Analytical Heterogeneous Die-to-Die 3-D Placement With Macros

Yuxuan Zhao, Peiyu Liao, Siting Liu, Jiaxi Jiang, Yibo Lin, *Member, IEEE*, and Bei Yu, *Senior Member, IEEE*

***Abstract*—This article presents an innovative approach to 3-D mixed-size placement in heterogeneous face-to-face (F2F) bonded 3-D ICs. We propose an analytical framework that utilizes a dedicated density model and a bistratal wirelength model, effectively handling macros and standard cells in a 3-D solution space. A novel 3-D preconditioner is developed to resolve the topological and physical gap between macros and standard cells. Additionally, we propose a mixed-integer linear programming (MILP) formulation for macro rotation to optimize wirelength. Our framework is implemented with full-scale GPU acceleration, leveraging an adaptive 3-D density accumulation algorithm and an incremental wirelength gradient algorithm. Experimental results on ICCAD 2023 contest benchmarks demonstrate that our framework can achieve 5.9% quality score improvement compared to the first-place winner with 4.0× runtime speedup. Additional experiments on modern RISC-V designs further validate the generalizability and superiority of our framework.**

***Index Terms*—3-D integrated circuits, GPU acceleration, physical design, placement.**

I. Introduction

As TECHNOLOGY scaling approaches its physical limits, 3-D integrated circuits (3-D ICs) have emerged as a viable solution for extending Moore's Law. By stacking multiple dies vertically, 3-D ICs can integrate devices, such as CMOS, SRAM, and RRAM with one or multiple technology nodes onto a single chip [1]. However, circuit components like memory and analog blocks become the bottleneck of integration, which tend to scale at a slower pace than their logic counterpart. Heterogeneous 3-D ICs can benefit by using advanced technology nodes for standard cells without worrying about the technology node of the hard IPs, achieving better performance, area, and cost. Intel's Meteor Lake [2] serves as a notable example of such technology adoption.

Manuscript received 5 March 2024; revised 28 June 2024; accepted 9 August 2024. Date of publication 14 August 2024; date of current version 22 January 2025. This work was supported in part by the Research Grants Council of Hong Kong, SAR, under Grant CUHK14210723 and Grant CUHK14211824, and in part by the AI Chip Center for Emerging Smart Systems (ACCESS), Hong Kong. This article was recommended by Associate Editor V. Pavlidis. *(Yuxuan Zhao and Peiyu Liao contributed equally to this work.) (Corresponding author: Bei Yu.)*

Yuxuan Zhao, Peiyu Liao, Siting Liu, Jiaxi Jiang, and Bei Yu are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, SAR, China (e-mail: byu@cse.cuhk.edu.hk).

Yibo Lin is with the School of Integrated Circuits and the Beijing Advanced Innovation Center for Integrated Circuits, Peking University, Beijing 100871, China, and also with the Institute of Electronic Design Automation, Peking University, Wuxi 405800, China.

Digital Object Identifier 10.1109/TCAD.2024.3444716

There are three main variants of 3-D ICs: 1) through-silicon-via (TSV)-based; 2) monolithic; and 3) face-to-face (F2F) bonding. The large pitches and parasitics of TSVs [3] restrict TSV-based 3-D ICs to few interdie connections, thereby limiting the performance benefits. While monolithic 3-D (M3D) integration enables fine-grained vertical interconnects [4], [5], the manufacturing yield is low due to the sophisticated process steps. F2F bonded 3-D ICs [6], [7] are made up of two prefabricated dies connected via hybrid bonding terminals (HBTs) on the top-most metal layer. The ease of manufacturing and the small size of bonding terminals enable a high-integration density at a low cost, making it a preferred approach [8], [9].

3-D placement remains a challenging problem in the physical design flow of 3-D ICs. Existing methodologies are either designed for standard cell 3-D placement or aim to handle macros and standard cells together in mixed-size designs. Recent placers [10], [11], [12] for F2F bonded 3-D ICs focus on standard cell placement [13]. iPL-3-D [10] models the problem using bilevel programming to optimize partitioning and placement alternatively. To model the heterogeneous integration, MTWA [11] uses a sigmoid-based pin transition function, and the bistratal wirelength model [12] proposes the finite difference approximation for accurate wirelength modeling. However, with memory-intensive applications, such as machine learning proliferating, numerous memory macros are integrated into modern processors and accelerators to enhance performance. A 3-D placer capable of handling both standard cells and macros is essential to obtain the expected benefits [9] for these mixed-size designs.

Existing 3-D mixed-size placers form two broad categories: 1) pseudo-3-D and 2) true-3-D. Pseudo-3-D placers [5], [8], [14], [15], [16] separate the partitioning and placement phases, and adopt 2-D placement tools to determine instance locations. Cascade2-D [14] implements an M3D design using the partitioning-first flow. To fully utilize the physical information, recent partitioning-last flows [8], [16] perform tier partitioning after an intermediate placement stage. These design flows introduce placement blockages to consider preplaced macros from the floorplan stage. However, pseudo-3-D placers cannot fully explore the overall solution space, and their performance is particularly sensitive to partitioning results, exacerbated by the presence of macros.

Differently, true-3-D placers [17], [18], [19] relax the discrete tier partitioning and adopt analytical approaches. The analytical placers perform mixed-size placement in a 3-D cuboid region based on the smoothed wirelength model and

density model. NTUPlace3-3-D [18] utilizes a bell-shaped density model considering TSV insertion. The state-of-the-art (SOTA) analytical 3-D placer, ePlace-3-D [19], models the density constraint as a 3-D electrostatic field. Despite their efficiency in handling macros and standard cells, existing true-3-D placers focus on TSV minimization without an accurate model for heterogeneous integration.

In summary, the aforementioned previous approaches are hardly applicable to mixed-size designs in heterogeneous F2F bonded 3-D ICs. Most pseudo-3-D placers [14], [16] rely on the FM min-cut partitioning algorithm [20] and fail to utilize the advantages of F2F bonding technology. Conventional true-3-D placers [17], [18], [19] do not support heterogeneous integration and employ a simplistic 3-D net bounding box wirelength model, neglecting the wirelength reduction through interdie connections. Although recent studies [11], [12] have improved wirelength models to better-accommodate heterogeneous technology nodes, the placers lack key innovations for the significant topological and physical difference between macros and standard cells, resulting in challenges with optimization convergence.

GPU acceleration has achieved great success in 2-D placement [21], [22]. Liao et al. [12] pioneered GPU acceleration for 3-D placement, but their acceleration techniques are limited to standard cell placement, resulting in significant load balancing issues in mixed-size scenarios. In addition, the approach they employed for bistratal wirelength model [12] is hampered by high-computational complexity. Innovations are needed for efficient 3-D mixed-size placement on GPU.

In this article, we propose an analytical approach to 3-D mixed-size placement in heterogeneous F2F bonded 3-D ICs. Leveraging a dedicated density model and a bistratal wirelength model, our framework effectively optimizes instance partitioning and locations in a 3-D solution space. Our contributions are summarized as follows.

1) We propose an analytical 3-D mixed-size placement framework with a density model and a bistratal wirelength model, incorporating a novel 3-D preconditioner, for heterogeneous F2F bonded 3-D ICs.
2) A mixed-integer linear programming (MILP) formulation is proposed to assign macro rotations for wirelength optimization.
3) We implement our framework with full-scale GPU acceleration, leveraging adaptive 3-D density accumulation and incremental wirelength gradient algorithms.
4) Experimental results on ICCAD 2023 contest benchmarks demonstrate that our framework can achieve 5.9% quality score improvement over the first-place winner with 4.0× runtime speedup.
5) We also evaluated our framework on modern RISC-V designs. Compared to the baseline, our placer achieves 20% better wirelength with 12.0× runtime speedup.

The remainder of this article is organized as follows. Section II provides the background and the problem formulation. Section III presents the overall mixed-size placement flow of the proposed framework for heterogeneous F2F bonded 3-D ICs. In Section IV, we detail our density and wirelength algorithms. Section V presents experimental results and related analysis, followed by conclusion in Section VI.

II. Preliminaries

A. 3-D Analytical Global Placement

Given a netlist (V, E) where $V = \{v_1, \ldots, v_n\}$ is the instance set and $E = \{e_1, \ldots, e_m\}$ is the net set, all the instances are placed within a 3-D cuboid region $\Omega = [0, d_x] \times [0, d_y] \times [0, d_z]$. And we use $V_M \subset V$ and $E_M \subset E$ to denote the movable macros and the nets connecting the macros. Let $\boldsymbol{v} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ denote the physical coordinates of the instances. The placement objective is to minimize the total half-perimeter wirelength (HPWL) while satisfying the target density constraints. Conventionally, the 3-D HPWL is adopted as the objective function defined below.

Definition 1 (3-D HPWL): Given instance locations $\boldsymbol{v} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, the 3-D HPWL of any net $e \in E$ is given by

$$W_e(\boldsymbol{v}) = p_e(\boldsymbol{x}) + p_e(\boldsymbol{y}) + \alpha \cdot p_e(\boldsymbol{z}) \tag{1}$$

where $p_e(\boldsymbol{u}) = \max_{v_i \in e} u_i - \min_{v_i \in e} u_i$ denotes the partial HPWL along one axis, and a weight factor $\alpha \geq 0$ is introduced for the vertical interconnects in 3-D ICs.

To model the density constraints, the cuboid region Ω is uniformly divided into $N_x \times N_y \times N_z$ bins denoted as set B. And the density ρ_b in each bin should not exceed the target density ρ_t. The nonlinear placement optimization is formulated as

$$\min_{\boldsymbol{v}} \sum_{e \in E} W_e(\boldsymbol{v}) \quad \text{s.t. } \rho_b(\boldsymbol{v}) \leq \rho_t \quad \forall b \in B. \tag{2}$$

Analytical methods conduct the 3-D global placement using gradient-based optimization. As $p_e(\cdot)$ in 3-D HPWL is nonsmooth and nonconvex, it is approximated by a differentiable wirelength model, e.g., the weighted-average model [18] given a smoothing parameter $\gamma > 0$

$$\hat{p}_e(\boldsymbol{x}) = \frac{\sum_{v_i \in e} x_i \, \mathrm{e}^{\frac{1}{\gamma} x_i}}{\sum_{v_i \in e} \mathrm{e}^{\frac{1}{\gamma} x_i}} - \frac{\sum_{v_i \in e} x_i \, \mathrm{e}^{-\frac{1}{\gamma} x_i}}{\sum_{v_i \in e} \mathrm{e}^{-\frac{1}{\gamma} x_i}}. \tag{3}$$

Similarly, a density model $U(\cdot)$ relaxes all the $|B|$ constraints in (2) and evaluates the overall density penalty within the entire region Ω. The SOTA density model $U(\cdot)$ is the eDensity family [19], [23], [24] based on electrostatics field, converting instances $v_i \in V$ to charges. The electric force spreads charges toward the equilibrium state, producing a globally even density distribution. Putting the density penalty into the wirelength objective, the 3-D analytical global placement is formulated as the following unconstrained optimization:

$$\min_{\boldsymbol{v}} \sum_{e \in E} \hat{W}_e(\boldsymbol{v}) + \lambda \hat{U}(\boldsymbol{v}) \tag{4}$$

where $\hat{W}_e(\cdot)$ is the smoothed wirelength model, $\hat{U}(\cdot)$ is the smoothed density model, and λ is the density weight introduced as the Lagrangian multiplier of the density constraints.

