiting the number of inter-die connections between the top and base dies. Ideally, the projected overlap between interacting chiplets across the dies ensures that the local inter-die connection density remains below $\rho_{\max}$. In practice, insufficient overlap between connected chiplet pairs may result in a vertical connection supply that cannot satisfy the inter-die connection demand. We define $Overflow$ as the difference between demand and supply, and $|Demand_{Die}|$ denotes the total vertical connection demand of inter-die nets.

Our problem is defined as finding a partition P of a given circuit $G = G(V, E, c, w)$, with a floorplan that specifies all the chiplets within their respective dies such that a weighted objective function is minimized, subject to a series of design constraints. The optimization problem can be formulated as:

$$\begin{aligned} \min \quad & HPWL + \lambda \cdot Overflow \\ \text{s.t.} \quad & \text{constraints (1)–(4) hold.} \end{aligned} \tag{1}$$

Here, half-perimeter wirelength (HPWL) is used to measure the communication cost, while $Overflow$ evaluates the vertical inter-die connection overflow for 3D bonding with factor λ. We will optimize local inter-die density as a surrogate for global $Overflow$, since they are well correlated in practice and both serve the overall goal of accommodating inter-die nets.

Constraints. To ensure feasibility in 3D chiplet packaging, the following constraints, which are commonly required in industry, must be satisfied.

1) **Module Assignment**: Each module must be assigned to one and only one chiplet on a single die.
2) **Chiplet Area**: Chiplets within each die must be non-overlapping and placed entirely within the core area. Each individual chiplet area A_i must be greater than a predefined minimum area A_{min}, while the sum of all chiplet areas cannot exceed the die area A_{total}.

$$A_i \geq A_{min} \quad \text{and} \quad \sum_{i=1}^{k} A_i \leq A_{total} \tag{2}$$

3) **Chiplet Utilization**: The area utilization for each individual chiplet must fall within a specified range. This is often represented as:

$$util_{min}^{i} \leq \frac{c(p_i)}{A_i} \leq util_{max}^{i} \tag{3}$$

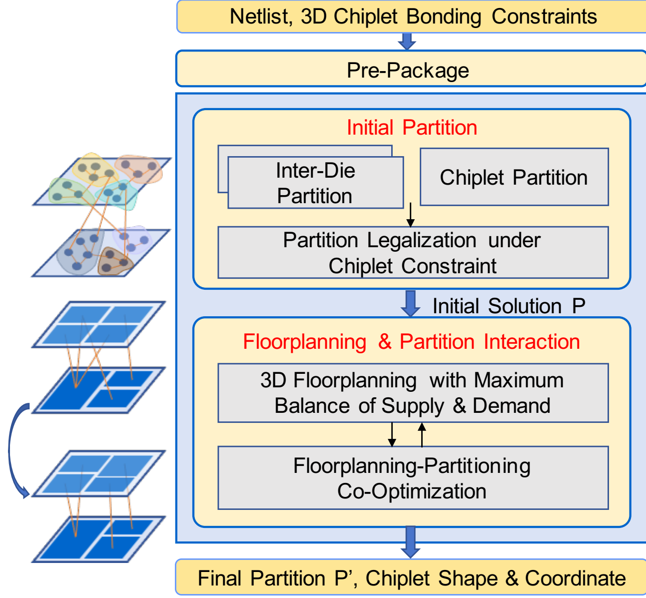

Fig. 2. Framework overview of our method.

For each die, the number of chiplets k and the utilization bounds $(util_{min}, util_{max})$ are specified independently.

4) **Vertical Interconnect Density**: The overall interconnect density between dies must not exceed ρ_{max}. This constraint is crucial for ensuring the physical feasibility of the design and is defined as:

$$\frac{|Demand_{Die}|}{A_{total}} \leq \rho_{max} \tag{4}$$

III. Overview

The overall flow of the proposed methodology is illustrated in Fig. 2. The input includes the module-level netlist together with the design constraints, and it generates a face-to-face chiplet solution through interaction of partitioning and floorplanning procedures. Each chiplet comprises multiple functional modules, and the final output specifies both the geometric shape and coordinates of all chiplets. The methodology consists of two principal stages: (1) initial partitioning and (2) 3D floorplanning & partitioning interaction.

In the initial partitioning stage, our objective is to obtain a feasible initial solution that satisfies all constraints. To meet the constraints on vertical interconnect resource limits and chiplet partitioning, we employ a hierarchical partitioning strategy, starting with an inter-die partition and subsequently a multi-way partition at each die. To ensure the partitioning result satisfies the area and utilization requirements, we designed an unbalanced chiplet partition method under area and utilization constraints for intra-die multi-way partitioning.

In the floorplanning & partitioning interaction stage, chiplet floorplanning is determined with the objectives of minimizing wirelength and reducing inter-die connection overflow. To allocate vertical interconnect space under hybrid bonding effectively, the method incorporates dynamic adjustments of chiplet aspect ratios and inter-die overlaps, which supports the rationale and effectiveness of the overflow optimization. Furthermore, to balance vertical bonding overflow between inter-die connection supply and demand while further reducing wirelength, we introduce a distance-aware floorplanning scheme jointly optimized with repartitioning.

In summary, the proposed framework offers a feasible and effective solution for 3D chiplet design with vertical bonding structures. By explicitly incorporating bonding requirements into the partitioning and floorplanning process, it achieves a balance between the supply and demand of vertical interconnections. The following sections present the proposed methods and strategies in detail.

IV. 3D Chiplet Partitioning & Floorplanning

A. Initial Partition

The initial partitioning phase aims to generate a feasible solution satisfying constraints and optimizes the objective. The process begins with a min-cut-based 2-way partitioning to assign modules to either top-die or base-die. This method is chosen to minimize inter-die connections to satisfy the overall vertical interconnect density limit. Additionally, modules with explicit hierarchical preferences, such as high-speed computing units or I/O components, are assigned to their designated dies according to fixed constraints from standard partitioning tools.

Traditional partitioning algorithms target balanced partitions and often struggle with unbalanced characteristics of this problem. For instance, tools like KaHyPar [12] can only limit parts to the upper bound a_{max} through its variable block weights setting [13], but cannot enforce the lower bound a_{min} from the given core area and utilization constraints. This limitation can result in an infeasible initial solution. To address this, we propose a utilization-constrained FM local search to legalize the solution while preserving and optimizing the original cutsize.

Given the partitioner output P_0 for a die, we define a violation metric as

$$violation = \sum_{i=1}^{k} \left(\max(0, c(p_i) - a_{max}) + \max(0, a_{min} - c(p_i)) \right) \tag{5}$$

where $c(p_i)$ is the total area of modules in p_i, $a_{\min}$ and $a_{\max}$ are the minimum and maximum capacity for part i. This metric measures the solution's legality. Then, we design a dynamic adjustment strategy to overcome the limitations of fixed-bound FM [14], where each part is constrained by fixed capacity bounds, and any node move that would violate these bounds is rejected. In our problem, the upper bound of a part is flexible and can float according to the current allocation. When a part exceeds its limit temporarily after receiving a movement, we will dynamically update the other parts' capacities and expand the target part's upper bound $a^i_{\max}$, which is computed as:

$$a^i_{max} = \begin{cases} A^i_{max} \times util^i_{max} & i = i^* \\ A^i_{min} \times util^i_{max} & \text{otherwise} \end{cases} \tag{6}$$

$$A^i_{max} = A_{total} - \sum_{i \neq j} A^j_{min} \tag{7}$$

$$A^j_{min} = \max\left(A_{min}, \frac{c(p_j)}{u^j_{max}} \right) \tag{8}$$

where i^* denotes the target chiplet receiving the movement.

By iteratively applying this strategy during FM gain calculation and partition updates, the legality and quality of the initial solution are improved.

B. Inter-die Overlap-Aware Floorplanning

The goal of our floorplanning stage is to optimize chiplets' positions and shapes to minimize HPWL and the overflow of inter-die connections. While most existing methods target 2.5D floorplanning, 3D designs require considering additional factors such as inter-die connections and chiplet overlap. In this section, our co-optimized 3D floorplanning method for the chiplet bonding problem is presented.

The input consists of k_{top} and k_{base} chiplets in top-die and base-die respectively and their core areas. Each chiplet is modeled as a soft module, with position and shape optimized via simulated annealing (SA). Optimizing chiplet overlap is crucial in 3D architectures: sufficient overlap between strongly connected chiplets enables low-cost vertical links via hybrid bonding, while insufficient overlap forces some connections to traverse longer horizontal paths, defined as overflow. Therefore, explicitly considering overlap during floorplanning is critical to improving power efficiency and reducing HPWL.