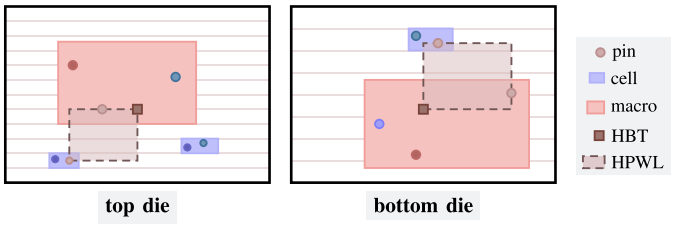


Fig. 1. D2D wirelength of a net is the sum of the wirelength of the top net and bottom net. HBTs are on the top-most layer for both dies. Pins connected by a net are in the same color.

B. Problem Formulation

This article considers the 3-D mixed-size placement problem specified in the ICCAD 2023 contest [25]. We intend to determine the locations of standard cells and macros on the two dies with the same or different technology nodes, and insert HBTs for die-to-die (D2D) vertical connections so that the total D2D wirelength and HBT cost are minimized while the following constraints are satisfied.

1) All the instances must be nonoverlapping, and the standard cells must be aligned to rows and sites. HBT spacing constraints must be satisfied.
2) All the instances are placed on either top or bottom die, and the maximum utilization of each die must be satisfied.
3) For any crossing-die net, one and only one HBT is inserted for vertical connection.
4) All the standard cells cannot be rotated or mirrored. Macros, on the other hand, can be rotated with 0°, 90°, 180°, and 270° counterclockwise without mirroring.

It is worth noting that the cells and macros may be fabricated using different technology nodes on different dies, i.e., the instance height, width, and pin location would be different. And the center points of the HBTs are included in the wirelength calculation for accurate modeling of F2F bonded ICs, as illustrated in Fig. 1. Therefore, the instance partition $\boldsymbol{\delta}$ must be explicitly considered.

A partition is determined by a binary vector $\boldsymbol{\delta} \in \{0,1\}^n$, where $\delta_i = 0$ indicates that $v_i \in V$ is placed on the bottom die, otherwise on the top die. δ_i can be derived from the instance center z-coordinate by $\delta_i = \mathbb{1}_{\mathbb{R}^+}(z_i - [d_z/2])$, where $\mathbb{1}_{\mathbb{R}^+}(\cdot)$ is the indicator function of positive real numbers. Accordingly, the HBT t_e is inserted for crossing-die net e with $C_e(\boldsymbol{\delta}) = \max_{v_i \in e} \delta_i - \min_{v_i \in e} \delta_i = 1$, which means the net e connects the instances placed on the different dies. The D2D wirelength [25] includes the top net $\hat{e}^+ = e^+ \cup \{t_e\}$ and the bottom net $\hat{e}^- = e^- \cup \{t_e\}$ considering both instances and HBTs, where $e^+ = \{v_i \in e : \delta_i = 1\}$ and $e^- = \{v_i \in e : \delta_i = 0\}$.

Definition 2 (D2D HPWL): Given partition $\boldsymbol{\delta}$, the D2D HPWL of net e is defined by $W_e = W_{\hat{e}^+} + W_{\hat{e}^-}$, where

$$\begin{aligned} W_{\hat{e}^+} &= p_{\hat{e}^+}(\boldsymbol{x}) + p_{\hat{e}^+}(\boldsymbol{y}) \\ W_{\hat{e}^-} &= p_{\hat{e}^-}(\boldsymbol{x}) + p_{\hat{e}^-}(\boldsymbol{y}). \end{aligned} \tag{5}$$

If $C_e(\boldsymbol{\delta}) = 0$, it reduces to the 2-D net HPWL without the HBT.

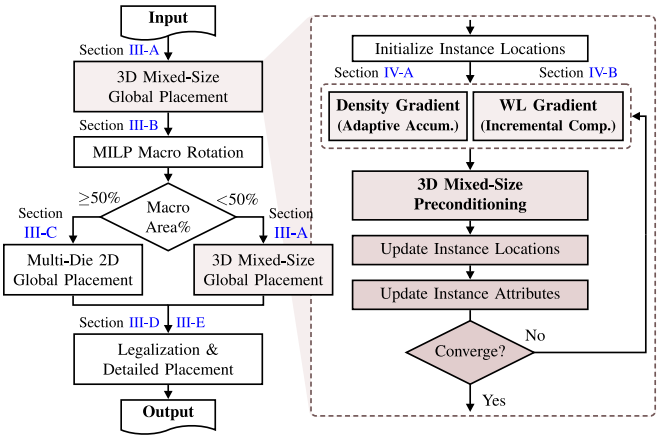


Fig. 2. Overall 3-D mixed-size placement flow.

Based on Definition 2, we formally define the 3-D D2D placement problem as follows:

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \ & \sum_{e \in E} W_e(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) + \beta \sum_{e \in E} C_e(\boldsymbol{\delta}) \\ \text{s.t.} \ & \rho_b(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \le \rho_t \quad \forall b \in B \\ & \delta_i = \mathbb{1}_{\mathbb{R}^+}\left(z_i - \frac{d_z}{2}\right) \quad \forall v_i \in V \\ & \theta_i \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\} \quad \forall v_i \in V_M \\ & \text{legality constraints} \end{aligned} \tag{6}$$

where $W_e(\cdot)$ is the D2D HPWL and $C_e(\boldsymbol{\delta})$ is the crossing-die net indicator. β denotes the cost of each HBT provided by the design specification, and θ_i denotes the rotation of each macro. Following the 3-D analytical approaches, we transform the above problem into unconstrained optimization in (4). We adopt the bistratal wirelength model [12] and eDensity3-D model [19], respectively. Dedicated customizations are proposed for accurate modeling of heterogeneous mixed-size designs, and full GPU acceleration is implemented in our framework for ultrafast performance.

III. Proposed 3-D Placement Framework

The overall placement flow of our framework is illustrated in Fig. 2, which consists of four stages. First, our framework performs 3-D mixed-size global placement (Section III-A) to optimize the instance partitioning and locations simultaneously with the initial macro orientation 0°. Second, we optimize the macro rotations based on the physical information of the initial 3-D placement solution. We propose a MILP formulation (Section III-A) to minimize the wirelength. Then, we perform global placement again with the optimized macro rotations to further improve wirelength. Our framework applies multidie 2-D global placement (Section III) for the designs of macro area ratio larger than 50%, which avoids the large macro density obstacle and leads to better-macro placement. For the designs with macro area ratio smaller than 50%, we perform 3-D mixed-size global placement to explore the entire solution space for better wirelength. The macro area ratio is calculated using the technology information of the top die, which has a

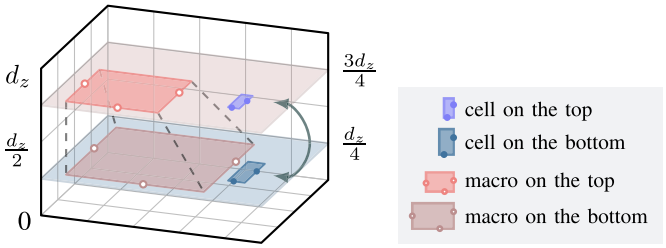


Fig. 3. Our density model and wirelength model consider instance partitioning explicitly for accurate modeling of heterogeneous technology nodes. The instance attributes are updated dynamically in the global placement stage. The macro size transition is smoothed for stable density optimization.

smaller feature size. At last, we apply die-by-die legalization and detailed placement to obtain the final placement result.

A. 3-D Mixed-Size Global Placement

The heterogeneous F2F bonded ICs bring unique challenges for global placement. The instance attributes, including size and pin offsets, are different on the two dies for heterogeneous technology nodes, and the macros show particularly large variation. Such property requires our density model and wirelength model to consider the instance partitioning explicitly for accurate modeling.

Electrostatics-Based Density Model: The SOTA eDensity3-D [19] model sets the electric quantity q_i as the physical volume of instance v_i. To consider the heterogeneous technology nodes, we update the attributes of instance dynamically according to the z-coordinate, i.e., $\delta_i = \mathbb{1}_{\mathbb{R}^+}(z_i - [d_z/2])$, as shown in Fig. 3. Let w_i^+ and h_i^+ denote the instance width and height on the top die, and w_i^- and h_i^- on the bottom die. The dynamic width w_i and height h_i can be derived as

$$\begin{aligned} w_i &= \delta_i w_i^+ + (1-\delta_i) w_i^- \\ h_i &= \delta_i h_i^+ + (1-\delta_i) h_i^- . \end{aligned} \tag{7}$$

To accommodate the D2D placement, we set all instances with the same depth $d = (1/2)d_z$ so that the instances can be distributed to exactly two dies. Although the update scheme provides accurate heterogeneous information, the step transition introduces discreteness for density optimization. The impact of standard cells is small, but the large variation in macro size incurs sudden changes in the density map, as shown in Fig. 3, resulting in challenges with convergence. For any macro $v_i \in V_M$, we propose to linearly transform macro width and height as

$$\begin{aligned} w_i &= \left(\frac{2z_i}{d_z} - \frac{1}{2}\right) w_i^+ + \left(\frac{3}{2} - \frac{2z_i}{d_z}\right) w_i^- \\ h_i &= \left(\frac{2z_i}{d_z} - \frac{1}{2}\right) h_i^+ + \left(\frac{3}{2} - \frac{2z_i}{d_z}\right) h_i^- . \end{aligned} \tag{8}$$

The movable range of z_i is $[(d_z/4), (3d_z/4)]$ based on our depth setting. While [11], [26] adopt nonlinear size transformation for both standard cells and macros, our approach only scales the macro size for more accurate heterogeneous modeling.