Based on this, we introduce new optimization terms for reasonable area overlap and shape. The objective function is defined as follows:

$$Cost = \alpha f_{Overlap} + \beta f_{HPWL} + \gamma f_{AR} \tag{9}$$

The parameters α, β, and γ are weighting coefficients that control the trade-off between the three sub-objectives. These weights are normalized such that $\alpha + \beta + \gamma = 1$.

1) Aspect Ratio Optimization: Flexible chiplet shaping provides a key optimization lever for both HPWL and the overflow of inter-die connections by enabling overlap. However, to avoid extreme aspect ratios in chiplet shapes, we introduce an aspect ratio penalty to enforce shape regularity across all chiplets. Specifically, for each chiplet, if its aspect ratio AR_i lies outside $[AR_{min}, AR_{max}]$, a penalty proportional to the deviation from the allowed range is applied. The total aspect ratio penalty is computed as the sum of deviations across all chiplets, normalized to the ideal square shape ($AR = 1$), ensuring chiplets remain compact and well-proportioned.

2) Overlap Optimization: Inter-die connection overflow arises from high local density between chiplet pairs: when chiplets lack sufficient overlap, some nets cannot be connected vertically. To address this and stabilize the optimization, we use a density-based penalty function instead of optimizing $Overflow$ directly. When the locally available supply is insufficient to meet the bonding demand, we adopt a proportional strategy to allocate the limited overlap resources, and design the optimization objective accordingly, defined as Equation 10. The quadratic term emphasizes chiplet pairs with heavy interconnect demand, while a step function $H(\cdot)$ activates the penalty only when the local inter-die density exceeds a threshold $\rho_{\max}$, avoiding unnecessary adjustments for well-overlapping chiplets. Accordingly, the overlap penalty is defined as:

$$f_{Overlap} = \sum_{(i,j)\in D} H\left(\rho_{ij} - \rho_{\max}\right) \cdot \frac{c_{ij}^2}{ov_{ij} + \epsilon}, \tag{10}$$

where D denotes the set of inter-die connections, c_{ij} is the number of connections between chiplets i and j, ov_{ij} is their overlap area, and ϵ is a small constant to prevent division by zero. The local inter-die density and the step function $H(x)$ are defined as (11) and (12), respectively. For consistency with other cost terms, $f_{Overlap}$ is normalized by its maximum value, which corresponds to the scenario with zero overlap area.

$$\rho_{ij} = \frac{c_{ij}}{ov_{ij} + \epsilon}, \tag{11}$$

$$H(x) = \begin{cases} 1, & x > 0, \\ 0, & x \le 0. \end{cases} \tag{12}$$

The quadratic density term ensures that limited overlap is allocated proportionally to inter-die connection counts. For example, consider several base-die chiplets connected to a top-die chiplet, with inter-die connections $\{c_i\}$, assigned overlaps $\{x_i\}$, and a total overlap budget K, as shown in Fig. 3.

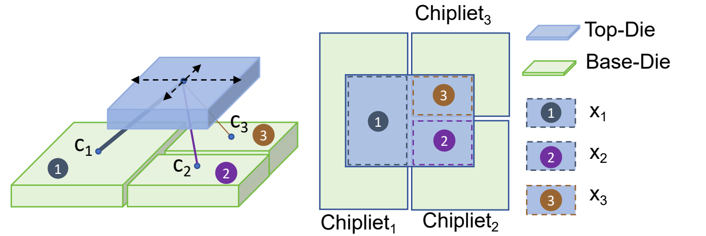

Fig. 3. Overlap-area allocation example.

The optimization problem can be formulated as

$$\min_{x_i > 0} \sum_i \frac{c_i^2}{x_i} \quad \text{s.t.} \quad \sum_i x_i = K, \tag{13}$$

where c_i denotes the number of inter-die connections for base-die chiplet i, x_i is the overlap area allocated to i, and K is the total available overlap area.