The eDensity3-D models the density penalty $\hat{U}$ as the total potential energy of the system $\hat{U}(\boldsymbol{v}) = \sum_{v_i \in V} q_i \phi_i(\boldsymbol{v})$. It computes the potential map $\phi(\boldsymbol{v})$ by solving the 3-D Poisson's equation

$$\begin{aligned} \Delta\phi(\boldsymbol{v}) &= -\rho(\boldsymbol{v}), \quad \boldsymbol{v} \in \Omega \\ \hat{\boldsymbol{n}} \cdot \nabla\phi(\boldsymbol{v}) &= 0, \quad \boldsymbol{v} \in \partial\Omega . \end{aligned} \tag{9}$$

eDensity3-D solves the Poisson's equation by efficient spectral methods [19]. Let $(\omega_j, \omega_k, \omega_l) = ([j\pi/d_x], (k\pi/d_y), [l\pi/d_z])$ denote the frequency indices. The density frequency coefficients a_{jkl} are computed as

$$a_{jkl} = \frac{1}{N} \sum_{x,y,z} \rho(x,y,z) \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z) \tag{10}$$

where the denominator $N = N_x N_y N_z$ denotes the total number of bins. And according to (9), the potential map solution $\phi(x,y,z)$ under constraint $\int_\Omega \phi(\boldsymbol{v})\, \mathrm{d}\Omega = 0$ is given by

$$\phi(x,y,z) = \sum_{j,k,l} \frac{a_{jkl}}{\omega_j^2 + \omega_k^2 + \omega_l^2} \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z). \tag{11}$$

By differentiating (11), we have the electric field $\boldsymbol{E}(x,y,z) = (E_x, E_y, E_z)$ shown as follows:

$$\begin{aligned} E_x &= \sum_{j,k,l} \frac{a_{jkl}\omega_j}{\omega_j^2 + \omega_k^2 + \omega_l^2} \sin(\omega_j x) \cos(\omega_k y) \cos(\omega_l z) \\ E_y &= \sum_{j,k,l} \frac{a_{jkl}\omega_k}{\omega_j^2 + \omega_k^2 + \omega_l^2} \cos(\omega_j x) \sin(\omega_k y) \cos(\omega_l z) \\ E_z &= \sum_{j,k,l} \frac{a_{jkl}\omega_l}{\omega_j^2 + \omega_k^2 + \omega_l^2} \cos(\omega_j x) \cos(\omega_k y) \sin(\omega_l z). \end{aligned} \tag{12}$$

The above spectral equations can be efficiently solved using fast Fourier transform (FFT) with $O(N \log N)$ complexity. However, during the forward phase, we need to compute the density map $\rho(x,y,z)$, and during the backward phase, the electric force is $\nabla \hat{U}_i = q_i \boldsymbol{E}_i$, which both require the density accumulation over the 3-D grid bins. As a result, the density accumulation becomes the runtime bottleneck.

Two types of dummy fillers are inserted into our placement system.

1) *z-Fixed Fillers:* These fillers are used to manage maximum utilization constraints and maintain fixed z-coordinates. Fillers on the same die are equally sized (cuboid) with depth $d = (1/2)d_z$. We set the total volume of top z-fixed fillers vol_f^+ and bottom z-fixed fillers vol_f^- as

$$\begin{aligned} \mathrm{vol}_f^+ &= \tfrac{1}{2} d_x d_y d_z \left(1 - u^+\right) \\ \mathrm{vol}_f^- &= \tfrac{1}{2} d_x d_y d_z \left(1 - u^-\right) \end{aligned} \tag{13}$$

where u^+ and u^- are the maximum utilization rate for the top die and the bottom die, respectively. The top z-fixed fillers are initialized with $z_i = (3d_z/4)$, and bottom z-fixed fillers are initialized with $z_i = (d_z/4)$. During the optimization, these fillers' z-gradients are set to zero. Once a die's maximum utilization rate is exceeded, the fillers will push the instances to the other die.

2) *Free Fillers:* To address the potential white space, we insert free fillers that are initialized at the center of the region Ω and follow a normal distribution. Unlike the z-fixed fillers,

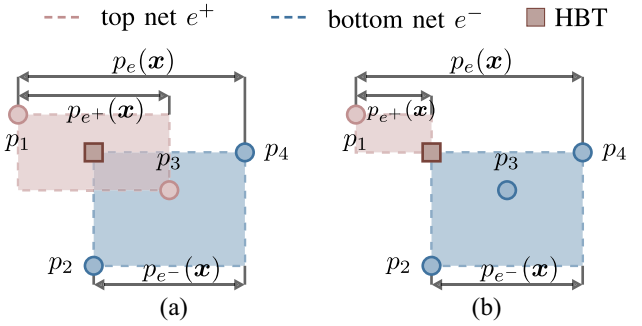


Fig. 4. Illustration of D2D HPWL wirelength in x-axis, the y-axis is similar. (a) 3-D HPWL is inconsistent with the D2D HPWL. The x-axis D2D HPWL is larger than the x-axis HPWL of the entire bounding box. (b) With the planar locations fixed, changing the pin partition can significantly reduce the x-axis D2D HPWL in some cases.

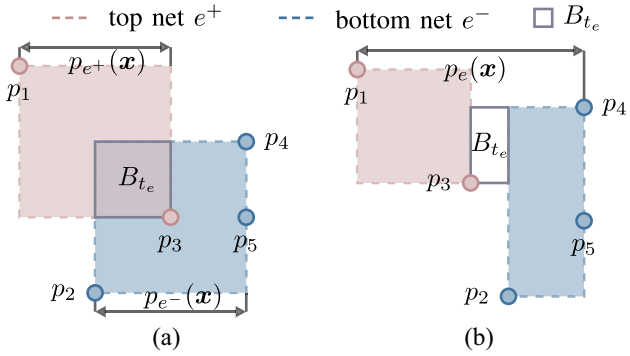


Fig. 5. Optimal region B_{t_e} is the region bounded by the median values of the top net box B_e^+ and bottom net box B_e^-.HBT t_e placed outside B_{t_e} will introduce extra wirelength. (a) If B_e^+ and B_e^- overlap in x-axis, the minimal x-axis D2D HPWL is $p_{e^+}(\boldsymbol{x}) + p_{e^-}(\boldsymbol{x})$. (b) If B_e^+ and B_e^- have no overlap in x-axis, the minimal x-axis D2D HPWL is $p_e(\boldsymbol{x})$.

free fillers can move freely in the z-direction. The total volume of free fillers vol_{fr} is calculated as

$$\text{vol}_{fr} = \max\left\{d_x d_y d_z - \left(\text{vol}_f^+ + \text{vol}_f^- + \text{vol}_a\right), 0\right\} \quad (14)$$

where vol_a is the total volume of all the instances. We randomly initialize all the instances at the center following a normal distribution. The total instance volume vol_a is calculated based on this initial position.

Bistratal Wirelength Model: According to the objective in (6), the primary optimization goal is to minimize the D2D wirelength in (5), and a small HBT cost β is specified by the design to encourage more usage of HBTs. The HBT cost can be naturally modeled by $p_e(z)$, reflecting the cut size. However, the traditional 3-D HPWL model in (1) cannot match the D2D wirelength in (5) for the planar wirelength, which contributes most to the objective.

As illustrated in Fig. 3(a), the x-axis 3-D HPWL $p_e(\boldsymbol{x})$ is smaller than x-axis D2D HPWL $p_{e^+}(\boldsymbol{x}) + p_{e^-}(\boldsymbol{x})$. In fact, the D2D HPWL can be $2\times$ of the 3-D HPWL if the bounding boxes of the top net and the bottom net are the same, and only if the bounding boxes have no overlap as shown in Fig. 3(b), they are of the same value. The error of the 3-D HPWL arises from the negligence of the HBTs in D2D placement. To decide the locations of the HBTs, we first introduce the optimal region for an HBT t_e. Given the bounding box $B_e^+ = [x_{\min}^+, x_{\max}^+] \times [y_{\min}^+, y_{\max}^+]$ for the top net and $B_e^- = [x_{\min}^-, x_{\max}^-] \times [y_{\min}^-, y_{\max}^-]$ for the bottom net, the optimal region $B_{t_e} = [x'_{\min}, x'_{\max}] \times [y'_{\min}, y'_{\max}]$ for the HBT t_e is defined as

$$\begin{aligned} x'_{\min} &= \min\left\{\max\left\{x_{\min}^+, x_{\min}^-\right\}, \min\left\{x_{\max}^+, x_{\max}^-\right\}\right\} \\ x'_{\max} &= \max\left\{\max\left\{x_{\min}^+, x_{\min}^-\right\}, \min\left\{x_{\max}^+, x_{\max}^-\right\}\right\} \end{aligned} \quad (15)$$

and $y'_{\min}, y'_{\max}$ are defined similarly. With the HBT in its optimal region, the D2D HPWL is minimized as illustrated in Fig. 5.

Based on the above analysis, we can derive the minimal D2D wirelength in x-axis

$$W_{e_x}(\boldsymbol{x}) = \max\{p_e(\boldsymbol{x}),\ p_{e^+}(\boldsymbol{x}) + p_{e^-}(\boldsymbol{x})\}. \quad (16)$$

Equation (16) demonstrates how to explicitly optimize D2D wirelength in 3-D global placement. If the bounding boxes B_e^+ and B_e^- overlap, we optimize the HPWL of each partial net as shown in Fig. 4(a). Otherwise, we optimize the entire bounding box at the nonoverlapping direction as shown in Fig. 4(b). Combining y-axis wirelength which is calculated similarly, the bistratal wirelength [12] is defined by $W_{e,\text{Bi}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = W_{e_x}(\boldsymbol{x}) + W_{e_y}(\boldsymbol{y})$. It is worth mentioning that $W_{e,\text{Bi}}(\cdot)$ is also a function of $\boldsymbol{z}$ as z-coordinates determine partial nets e^+, e^- directly. Meanwhile, we also dynamically update the pin offset values in the same approach as (7) to model the heterogeneous technology nodes. Combining the HBT cost, our wirelength model for the 3-D global placement is

$$W_e(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = W_{e,\text{Bi}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) + \alpha p_e(\boldsymbol{z}). \quad (17)$$

Applying the weighted-average model in (3) to $p_e(\cdot)$, we can perform gradient-based optimization on the smoothed objective $\hat{W}_e$. However, the weighted-average model only minimizes $p_e(\boldsymbol{z})$ to reduce the cut size, incapable of optimizing the partition for wirelength. As shown in Fig. 3(b), the D2D wirelength can be greatly reduced with a better distribution of $\boldsymbol{z}$. The smoothed bistratal wirelength $\hat{W}_{e,\text{Bi}}$ is discontinuous with respect to $\boldsymbol{z}$, therefore, the gradient $\nabla_{\boldsymbol{z}} W_{e,\text{Bi}}$ does not exist. To approximate the impact of $\boldsymbol{z}$ on wirelength, we leverage finite difference approximation [12], [27] to perform numerical optimization on $\boldsymbol{z}$ with gradient defined by

$$\begin{aligned} \left(\nabla_{\boldsymbol{z}} \hat{W}_{e,\text{Bi}}\right)_i = \tfrac{4}{d_z}\Big(&W_{e,\text{Bi}}\left(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{z}}_i + \tfrac{3d_z}{4}\boldsymbol{e}_i\right) \\ &- W_{e,\text{Bi}}\left(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{z}}_i + \tfrac{d_z}{4}\boldsymbol{e}_i\right)\Big) \end{aligned} \quad (18)$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are fixed for wirelength evaluation, and $\tilde{\boldsymbol{z}}_i = \boldsymbol{z} \odot (\boldsymbol{1} - \boldsymbol{e}_i)$ and $\boldsymbol{e}_i \in \mathbb{R}^{|V|}$ is the unit vector with the ith entry being 1 and others being 0. For each instance, we perturb z_i with $\Delta z = (d_z/4)$ to alter its partition and evaluate the bistratal wirelength change $\Delta W_{e,\text{Bi}}$. The difference quotient is adopted as the gradient, which provides a local view of wirelength benefits for updating z_i.