Introducing a Lagrange multiplier λ, the Lagrangian is

$$\mathcal{L}(x_1, \ldots, x_n, \lambda) = \sum_i \frac{c_i^2}{x_i} + \lambda \left(\sum_i x_i - K \right). \tag{14}$$

Taking derivatives of $\mathcal{L}$ with respect to each x_i and setting them to zero yields the first-order optimality condition:

$$\frac{\partial \mathcal{L}}{\partial x_i} = -\frac{c_i^2}{x_i^2} + \lambda = 0 \quad \Rightarrow \quad x_i = \frac{c_i}{\sqrt{\lambda}}. \tag{15}$$

Applying $\sum_i x_i = K$ and solving for λ, we obtain:

$$\sqrt{\lambda} = \frac{\sum_j c_j}{K} \quad \text{and thus} \quad x_i = K \frac{c_i}{\sum_j c_j}, \tag{16}$$

which demonstrates that the chiplet overlap is allocated proportionally to the inter-die connection counts.

The overall floorplanning procedure is as follows. It starts with a random initial layout, followed by the simulated annealing (SA) loop that uses two quadtree structures representing the top and base dies. The quadtree hierarchically represents the geometric topology of the floorplan within each die, enabling efficient perturbation and cost evaluation during SA, inspired by SAFFOA [15]. In each iteration, a tree is randomly selected and perturbed, then evaluate and accept the resulting layout according to the SA criterion. After convergence, a distance-aware repartition is invoked to adjust modules based on the current partition and floorplan. The SA loop is resumed after updating the current floorplan configuration, and this cycle of SA and repartition repeats until joint convergence of partitioning and floorplanning. The algorithm finally outputs all chiplets positions along with the corresponding cost.

C. Partitioning–Floorplanning Co-Optimization

After the floorplanning has converged, we further optimize the design by adjusting the partitioning, which can improve both HPWL and overflow due to the tight coupling between demand and supply in these two stages. To jointly consider partitioning and chiplet floorplanning, we propose an efficient distance-aware repartition algorithm that iteratively updates module assignments by moving nodes, while keeping the floorplanning fixed at its current converged state.

Our approach directly targets minimizing total HPWL, rather than relying on coarse metrics such as cutsize that ignore how module locations affect routing resources. For instance, consider moving module 1 from the source part to a destination part (Fig. 4). Although the number of external and internal connections remains the same, the move shortens long-distance connections to other modules, significantly reducing the chiplet-level bounding box, which directly determines the half-perimeter wirelength (HPWL). This demonstrates that evaluating moves based on location-aware connectivity rather than simple cut counts can achieve substantial improvements in HPWL and inter-die overflow.

To address this, we incorporate positional information provided by the converged floorplanning into this step. Our optimization objective is defined as:

$$HPWL = w(e) \cdot d(p(u), p(v)) \tag{17}$$

where $d(p(u), p(v))$ is the Manhattan distance between chiplet centers u and v, and $w(e)$ is the net weight; the total HPWL is computed as the sum over all inter-chiplet connections.

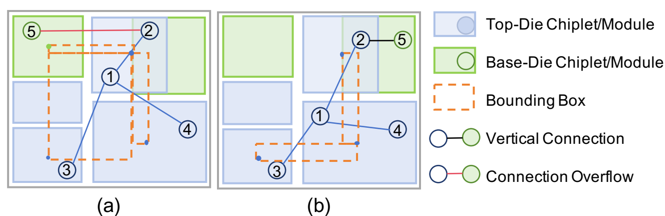

Fig. 4. An illustration of the reduction in bounding boxes and overflow after module movements: (a) current bounding boxes and overflow on two dies; (b) updated bounding boxes and overflow after moving module 1 and module 5.

For chiplets on different dies, their centers are projected onto the same horizontal plane since vertical cost is negligible. This objective enables distance-aware refinement and allows modules to be adjusted flexibly both within a die and across dies. Some inter-die connection overflows arise from large distances between chiplet pairs across the dies and insufficient overlap, representing a subset of long-distance cuts and a primary target for this optimization. By moving such modules to the same die or aligning them for vertical connections, the algorithm not only reduces long-distance HPWL but also effectively mitigates the overflow of inter-die connections, as confirmed by experimental results. Fig. 4 illustrates how the movement of module 5 realigns an overflowing inter-die connection into a vertical connection, significantly reducing HPWL and overflow.

To prevent moves exceeding the available overlap for any chiplet pair, we assign them a large negative gain, steering the optimization away from moves that could increase overflow.