To better demonstrate the derivation of (19), we analyze z_i within interval $[(d_z/4), (3d_z/4)]$ for instance $v_i \in V$ and fix all other variables. Then, $W_{e,\text{Bi}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ simplifies to $W_{e,\text{Bi}}(z_i)$ which is a step function that only takes two possible values $W_{e,\text{Bi}}(d_z/4)$ and $W_{e,\text{Bi}}(3d_z/4)$, corresponding to the wirelength when instance v_i is on the bottom and top dies, respectively.

Specifically, we always have $W_{e,\mathrm{Bi}}(z_i) = W_{e,\mathrm{Bi}}(d_z/4)$ if $z_i < (d_z/2)$ and $W_{e,\mathrm{Bi}}(z_i) = W_{e,\mathrm{Bi}}(3d_z/4)$ otherwise. In fact, the finite difference approximation computes the partial derivative $(\nabla_z \hat{W}_{e,\mathrm{Bi}})_i$ as follows for $z_i \in [(1/4)d_z, [1/2]d_z)$:

$$\underset{\frac{1}{4}d_z}{\Delta} W_{e,\mathrm{Bi}}(z_i) = \frac{W_{e,\mathrm{Bi}}\left(z_i + \frac{1}{4}d_z\right) - W_{e,\mathrm{Bi}}(z_i)}{\frac{1}{4}d_z} = \frac{4}{d_z}\left(W_{e,\mathrm{Bi}}\left(\frac{3d_z}{4}\right) - W_{e,\mathrm{Bi}}\left(\frac{d_z}{4}\right)\right). \quad (19)$$

Similarly, if $z_i \in [(1/2)d_z, (3/4)d_z]$, it computes

$$\underset{-\frac{1}{4}d_z}{\Delta} W_{e,\mathrm{Bi}}(z_i) = \frac{W_{e,\mathrm{Bi}}\left(z_i - \frac{1}{4}d_z\right) - W_{e,\mathrm{Bi}}(z_i)}{-\frac{1}{4}d_z} = \frac{4}{d_z}\left(W_{e,\mathrm{Bi}}\left(\frac{3d_z}{4}\right) - W_{e,\mathrm{Bi}}\left(\frac{d_z}{4}\right)\right). \quad (20)$$

Combining (20) and (21), we obtain (19) as a conclusion.

3-D Mixed-Size Preconditioning: Preconditioning is a critical component of numerical optimization which reduces the condition number and stabilizes the optimization process. The large topological and physical difference between macros and standard cells makes the preconditioner indispensable in nonlinear placement optimization.

Equation (19) provides the optimization direction for z. However, $\nabla_z \hat{W}_{e,\mathrm{Bi}}$ is not on the same scale as planar gradients $\nabla_x \hat{W}_{e,\mathrm{Bi}}$ and $\nabla_y \hat{W}_{e,\mathrm{Bi}}$, leading to suboptimal results. Hence, we normalize (19) before applying gradient descent

$$\boldsymbol{g} = \frac{\|\nabla_x \hat{W}_{e,\mathrm{Bi}}\|_1 + \|\nabla_y \hat{W}_{e,\mathrm{Bi}}\|_1}{2\|\nabla_z \hat{W}_{e,\mathrm{Bi}}\|_1} \nabla_z \hat{W}_{e,\mathrm{Bi}} \quad (21)$$

and we use $(\nabla_x \hat{W}_{e,\mathrm{Bi}}, \nabla_y \hat{W}_{e,\mathrm{Bi}}, \boldsymbol{g} + \alpha \nabla_z \hat{p}_e(z))$ as the gradient of our wirelength objective in 3-D mixed-size preconditioning, which ensures the continuity of the optimization process.

In 2-D placement, ePlace [23], [24] adopts the Jacobi preconditioner, which only selects diagonal entries of the Hessian matrix, to perform preconditioning on gradients. Let $f(\boldsymbol{v})$ be the objective function in (4). Considering x direction, the ith diagonal entry of Hessian matrix $\boldsymbol{H}_f = \nabla_x^2 f$ is given by

$$(\boldsymbol{H}_f)_{ii} = \sum_e \frac{\partial^2 \hat{W}_e}{\partial x_i^2} + \lambda \frac{\partial^2 \hat{U}}{\partial x_i^2}. \quad (22)$$

ePlace [23], [24] approximates (23) with $\sum_e (\partial^2 \hat{W}_e / \partial x_i^2) \approx |E_i|$, $(\partial^2 \hat{U}/\partial x_i^2) \approx q_i$, and $(\boldsymbol{H}_f)_{ii} \approx |E_i| + \lambda q_i$, for both standard cells and macros, where $|E_i|$ is the set cardinality of all nets incident to instance $v_i \in V$ and q_i stands for the electric quantity of v_i. Specifically, $|E_i|$ is the number of pins on v_i, and q_i is the corresponding instance area or volume. To better adapt to the third dimension, ePlace3-D [19] removes the first item and only uses $(\boldsymbol{H}_f)_{ii} \approx \lambda q_i$ as the preconditioner for both standard cells and macros.

However, we have $\lambda q_i \ll 1$ at the early global placement stage when the density weight λ is small, resulting in a stability issue and subsequent divergence. The wirelength gradients of macros are significantly larger than those of standard cells, as the macros have a larger number of pins. The large movement of macros frequently perturbs the optimization direction. Therefore, we propose the 3-D mixed-size preconditioner as follows:

$$(\boldsymbol{H}_f)_{ii} \approx \begin{cases} \max\{1, |E_i| + \lambda q_i\}, & \text{if } v_i \in V_M \\ \max\{1, \lambda q_i\}, & \text{otherwise.} \end{cases} \quad (23)$$

Through the mixed-size preconditioning in (24), the macros are allowed to move at the pace of standard cells at the early global placement stage, preventing the optimization from divergence. With density weight λ increasing, the spreading standard cells provide enough physical information to drive the macros to the proper die.

TABLE I
NOTATIONS FOR THE MILP FORMULATION OF MACRO ROTATION ASSIGNMENT

Notations	Descriptions
S_j	a set of standard cell instances connected by e_j
M_j	a set of macro instances connected by e_j
(x_i, y_i)	center location of instance v_i
(o_{ij}^x, o_{ij}^y)	pin offsets on v_i connected by e_j with respect to the center of v_i
(r_i, r_i')	binary variables to encode the rotation of instance v_i

B. MILP Macro Rotation Assignment

The initial 3-D placement solution provides valuable information about the locations and partition of the macros and standard cells. Based on the physical information, we propose a MILP formulation to assign macro rotations to minimize wirelength.

We only need to consider the net set E_M connecting the macros V_M to find the optimal rotation assignment. The notations used in the formulation are summarized in Table I. Consider arbitrary net $e_j \in E_M$ connecting a set of instances, including a set of standard cells S_j and a set of macros M_j. For standard cell $v_k \in S_j$, which cannot be rotated, the coordinates of the pin on v_k connecting to e_j are given by $(x_{kj}, y_{kj}) := (x_k + o_{kj}^x, y_k + o_{kj}^y)$. For the pin location of rotatable macro $v_i \in M_j$ connecting to e_j, we use two binary variables to represent its coordinates (x_{ij}, y_{ij})

$$\begin{aligned} x_{ij} &= x_i + (1 - r_i - r_i')o_{ij}^x + (r_i - r_i')o_{ij}^y \\ y_{ij} &= y_i + (r_i' - r_i)o_{ij}^x + (1 - r_i - r_i')o_{ij}^y. \end{aligned} \quad (24)$$

The binary variables (r_i, r_i') with values $(0, 0)$, $(0, 1)$, $(1, 1)$, and $(1, 0)$ indicate that macro rotates counterclockwise by 0°, 90°, 180°, and 270°, respectively.

Since the rotation assignment is performed after the initial 3-D placement, all the instances are distributed to the corresponding dies according to z-coordinates, and HBTs for the crossing-die nets are inserted at the center point of the optimal region in (15). The problem is reduced to the 2-D scenario. Our objective is to minimize the total D2D wirelength of net set E_M, leading to the following MILP formulation:

$$\begin{aligned} \min \quad & \sum_{e_j \in E_M} \left(R_j^x - L_j^x + R_j^y - L_j^y\right) \\ \text{s.t.} \quad & L_j^x \le x_k + o_{kj}^x \le R_j^x \quad \forall v_k \in S_j \\ & x_i + (1 - r_i - r_i')o_{ij}^x + (r_i - r_i')o_{ij}^y \le R_j^x \quad \forall v_i \in M_j \end{aligned}$$

$$\begin{aligned}
&x_i + \left(1 - r_i - r_i'\right)o_{ij}^x + \left(r_i - r_i'\right)o_{ij}^y \geq L_j^x \quad \forall v_i \in M_j \\
&L_j^y \leq y_k + o_{kj}^y \leq R_j^y \quad \forall v_k \in S_j \\
&y_i + \left(r_i' - r_i\right)o_{ij}^x + \left(1 - r_i - r_i'\right)o_{ij}^y \leq R_j^y \quad \forall v_i \in M_j \\
&y_i + \left(r_i' - r_i\right)o_{ij}^x + \left(1 - r_i - r_i'\right)o_{ij}^y \geq L_j^y \quad \forall v_i \in M_j. \quad (25)
\end{aligned}$$

R_j^x (R_j^y) and L_j^x (L_j^y) represent the x (y) bounding box boundary to optimize. Note that we consider the HBT locations by treating the HBT as standard cell at this stage. There are $O(|V_M|)$ binary variables and $O(|E_M|)$ linear constraints, which are relatively small. We can solve it optimally by invoking an MILP solver with negligible runtime overhead. Additionally, our MILP formulation is sufficiently flexible to accommodate the orientation constraints of the latest technology node by disabling the corresponding orientation variables. The potential instance overlap resulting from macro rotation will be resolved during the global placement at a later stage.

C. Multidie 2-D Global Placement

With the optimized macro rotations, our framework performs global placement again to improve the placement quality. Although the 3-D mixed-size global placement can explore the entire solution space, it has difficulty in finding a good macro placement for the design with excessively large macros. We propose a multidie 2-D global placement formulation removing z-dimension to resolve the issue. The instance partition is determined by the initial 3-D placement solution.

We model the top die, bottom die, and bonding terminal layer as independent 2-D electrostatic fields [23] so that the partitioned macros can spread more easily without the influence of the macro density obstacle on the other die. The objective of multidie 2-D global placement is to minimize the D2D wirelength while the instances and HBTs on the three layers have minimal overlap, shown as follows:

$$\min_{\boldsymbol{x},\boldsymbol{y}} \sum_{e \in E} \hat{W}_e(\boldsymbol{x}, \boldsymbol{y}) + \langle \boldsymbol{\lambda}, \hat{\boldsymbol{U}} \rangle \quad (26)$$

where $\boldsymbol{\lambda} = (\lambda^+, \lambda^-, \lambda')$ is the vector of the density weights and $\hat{\boldsymbol{U}} = (\hat{U}^+, \hat{U}^-, \hat{U}')$ is the vector of the density penalty for the top die, bottom die, and HBT layer, respectively. We insert dummy fillers into the placement system. The total area of fillers is determined by subtracting the total instance area from the die area. The independent 2-D density models give more flexibility for the macros compared to the 3-D density model, and the HBTs, connecting the top partial nets and the bottom partial nets, guide the connected instances to align in an F2F manner during the placement.

D. Legalization

Die-by-die legalization is performed for macros, standard cells, and HBTs to remove the overlap. We utilize the transitive closure graph (TCG) [28], [29] to represent the relation between macros, and the dual problem of TCG-based macro legalization is associated with the min-cost flow problem [30], which can be solved efficiently by the network simplex algorithm. We legalize the standard cells die-by-die with Tetris [31] and Abacus [32] algorithms. The HBTs share the same square size $w' \times w'$ and require a minimum spacing s' between each other. Hence, we pad the HBT to a square with size $w' + s'$ and legalize them as ordinary standard cells with row height $w' + s'$. The actual position of the HBTs can be derived from the padded HBTs.

E. Detailed Placement

We adopt ABCDPlace [33] as our detailed placement engine, including strategies of global swap [34], independent set matching [35], and local reordering [36]. The instances and HBTs on each layer are refined sequentially. After one iteration of detailed placement, the optimal regions of HBTs may be changed. Hence, we can map the HBT to the center point of the updated optimal region, followed by a new iteration of HBT legalization and detailed placement. The wirelength improvement is negligible for more iterations of the process. Therefore, we only perform one additional iteration of the detailed placement.

IV. Density and Wirelength Algorithms

A. Adaptive 3-D Density Accumulation

The density accumulation is computation-intensive, becoming the runtime bottleneck in 3-D global placement. Density accumulation includes two phases: 1) the forward phase to compute the density map ρ from instances and 2) the backward phase to accumulate the weights from the electric field maps $\boldsymbol{E}$ to instances. Two phases share the same primitive operation to compute the overlapping region of instances and bins. The computation workload can be very imbalanced for standard cells and macros. Therefore, adaptive algorithms are desired for mixed-size designs in 3-D scenarios.

For an instance v_i with size $w_i \times h_i \times (d_z/2)$, the corresponding cuboid region is $D_{v_i} = [x_i - (w_i/2), x_i + (w_i/2)] \times [y_i - (h_i/2), y_i + (h_i/2)] \times [z_i - (d_z/4), z_i + (d_z/4)]$. The density map ρ has a size of $|B| = N_x \times N_y \times N_z$. For each bin $b \in B$ as a cuboid with size $w_b \times h_b \times d_b$, the density is calculated as

$$\rho_b = \sum_{v_i \in V} \omega_{v_i} \frac{\text{vol}\left(D_{v_i} \cap b\right)}{\text{vol}(b)} \quad (27)$$

where ω_{v_i} is the weight of instance v_i and $\text{vol}(\cdot)$ is the volume of the cuboid region. We implement the local smoothness technique as described in [23], and ω_{v_i} is determined by the relative sizes of the instance and the bin, $\omega_{v_i} = \min\{1, (w_i/\sqrt{2}w_b)\} \times \min\{1, (h_i/\sqrt{2}h_b)\}$. Given that all instances have the same depth, we do not apply local smoothness to the depth dimension. The approach used in prior work [12], [21] for calculating (26) is to allocate one thread for each instance and sequentially update all the overlapped bins within that thread. However, large macros in 3-D scenarios may cover many bins, causing severe load balancing issues.

A natural idea for solving the problem is to exploit different levels of parallelism for standard cells and macros. Instance parallelism [12] for standard cells is abundant, and the workload for each thread is light and balanced. In contrast to the

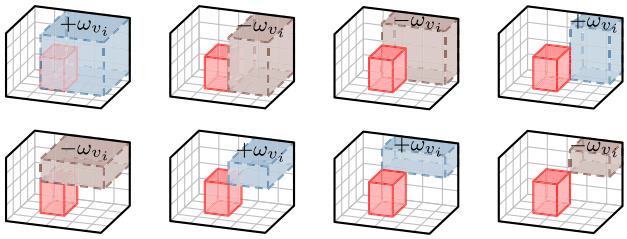


Fig. 6. 3-D density map calculation for macro v_i colored in red is decomposed to the weighted sum of eight corner maps, performed by 3-D prefix sum. Blue submaps indicate addition region and brown submaps indicate subtraction region.

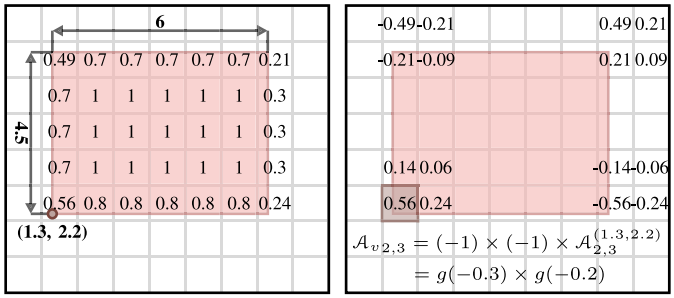


Fig. 7. 2-D density accumulation example using our prefix sum approach. (a) Resulting density map ρ. Entries without a number represent zero. (b) 2-D map $\mathcal{A}_v$ calculated using (28) and (30). Apparently, the prefix sum of $\mathcal{A}_v$ equals the density map ρ. 3-D density accumulation follows a similar rationale.

standard cells, the number of macros is small, but the number of bins to traverse for density calculation is much larger. The key to achieving efficient macro density accumulation is to effectively exploit the bin parallelism.

Guo et al. [37] applied bin parallelism using the prefix sum algorithm in 2-D scenarios. Specifically, they decomposed each macro into four bottom-right instances, with each instance further divided into four subinstances. These subinstances can be processed as increments on bottom-right submatrices or on individual grids. However, their decomposition strategy, tailored for the 2-D density model, is challenging to extend to 3-D scenarios due to its reliance on manual design. In contrast, we propose a general formulation for 3-D density accumulation with a theoretical guarantee of correctness.

The *3-D prefix sum* operator is a function $\varphi : \mathbb{R}^{N_x \times N_y \times N_z} \to \mathbb{R}^{N_x \times N_y \times N_z}$ such that

$$\varphi(\mathcal{A})_{ijk} = \sum_{i'=1}^{i} \sum_{j'=1}^{j} \sum_{k'=1}^{k} \mathcal{A}_{i'j'k'} \tag{28}$$

holds for any 3-D map $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_z}$ and valid index tuple (i, j, k). The prefix sum can propagate a single value in $\mathcal{A}$ to the region with larger indices in time complexity $O(N_x N_y N_z)$. Based on this idea, we can efficiently compute the density map for macros by only considering the 8 corners, as illustrated in Fig. 6.

We first gather the density values at the macro corners. Consider a corner (x, y, z) and its normalized coordinates $(\hat{x}, \hat{y}, \hat{z}) \coloneqq ([x/w_b], [y/h_b], [z/d_b])$. Note that a corner may not align precisely with the 3-D grid of bins. And we introduce the function $g(a) = \max\{1 - |a|, 0\}$ for the partial density introduced by the noninteger coordinates. Let the 3-D map $\mathcal{A}^{(x,y,z)}$ be induced according to the following mechanism:

$$\mathcal{A}^{(x,y,z)}_{ijk} = g\big(i - 1 - \hat{x}\big) g\big(j - 1 - \hat{y}\big) g\big(k - 1 - \hat{z}\big). \tag{29}$$

$\mathcal{A}^{(x,y,z)}$ is sparse with at most 8 nonzero entries adjacent to the bin index $(\lceil \hat{x} \rceil, \lceil \hat{y} \rceil, \lceil \hat{z} \rceil)$. Then, we have the following theorem.

Theorem 1: For macro v_i with size $w_i \times h_i \times (d_z/2)$, center coordinate (x_i, y_i, z_i), and corresponding cuboid region D_{v_i}, consider 3-D map

$$\mathcal{A}_{v_i} = \sum_{\sigma_x, \sigma_y, \sigma_z \in \{-1, 1\}} -\sigma_x \sigma_y \sigma_z \mathcal{A}^{\left(x_i + \sigma_x \frac{w_i}{2}, y_i + \sigma_y \frac{h_i}{2}, z_i + \sigma_z \frac{d_z}{4}\right)}. \tag{30}$$

Then, its prefix sum satisfies $\varphi(\mathcal{A}_{v_i})_b = [\mathrm{vol}(D_{v_i} \cap b)/\mathrm{vol}(b)]$ for each bin $b \in B$.

Theorem 1 demonstrates the way to simplify the 3-D density accumulation for macros. With $\sigma_x, \sigma_y, \sigma_z$ as binary variables, and the maps being sparse with 8 real numbers, the summation in (30) can be finished in constant time. The calculation of density map for macros is extremely fast by adding $\mathcal{A}_{v_i}$ for all macros V_M followed by a single time of 3-D prefix sum. Therefore, the prefix sum density accumulation runs in $O(N_x N_y N_z + |V_M|)$, which is linear in both the number of bins and macros. Fig. 7 illustrates a 2-D density accumulation example using our prefix sum approach. The prefix sum of $\mathcal{A}_{v_i}$, calculated according to (28) and (30), equals the density map ρ. And 3-D density accumulation follows a similar rationale.

In the backward phase, each instance receives the electric force from the overlapped bins in three directions, which requires performing prefix sum on electric field maps $\boldsymbol{E}$. The procedure is similar to the forward density accumulation. The 3-D prefix sum is a one-time cost, and the electric force for each macro can be induced at the 8 corners with constant-time summation operations. Hence, the time complexity for the backward phase is also $O(N_x N_y N_z + |V_M|)$.

Our adaptive method utilizes the instance parallelism for the standard cells and bin parallelism for the macros, which reduces the runtime of 3-D global placement from 400s to 157s on `case4` of the ICCAD 2023 contest benchmarks [25] compared to the approach [12].