The input of the algorithm is a partition obtained from the initial partition stage, then adjusted based on the remaining inter-die capacities and the distance matrix derived from the floorplanning. Inspired by the FM algorithm, our distance-aware refinement iteratively evaluates candidate node moves in each pass and selects the one that maximizes a location-aware gain, defined as:

$$Gain = \Delta HPWL + OverflowPenalty \tag{18}$$

where $\Delta HPWL$ is the HPWL change and $OverflowPenalty$ penalizes moves exceeding inter-die capacities. Moves exceeding inter-die capacity receive a large negative $OverflowPenalty$; otherwise, it is zero. Through such adjustments, the refinement process prioritizes node movements that improve overall HPWL and the inter-die overflow.

TABLE I
BENCHMARK SCALE AND PARTITIONING PARAMETERS

Circuit	#Cell	#Net	#Module	k_{top}	k_{base}
Industry1	84,546	101,061	25	3	2
Industry2	136,617	158,510	39	4	2
Industry3-1	1,221,712	1,385,980	224	4	2
Industry3-2	1,221,712	1,385,980	224	6	2
Industry4-1	127,303,212	162,365,782	136,678	8	4
Industry4-2	127,303,212	162,365,782	136,678	16	8
Industry4-3	127,303,212	162,365,782	136,678	32	16
Industry5-1	131,642,266	167,896,424	141,605	8	4
Industry5-2	131,642,266	167,896,424	141,605	16	8
Industry5-3	131,642,266	167,896,424	141,605	32	16

TABLE II
FLOORPLANNING & PARTITIONING CO-OPTIMIZATION ANALYSIS

Circuit	w/o Co-optimization			Ours		
	HPWL	Overflow	Time	HPWL	Overflow	Time
Industry1	1.58E9	0.20	1.16	**1.49E9**	**0.09**	2.27
Industry2	**3.66E9**	0.07	1.17	4.04E9	**0.00**	2.29
Industry3-1	**8.50E9**	**0.00**	2.21	**8.50E9**	**0.00**	4.26
Industry3-2	**1.84E10**	**0.03**	2.52	**1.84E10**	**0.03**	4.57
Industry4-1	9.57E11	0.21	250.71	**9.25E11**	**0.15**	303.81
Industry4-2	5.89E12	0.35	292.68	**5.83E12**	**0.14**	395.31
Industry4-3	1.66E13	0.43	345.16	**1.43E13**	**0.35**	679.22
Industry5-1	1.14E12	0.23	330.05	**9.94E11**	**0.11**	388.74
Industry5-2	3.47E12	0.35	349.12	**3.33E12**	**0.26**	454.21
Industry5-3	9.13E12	0.51	447.58	**9.01E12**	**0.29**	677.94
Avg.	1.08	1.68	0.69	**1.00**	**1.00**	1.00

TABLE III
COMPARISON OF CHIPLETPART, TRITONPART, AND OURS

Circuit	ChipletPart				TritonPart				Ours			
	HPWL	Overflow	Meet	Time	HPWL	Overflow	Meet	Time	HPWL	Overflow	Meet	Time
Industry1	1.92E9	0.44	✓	25.31	1.69E9	0.21	✓	2.95	**1.49E9**	**0.09**	✓	2.27
Industry2	5.29E9	**0.00**	✓	34.65	**4.01E9**	**0.00**	✓	31.70	4.04E9	**0.00**	✓	2.29
Industry3-1	9.28E9	**0.00**	✓	36.80	8.76E9	**0.00**	✓	104.33	**8.50E9**	**0.00**	✓	4.26
Industry3-2	2.45E10	0.01	✓	55.57	2.20E10	**0.00**	×	316.49	**1.84E10**	0.03	✓	4.57
Industry4-1	9.64E11	0.27	×	475.52	1.33E12	0.22	×	620.87	**9.25E11**	**0.15**	✓	303.81
Industry4-2	6.33E12	0.41	×	517.11	6.15E12	0.40	×	1766.21	**5.83E12**	**0.14**	✓	395.31
Industry4-3	1.57E13	0.35	×	741.23	2.42E13	0.42	×	1255.55	**1.43E13**	**0.35**	✓	679.22
Industry5-1	1.26E12	0.25	×	569.33	1.45E12	0.26	×	452.76	**9.94E11**	**0.11**	✓	388.74
Industry5-2	4.03E12	0.31	×	602.36	4.92E12	0.35	×	821.34	**3.33E12**	**0.26**	✓	454.21
Industry5-3	9.70E12	0.43	×	738.36	1.01E13	0.43	×	1591.13	**9.01E12**	**0.29**	✓	677.94
Avg.	1.10	1.76		1.30	1.40	1.63		2.39	**1.00**	**1.00**		1.00