B. Incremental Wirelength Gradient Algorithm

The depth gradient $\nabla_z \hat{W}_{e,\mathrm{Bi}}$ in (19) requires different computation mechanism compared to other weighted-average model-based gradients. We perturb the pin partition with $\Delta z = (d_z/4)$ and check the bistratal wirelength change $\Delta W_{e,\mathrm{Bi}}$, involving frequently updating the bounding boxes of the top and bottom nets.

Let $P_e = \{p_1, \ldots, p_l\}$ denote the set of all pins connected by net e. The vanilla approach in [12] evaluates the $W_{e,\mathrm{Bi}}$ after changing the partition of pin $p \in P_e$ by traversing the rest of pins $P_e \setminus \{p\}$ to check the maximum and minimum values. The time complexity is $O(|P_e|^2)$ to finish the computation for net e. However, we find that most computation in the vanilla approach is unnecessary, and the equivalent results can be calculated incrementally.

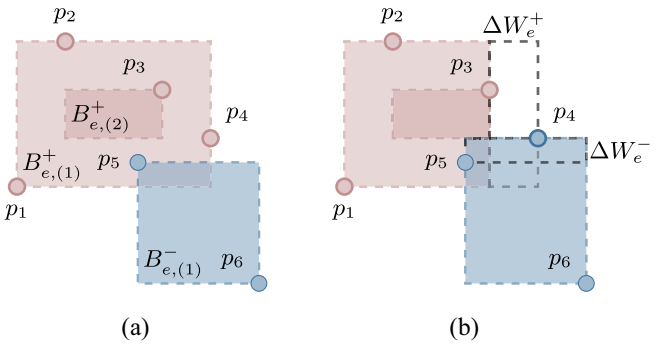


Fig. 8. Illustration of incremental computation for $\nabla_z \hat{W}_{e,\mathrm{Bi}}$ of p_4, other pins share the same procedure. (a) p_4 lies on the current bounding box $B^+_{e,(1)}$, and moving it to the bottom die makes $B^+_{e,(2)}$ become the boundary. (b) Bistratal wirelength change $\Delta W_{e,\mathrm{Bi}} = \Delta W^+_e + \Delta W^-_e$ can be calculated in constant time when moving p_4 to the bottom die.

The key observation is that updates to the coordinates of the top-net bounding box $B^+_{e,(1)}$ with coordinates $\{x^+_{\min}, y^+_{\min}, x^+_{\max}, y^+_{\max}\}$ or the bottom-net bounding box $B^-_{e,(1)}$ with coordinates $\{x^-_{\min}, y^-_{\min}, x^-_{\max}, y^-_{\max}\}$ are necessary only if the pin being moved is located at the boundary of these boxes. Rather than traversing all pins P_e to determine new maximum and minimum values, it is sufficient to refer to the coordinates of the second outermost bounding boxes, $B^+_{e,(2)}$ and $B^-_{e,(2)}$, which represent the second-largest and second-smallest values among the pin coordinates on the top and bottom dies, respectively. A similar approach has been adopted in [38].

If the top pin p is located on $B^+_{e,(1)}$, and is moved to the bottom die, then $B^+_{e,(2)}$ becomes the new bounding box for the top net. If the location of p on the bottom die is outside the bounding box $B^-_{e,(1)}$, we then update $B^-_{e,(1)}$ accordingly. The change of bistratal wirelength $\Delta W_{e,\mathrm{Bi}}$ can be calculated by $\Delta W_{e,\mathrm{Bi}} = W^p_{e,\mathrm{Bi}} - W_{e,\mathrm{Bi}}$, where $W^p_{e,\mathrm{Bi}}$ represents the bistratal wirelength after changing the partition of pin p and $W_{e,\mathrm{Bi}}$ represents the initial bistratal wirelength. The calculation costs constant time for each pin. Hence, the incremental algorithm computes the depth gradient $\nabla_z \hat{W}_{e,\mathrm{Bi}} = (\Delta W_{e,\mathrm{Bi}}/\Delta z)$ in time complexity $O(|P_e|)$ for net e. Fig. 8 illustrates the incremental computation for one pin, and the gradients for other pins can be calculated similarly. Compared to the vanilla approach in [12], our incremental algorithm exhibits lower-time complexity, making it more efficient.

V. Experimental Results

A. Experimental Setup

We conducted experiments on the ICCAD 2023 contest benchmarks [25] and open-source RISC-V designs. The detailed design statistics of ICCAD 2023 contest benchmark are shown in Table II. Top and bottom maximum utilization rate is 80%, and the HBT cost β is 10 for these designs. The contest evaluates the *raw score* = HPWL + β#HBTs with a runtime factor. Most designs adopt heterogeneous technology nodes with a large macro area ratio r_{MA}, bringing a significant challenge to optimizing the D2D wirelength. Additionally,

TABLE II
Statistics of ICCAD 2023 Contest Benchmarks [25]. RH^+ and RH^- Represent Row Height Values of the Top and Bottom Die, Respectively. w' Stands for the Pitch Size of HBTs. r_{MA} Stands for the Macro Area Ratio, Which Is Calculated Using the Technology Information of the Top Die

Bench.	#Cells	#Macros	#Nets	RH^+	RH^-	w'	r_{MA}
case2	13901	6	19547	33	33	92	0.88
case2h1	13901	6	19547	33	48	92	0.88
case2h2	13901	6	19547	33	48	92	0.88
case3	124231	34	164429	33	48	56	0.71
case3h	124231	34	164429	36	48	58	0.67
case4	740211	32	758860	92	115	54	0.36
case4h	740211	32	758860	55	69	32	0.36

TABLE III
Statistics of RISC-V Designs. The Top Die Uses NanGate 15 nm [39] With $RH^+ = 0.768\mu$m, and the Bottom Die Uses NanGate 45 nm [40] With $\mathrm{RH}^- = 1.4\mu$ m. u^+ and u^- Represent the Maximum Utilization Rate of the Top and Bottom Die, Respectively

Bench.	#Cells	#Macros	#Nets	u^+	u^-	r_{MA}
tinyRocket	24647	2	26085	0.50	0.60	0.07
SweRV	87587	28	91903	0.70	0.80	0.81
Ariane	145684	132	157129	0.80	0.90	0.67
BlackParrot	273187	24	265585	0.55	0.65	0.55

we evaluated our placer on four modern RISC-V designs, including `tinyRocket` [41], `SweRV` [42], `Ariane` [43], and `BlackParrot` [44]. The RTL designs were synthesized using Yosys in OpenROAD project [45]. The heterogeneous F2F stacking is set as NanGate 15 nm [39] on the top die and NanGate 45 nm [40] on the bottom die. The HBT pitch size w' is 1μm, and the HBT cost β is 10 for these designs. The detailed design statistics are shown in Table III.

We implemented the proposed 3-D mixed-size placement framework in `C++` and `CUDA` based on the open-source placer DREAMPlace [21]. And we used `Gurobi` [46] as the MILP solver. We set the z-bin size as $d_b = ([w_b + h_b]/2)$, and the region Ω depth is $d_z = N_z d_b$. We empirically set the HBT penalty factor as $\alpha = \alpha_0(d_x\eta^2/d_z)\log(90\beta\eta - 1)$ where $\alpha_0 = 3.5 \times 10^{-3}$ and $\eta = (2w'/[\mathrm{RH}^+ + \mathrm{RH}^-])$, considering the relationship between number of HBTs and design statistics. All the experiments were performed on a Linux machine with 20-core Intel Xeon Silver 4210R CPU (2.40 GHz), 1 GeForce RTX 3090Ti GPU, and 24-GB RAM. We compared our framework with the SOTA placers from top-3 teams in the ICCAD 2023 contest [25], and the reported results were evaluated using the official evaluator provided by the contest. We obtained the executables from the contest winners and ran them on our machine with eight CPU threads following the contest setting. Our placer was evaluated both on the CPU with eight threads and on the GPU.

B. Comparison With SOTA Placers

Table IV shows the official raw score of top-3 teams and our framework on the contest benchmarks. We also compared the detailed score decomposition, including D2D HPWL and

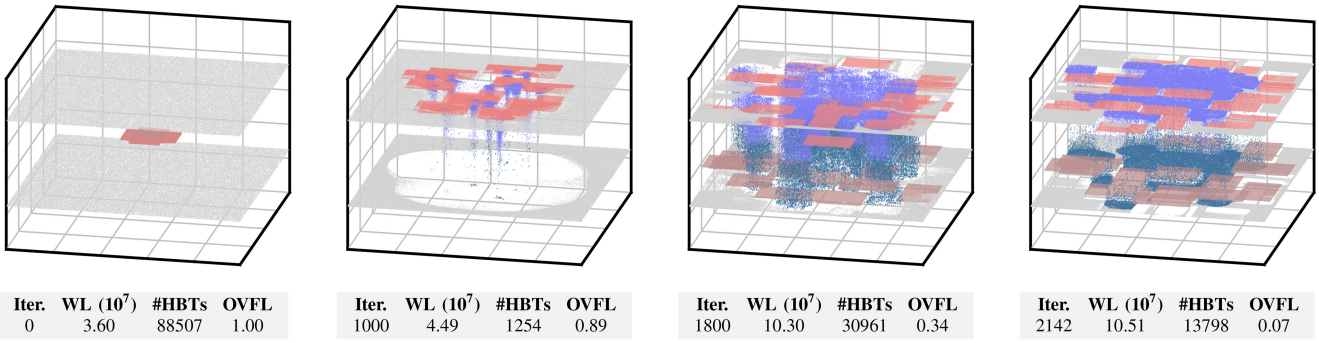


Fig. 9. 3-D mixed-size global placement process on `case3h` with heterogeneous technology nodes. Macros and standard cells spread at the same speed at the early global placement stage, leading to an optimized macro partitioning subsequently. The convergent placement solution with overflow 0.07 finds a clear instance partitioning.