V. EXPERIMENTAL RESULT

A. Benchmarks and Settings

We implemented our algorithms in C/C++ and Python and conducted all experiments on a Linux workstation equipped with dual Intel Xeon Silver 4215R CPUs (3.2 GHz). The evaluation is performed on industrial benchmarks, as summarized in Table I, where "k_{top}" and "k_{base}" indicate the number of chiplets assigned to the top and base dies, respectively. To further evaluate adaptability to different chiplet partitioning requirements, we applied additional chiplet number configurations to the cases "Industry3", "Industry4" and "Industry5".

In our experiments, for the basic chiplet constraint settings described in Section II, the minimum area A_{min} is set to provide sufficient space for chiplet partitioning, ranging from $A_{total}/(2k)$ to A_{total}/k. The minimum and maximum chiplet utilization, $util_{min}$ and $util_{max}$, are set based on the average utilization of the corresponding die, with a margin of $\pm 20\%$. The maximum inter-die bonding density ρ_{max} is set to 0.25 per μm^2 with advanced hybrid bonding at a 2 μm pitch.

B. Floorplanning with Repartition Study

Inter-die connection demand can be reduced through partitioning, while supply can be enhanced through floorplanning. To further reduce the gap between demand and supply, repartitioning within the floorplanning process is iteratively repeated until overall convergence. To evaluate the effectiveness of the co-optimization, we conduct a comparative study with the algorithm that excludes the co-optimization step "w/o Co-optimization", under identical experimental settings. Table II reports the results, where the left columns correspond to the variant without co-optimization, and the right columns correspond to the full method.

The results demonstrate that incorporating the co-optimization of floorplanning and repartitioning leads to substantial performance gains. Compared to the variant without co-optimization, our full algorithm achieves an average 8% reduction in HPWL and a 68% decrease in inter-die overflow. These improvements validate the effectiveness of using HPWL as the repartition objective and appropriately weighting the overflow penalty. Although the full method requires additional computational time, the resulting improvements in solution quality make this trade-off worthwhile.

C. Performance Comparison

Table III presents a comparison of the proposed method with two recent counterparts, ChipletPart [7] and TritonPart [16]. To align ChipletPart with our 3D chiplet scenario, we modified its objectives and settings to match our problem formulation. TritonPart is also included, as hypergraph partitioning tools are seen to be adaptable for chiplet-level partitioning [17]. Since neither method natively supports multi-die floorplanning, both are combined with our floorplanner without repartition to generate the final chiplet solution.

To provide a comprehensive comparison of partitioning performance in 3D chiplet scenarios, we evaluate the methods using the following metrics: "HPWL", inter-die connection "Overflow", and compliance with design constraints "Meet". The "HPWL" measures the total wirelength of chiplet interconnects. The "Overflow" measures the fraction of inter-die nets that exceed the available hybrid-bonding capacity, reflecting the feasibility of vertical integration. The "Meet" metric evaluates whether the final solution satisfies all basic constraints defined in Section II. In addition, we report runtime in seconds to assess the computational cost of each method.

As shown in Table III, our method meets all design constraints and achieves lower HPWL across almost all circuits, with an average reduction of 10%$\sim$40% compared to ChipletPart and TritonPart. Inter-die connection overflow is also significantly reduced, by 63%$\sim$76% on average. In terms of runtime, our approach is on average 1.76$\times\sim$2.39$\times$ faster. Overall, these results demonstrate that our method effectively balances design constraints, wirelength, and inter-die connectivity, confirming its superiority for industrial 3D chiplet designs.

VI. CONCLUSION

In this paper, we propose a unified framework for simultaneous 3D chiplet partitioning and floorplanning, managing their tight coupling to address the high demand for vertical interconnects of 3D hybrid bonding. Experimental results on industry benchmarks demonstrate that the proposed approach outperforms recent approaches in terms of wirelength, vertical bonding overflow, and compliance with design constraints.