TABLE IV
Official RAW SCORE COMPARISON WITH TOP-3 WINNERS PROVIDED BY ICCAD 2023 CONTEST. THE RAW SCORE = HPWL + β#HBTS. β IS 10 FOR ALL THE CASES

Bench.	1st Place	2nd Place	3rd Place	Ours
`case2`	16506066	16287082	16559126	**15635352**
`case2h1`	18123044	19055977	21180946	**16569703**
`case2h2`	18124483	19202109	21664974	**16820960**
`case3`	98928220	105647967	116317085	**98206238**
`case3h`	122459408	120820762	117889705	**108166770**
`case4`	1047716115	1110850494	1131599485	**1037676163**
`case4h`	656528147	682231267	703663946	**635259476**
Average	1.059	1.096	1.157	**1.000**

HBT number, and reported the runtime of each case with the baselines in Table V. Our analytical 3-D placement framework consistently obtained the best results for all the cases, as shown in Table IV, demonstrating the significant advantage of our 3-D placement paradigm with the dedicated density model and bistratal wirelength model. Compared to the top-3 teams, our placer achieved 5.9%, 9.6%, and 15.7% better score on average, respectively. The score is dominated by the wirelength due to the small HBT cost. Our placer obtained better wirelength than the baselines on all the cases, using a similar number of HBTs, as shown in Table V. When running on a CPU, our placer shows similar runtime with the first place and achieves 2.1× speedup over the third place. And our framework demonstrates better scalability, achieving up to 2.3× and 2.6× speedup over the first and third places, respectively, on the large cases. While the second place is 2.9× faster than our placer, it is limited to 2-D placement engine and yields lower-quality results. Additionally, our proposed algorithms are suitable for GPU acceleration. Leveraging the adaptive 3-D density accumulation and incremental wirelength gradient algorithms, our GPU-accelerated placer shows significant runtime improvements compared to the baselines. Specifically, it achieves 4.0× and 7.8× speedup over the first and third places, respectively, and up to 1.8× speedup over the second place on the large cases.

We also evaluated our framework on four modern RISC-V designs, with experimental results shown in Table VI. The second and third places obtained very low-quality results and failed to generate legal solutions for more than two designs, so their results have been omitted. The first place failed for `BlackParrot` because of diverged 3-D global placement and subsequent macro legalization error. In contrast, our placer obtained legal placement solution across all tested designs. Compared to the first place, our placer not only reduced wirelength by a significant 20% but also achieved a 4.1× speedup on CPU and a remarkable 12.0× speedup on GPU.

C. 3-D Mixed-Size Placement Analysis

The 3-D mixed-size global placement plays a dominant role in our framework, which optimizes the D2D wirelength while explicitly considering instance partitioning, visualized in Fig. 9. Fillers, standard cells on the top die, and standard cells on the bottom die are denoted by gray, purple, and blue rectangles. The macros are colored in red on the top die and colored in brown on the bottom die. The instance depth is omitted for clear visualization. All the standard cells and macros are randomly initialized around the center of the region from a normal distribution. During the global placement, the bistratal wirelength model effectively optimizes instance locations in the 3-D solution space. The proposed 3-D preconditioner allows macros and standard cells to spread at the same speed, leading to an optimized macro partitioning at a later stage. The customized density model finally drives all the instances to exactly two dies.

D. Acceleration of Density and Wirelength Algorithms

We further investigate the efficiency of our proposed density and wirelength algorithms. Fig. 10(a) compares our adaptive 3-D density accumulation with the instance-parallel approach [12]. Our approach achieves 4.7× speedup by exploiting the abundant bin parallelism for macros, avoiding the load balancing issue.

Fig. 10(b) compares our incremental wirelength gradient algorithm with the vanilla approach in [12]. Our algorithm reduces the time complexity from $O(|P_e|^2)$ to $O(|P_e|)$ for calculating the z-gradient, resulting in 1.7× speedup.

E. Ablation Study on 3-D Mixed-Size Preconditioning

The 3-D mixed-size placement is a highly nonlinear, nonconvex, and ill-conditioned problem. The heterogeneous

TABLE V
SCORE DECOMPOSITION COMPARED TO THE TOP-3 WINNERS ON THE ICCAD 2023 CONTEST BENCHMARK. *RT*(S) STANDS FOR THE TOTAL RUNTIME. ALL THE BASELINES ARE EVALUATED ON OUR MACHINE WITH EIGHT CPU THREADS. OUR PLACER IS EVALUATED BOTH ON THE CPU WITH EIGHT THREADS AND ON THE GPU

Bench.	1st Place			2nd Place			3rd Place			Ours-CPU			Ours-GPU		
	HPWL	#HBTs	RT	HPWL	#HBTs	RT	HPWL	#HBTs	RT	HPWL	#HBTs	RT	HPWL	#HBTs	RT
case2	16490836	1523	67	16277152	993	32	16537596	2153	144	**15528810**	1128	76	15622062	1329	38
case2h1	18121844	120	39	19047767	821	43	21156596	2435	160	16708363	1135	80	**16556213**	1349	35
case2h2	18123283	120	42	19193899	821	41	21640714	2426	162	**16748148**	1091	80	16807840	1312	36
case3	98706330	22189	534	105386847	26112	104	116022515	29457	602	**97281904**	10704	236	98081778	12446	92
case3h	122271798	18761	262	120770382	5038	104	117633295	25641	612	109386719	13224	239	**108028790**	13798	86
case4	1046106185	160993	3605	1108969124	188137	615	1130211865	138762	5309	1040202500	132109	3070	**1036364973**	131119	335
case4h	654962287	156586	1567	680554407	167686	592	702244786	141916	4492	634654510	133734	2640	**633920946**	133853	361
Average	1.058	0.981	4.000	1.096	1.019	1.289	1.156	1.660	7.830	1.000	0.907	4.047	**1.000**	1.000	1.000

TABLE VI
EXPERIMENTAL RESULTS ON MODERN RISC-V DESIGNS. HPWL IS MEASURED IN μM. THE BASELINE IS EVALUATED ON OUR MACHINE WITH EIGHT CPU THREADS. OUR PLACER IS EVALUATED BOTH ON THE CPU WITH EIGHT THREADS AND ON THE GPU

Bench.	1st Place				Ours-CPU				Ours-GPU			
	Score	HPWL	#HBTs	RT	Score	HPWL	#HBTs	RT	Score	HPWL	#HBTs	RT
tinyRocket	206290	145830	6046	356	182559	130749	5181	104	**181571**	**130631**	5094	42
SweRV	1020414	981634	3878	2208	924932	805952	11898	341	**922004**	**800464**	12154	108
Ariane	1828336	1477246	35109	1009	1328717	1184607	14411	397	**1311346**	**1174076**	13727	142
BlackParrot	NA	NA	NA	NA	**1586522**	**1488882**	9764	640	1613662	1515872	9779	156
Average	1.212	1.200	1.355	12.009	1.001	1.000	1.011	3.133	**1.000**	**1.000**	1.000	1.000

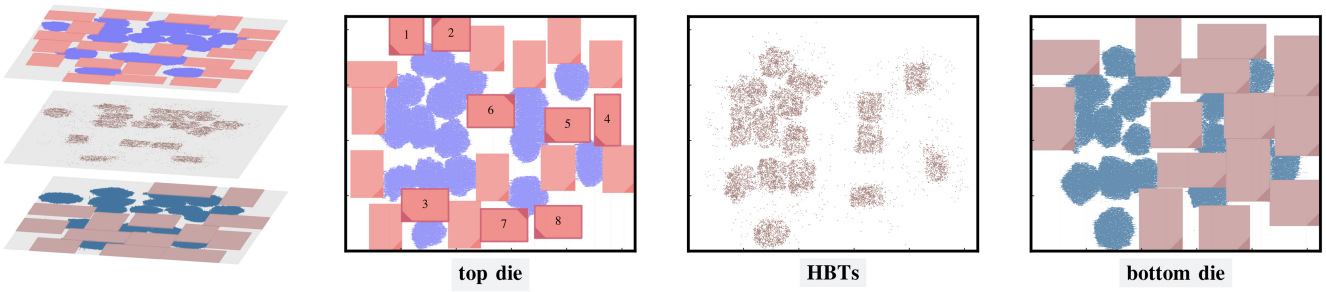


Fig. 10. Final layout of case3h after the detailed placement. Our MILP finds the optimized macro rotations, consequently improving wirelength. The eight rotated macros are marked with numbers. The HBTs are sparsely placed to connect the instances on different dies.

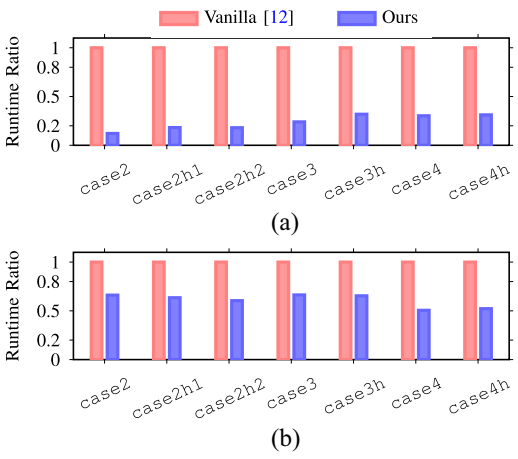


Fig. 11. Runtime comparison of (a) density and (b) wirelength algorithms between vanilla approach in [12] and ours on GPU.

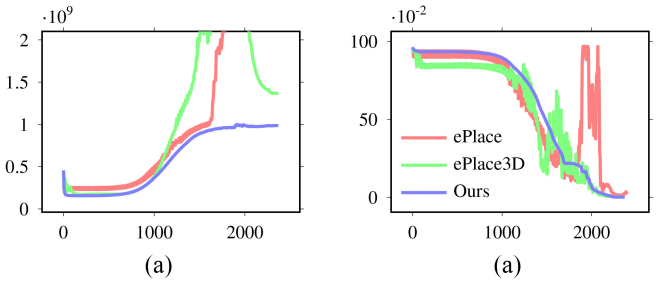


Fig. 12. Wirelength and macro density overflow curves over global placement iterations of different preconditioners on case4. The macro density overflow is calculated by the macro density map and the target density. (a) Wirelength curve. (b) Macro density overflow.

scenarios make the problem even more complex. The preconditioner should handle the large topological and physical difference between macros and standard cells. Replacing our proposed preconditioner with previous approaches adopted in ePlace [23], [24] and ePlace3-D [19], the 3-D global placement will diverge or obtain very low-quality results with wirelength increased by 30% on the ICCAD 2023 contest benchmarks [25]. Fig. 12 shows the effect of our 3-D mixed-size preconditioner on case4. And the trend for other designs is similar. The previous approaches [19], [23], [24] fail to stabilize the optimization of macro locations for the whole process, causing macro density overflow oscillation and

TABLE VII
RAW SCORE AND RUNTIME RESULTS OF OUR APPROACH WITHOUT AND WITH MILP MACRO ROTATION. #ROT STANDS FOR THE NUMBER OF ROTATED MACROS

Bench.	w/o. Rotation		w. Rotation		
	Score	RT	Score	RT	#Rot
case2	15635352	38	15635352	38	0
case2h1	16569703	35	16569703	35	0
case2h2	16820960	36	16820960	36	0
case3	100227409	91	98206238	92	6
case3h	111062583	88	108166770	86	8
case4	1058535164	336	1037676163	335	8
case4h	645574820	346	635259476	361	8
Average	1.012	0.996	1.000	1.000	-

TABLE VIII
RAW SCORE AND RUNTIME RESULTS FOR MULTIDIE 2-D GLOBAL PLACEMENT FLOW AND 3-D MIXED-SIZE GLOBAL PLACEMENT FLOW. r_{MA} STANDS FOR THE MACRO AREA RATIO

Bench.	r_{MA}	multi-die 2D		3D mixed-size	
		Score	RT	Score	RT
case2	0.88	**15635352**	38	17081257	33
case2h1	0.88	**16569703**	35	17413812	36
case2h2	0.88	**16820960**	36	17701896	37
case3	0.71	**98206238**	92	101230278	118
case3h	0.67	**108166770**	86	112539329	100
case4	0.36	1064731451	299	**1037676163**	335
case4h	0.36	662128786	300	**635259476**	361
Average	-	0.974	0.923	1.000	1.000

wirelength divergence. In contrast, our preconditioner makes the standard cells and macros equalized in the optimizer's perspective, enabling stable optimization.