REFERENCES

[1] R. Agarwal, P. Cheng, P. Shah, B. Wilkerson, R. Swaminathan, J. Wuu, and C. Mandalapu, "3d packaging for heterogeneous integration," in *2022 IEEE 72nd Electronic Components and Technology Conference (ECTC)*, 2022, pp. 1103–1107.

[2] C. Hou, "Physical design for 3d chiplets and system integration," in *Proceedings of the 2020 International Symposium on Physical Design*, ser. ISPD '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 73. [Online]. Available: https://doi.org/10.1145/3372780.3378167

[3] W. Bao, J. Zhang, H. Wong, J. Liu, and W. Li, "Emerging copper-to-copper bonding techniques: Enabling high-density interconnects for heterogeneous integration," *Nanomaterials*, vol. 15, no. 10, 2025. [Online]. Available: https://www.mdpi.com/2079-4991/15/10/729

[4] A. Smith, G. H. Loh, M. J. Schulte, M. Ignatowski, S. Naffziger, M. Mantor, M. F. N. Kalyanasundharam, V. Alla, N. Malaya, J. L. Greathouse, E. Chapman, and R. Swaminathan, "Realizing the amd exascale heterogeneous processor vision : Industry product," in *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, 2024, pp. 876–889.

[5] F. Li, Y. Wang, Y. Wang, M. Wang, Y. Han, H. Li, and X. Li, "Chipletizer: Repartitioning socs for cost-effective chiplet integration," in *2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2024, pp. 58–64.

[6] S. Chen, S. Li, Z. Zhuang, S. Zheng, Z. Liang, T.-Y. Ho, B. Yu, and A. L. Sangiovanni-Vincentelli, "Floorplet: Performance-aware floorplan framework for chiplet integration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 6, pp. 1638–1649, 2024.

[7] A. Graening, P. Gupta, A. B. Kahng, B. Pramanik, and Z. Wang, "Chipletpart: Cost-aware partitioning for 2.5d systems," 2025. [Online]. Available: https://arxiv.org/abs/2507.19819

[8] J. Cong, G. Luo, J. Wei, and Y. Zhang, "Thermal-aware 3d ic placement via transformation," in *2007 Asia and South Pacific Design Automation Conference*, 2007, pp. 780–785.

[9] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "Placement-driven partitioning for congestion mitigation in monolithic 3d ic designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 4, pp. 540–553, 2015.

[10] B. W. Ku, K. Chang, and S. K. Lim, "Compact-2d: A physical design methodology to build commercial-quality face-to-face-bonded 3d ics," in *Proceedings of the 2018 International Symposium on Physical Design*, ser. ISPD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 90–97. [Online]. Available: https://doi.org/10.1145/3177540.3178244

[11] J. Lu, H. Zhuang, I. Kang, P. Chen, and C.-K. Cheng, "eplace-3d: Electrostatics based placement for 3d-ics," ser. ISPD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 11–18. [Online]. Available: https://doi.org/10.1145/2872334.2872361

[12] S. Schlag, T. Heuer, L. Gottesbüren, Y. Akhremtsev, C. Schulz, and P. Sanders, "High-quality hypergraph partitioning," *ACM J. Exp. Algorithmics*, vol. 27, Feb. 2023. [Online]. Available: https://doi.org/10.1145/3529090

[13] S. Schlag *et al.*, "Kahypar: Karlsruhe hypergraph partitioning framework," https://github.com/kahypar/kahypar, 2025, accessed: 2025-09-13.

[14] C. Fiduccia and R. Mattheyses, "A linear-time heuristic for improving network partitions," in *19th Design Automation Conference*, 1982, pp. 175–181.

[15] O. He, S. Dong, J. Bian, S. Goto, and C.-K. Cheng, "A novel fixed-outline floorplanner with zero deadspace for hierarchical design," in *2008 IEEE/ACM International Conference on Computer-Aided Design*, 2008, pp. 16–23.

[16] I. Bustany, G. Gasparyan, A. B. Kahng, I. Koutis, B. Pramanik, and Z. Wang, "An open-source constraints-driven general partitioning multi-tool for vlsi physical design," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2023, pp. 1–9.

[17] M. A. Kabir and Y. Peng, "Chiplet-package co-design for 2.5d systems using standard asic cad tools," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020, pp. 351–356.