F. Ablation Study on MILP Macro Rotation

Our MILP utilizes the physical information of the initial 3-D placement solution, finding the macro rotations with optimal wirelength. The effect of MILP macro rotation is shown in Table VII. Except for the cases with few extremely large macros, our MILP finds the macro rotations leading to better wirelength, achieving on average 1.2% wirelength improvement. Since we only consider the nets connecting to macros, the runtime overhead is negligible, and it takes less than 1s for all the cases. Compared to other cases, we observe a larger difference of global placement iterations for `case4h`, leading to a larger runtime difference. The final placement solution for `case3h` with macro rotation is illustrated in Fig. 10.

G. Ablation Study on Placement Flow

Our framework adopts multidie 2-D global placement for designs with a macro area ratio exceeding 50%, while 3-D mixed-size global placement is utilized for other designs. Table VIII presents the experimental results comparing these two placement flows. We observe that multidie 2-D global placement achieves 5.3% better score than 3-D mixed-size global placement for designs with a large macro footprint. This improvement is attributed to the removal of macro density obstacle in the z-direction, which results in better-macro placement results. Conversely, 3-D mixed-size global placement excels in optimizing standard cell locations, obtaining 3.4% better score compared to its counterpart.

VI. CONCLUSION

This article proposed a new analytical 3-D mixed-size placement framework with full-scale GPU acceleration, leveraging dedicated density and wirelength algorithms, for heterogeneous F2F bonded 3-D ICs. Our customized density model and bistratal wirelength model, incorporating a novel 3-D preconditioner, enable stable optimization for macros and standard cells in a 3-D solution space. We further propose an MILP formulation for macro rotation to optimize the wirelength. Experimental results on ICCAD 2023 contest benchmarks demonstrate that our framework significantly surpasses the first-place winner by 5.9% on the quality of results with 4.0× runtime speedup. Additional experiments on modern RISC-V designs further validate the generalizability and superior performance of our framework.

APPENDIX

We use notation $\prod_{\text{cyc}}$ to represent multiplication over all three dimensions, e.g., $\prod_{\text{cyc}} f(x) = f(x)f(y)f(z)$ for any well-defined function f. Function $\mu(\cdot)$ is a measure defined for any measurable regions. In our 3-D regions, $\mu(\cdot)$ stands for the volume estimator $\mu(\cdot)$. To prove Theorem 1, we first present a lemma.

Lemma 1: Denote $D^{(x,y,z)} = [x, d_x] \times [y, d_y] \times [z, d_z]$ for any $(x, y, z) \in \Omega$. Then $\varphi(\mathcal{A}^{(x,y,z)})_b = [\mu(D^{(x,y,z)} \cap b)/\mu(b)]$ holds for any bin $b \in B$.

Proof: The total number of bins is $|B| = N_x N_y N_z$. Consider the normalized coordinate $(\hat{x}, \hat{y}, \hat{z}) = ([x/w_b], [y/h_b], [z/d_b])$ and arbitrary bin b with index (i, j, k). Clearly, we have $b = b_x \times b_y \times b_z$ where $b_x = [(i-1)w_b, iw_b]$, $b_y = [(j-1)h_b, jh_b]$ and $b_z = [(k-1)d_b, kd_b]$. Therefore, it must be true that

$$\frac{\mu\big(D^{(x,y,z)} \cap b\big)}{\mu(b)} = \frac{1}{w_b h_b d_b} \prod_{\text{cyc}} \mu([x, d_x] \cap b_x). \tag{31}$$

Consider the x dimension only as the other two dimensions are symmetric. If $i < \lceil \hat{x} \rceil$, we have $i < \hat{x}$ and then $iw_b < x$, which means $\mu([x, d_x] \cap b_x) = 0$. If $i > \lceil \hat{x} \rceil$, we have $i - 1 \geq \hat{x}$ and then $(i-1)w_b \geq x$, which means $\mu([x, d_x] \cap b_x) = \mu(b_x) = w_b$. If $i = \lceil \hat{x} \rceil$, we have $x \in b_x$ and then $\mu([x, d_x] \cap b_x) = iw_b - x$. Hence, we can summarize that

$$\frac{\mu([x, d_x] \cap b_x)}{w_b} = \begin{cases} 0, & \text{if } i < \lceil \hat{x} \rceil \\ \lceil \hat{x} \rceil - \hat{x}, & \text{if } i = \lceil \hat{x} \rceil \\ 1, & \text{elsewhere.} \end{cases} \tag{32}$$

On the other hand, consider function $g(\cdot)$ in (28). It is clear that we have

$$g\big(i - \hat{x}\big) = \begin{cases} \lceil \hat{x} \rceil - \hat{x}, & \text{if } i = \lceil \hat{x} - 1 \rceil \\ \hat{x} + 1 - \lceil \hat{x} \rceil, & \text{if } i = \lceil \hat{x} \rceil \\ 0, & \text{elsewhere} \end{cases} \tag{33}$$

for any integer i. Apply the 1-D prefix sum on this function, then it is straightforward to see

$$\sum_{i'=0}^{i-1} g\big(i'-\hat{x}\big) = \frac{\mu([x,d_x]\cap b_x)}{w_b} \tag{34}$$

according to (32). Now, consider the prefix sum of $\mathcal{A}^{(x,y,z)}$, defined by $\mathcal{P}=\varphi(\mathcal{A}^{(x,y,z)})$. Combining (27)–(29), we have

$$\begin{aligned}\mathcal{P}_{ijk} &= \sum_{i'=0}^{i-1}\sum_{j'=0}^{j-1}\sum_{k'=0}^{k-1} g\big(i'-\hat{x}\big)g\big(j'-\hat{y}\big)g\big(k'-\hat{z}\big)\\ &= \frac{1}{w_b h_b d_b}\prod_{\text{cyc}}\mu([x,d_x]\cap b_x) = \frac{\mu\big(D^{(x,y,z)}\cap b\big)}{\mu(b)}\end{aligned} \tag{35}$$

and therefore we obtain $\varphi(\mathcal{A}^{(x,y,z)})_b = [\mu(D^{(x,y,z)}\cap b)/\ \mu(b)]$. ■

Now we are going to complete the proof to Theorem 1 with the help of Lemma 1.

Proof: Denote $\mu_b(\Omega) = [\mu(\Omega\cap b)/\mu(b)]$ for any measurable region Ω. According to the *inclusion-exclusion principle*, we have the following relationship:

$$\mu_b(D_v) = \sum_{\sigma_x,\sigma_y,\sigma_z} -\sigma_x\sigma_y\sigma_z\mu_b\left(D^{\left(x+\sigma_x\frac{w}{2},y+\sigma_y\frac{h}{2},z+\sigma_z\frac{d_z}{4}\right)}\right) \tag{36}$$

where variables $\sigma_x,\sigma_y,\sigma_z$ are taken over $\{-1,1\}$. The prefix sum operator φ is linear. Hence, we have

$$\begin{aligned}\varphi(\mathcal{A}_v)_b &= \sum_{\sigma_x,\sigma_y,\sigma_z} -\sigma_x\sigma_y\sigma_z\varphi\left(\mathcal{A}^{\left(x+\sigma_x\frac{w}{2},y+\sigma_y\frac{h}{2},z+\sigma_z\frac{d_z}{4}\right)}\right)_b\\ &\overset{(*)}{=} \sum_{\sigma_x,\sigma_y,\sigma_z} -\sigma_x\sigma_y\sigma_z\mu_b\left(D^{\left(x+\sigma_x\frac{w}{2},y+\sigma_y\frac{h}{2},z+\sigma_z\frac{d_z}{4}\right)}\right)\\ &= \mu_b(D_v) = \frac{\mu(D_v\cap b)}{\mu(b)}\end{aligned} \tag{37}$$

where the equation marked with symbol $(*)$ holds according to Lemma 1. The proof is completed. ■

REFERENCES

[1] G. Murali and S. K. Lim, "Heterogeneous 3-D ICs: Current status and future directions for physical design technologies," in *Proc. DATE*, 2021, pp. 146–151.

[2] W. Gomes, S. Morgan, B. Phelps, T. Wilson, and E. Hallnor, "Meteor lake and arrow lake Intel next-gen 3-D client architecture platform with Foveros," in *Proc. IEEE Hot Chips 34th Symp. (HCS)*, 2022, pp. 1–40.

[3] X. Dong, J. Zhao, and Y. Xie, "Fabrication cost analysis and cost-aware design space exploration for 3-D ICs," *IEEE TCAD*, vol. 29, no. 12, pp. 1959–1972, Dec. 2010.

[4] S. K. Samal, D. Nayak, M. Ichihashi, S. Banna, and S. K. Lim, "Monolithic 3-D IC vs. TSV-based 3-D IC in 14-nm FinFET technology," in *Proc. IEEE SOI-3-D-Subthreshold Microelectron. Technol. Unified Conf. (S3S)*, 2016, pp. 1–2.

[5] S. S. K. Pentapati, K. Chang, V. Gerousis, R. Sengupta, and S. K. Lim, "Pin-3-D: A physical synthesis and post-layout optimization flow for heterogeneous monolithic 3-D ICs," in *Proc. ICCAD*, 2020, pp. 1–9.

[6] T. Song, A. Nieuwoudt, Y. S. Yu, and S. K. Lim, "Coupling capacitance in face-to-face (F2F) bonded 3-D ICs: Trends and implications," in *Proc. ECTC*, 2015, pp. 529–536.

[7] M. Jung, T. Song, Y. Wan, Y. Peng, and S. K. Lim, "On enhancing power benefits in 3-D ICs: Block folding and bonding styles perspective," in *Proc. DAC*, 2014, pp. 1–6.

[8] B. W. Ku, K. Chang, and S. K. Lim, "Compact-2-D: A physical design methodology to build commercial-quality face-to-face-bonded 3-D ICs," in *Proc. ISPD*, 2018, pp. 90–97.

[9] L. Bamberg, A. García-Ortiz, L. Zhu, S. Pentapati, D. E. Shim, and S. K. Lim, "Macro-3-D: A physical design methodology for face-to-face-stacked heterogeneous 3-D ICs," in *Proc. DATE*, 2020, pp. 37–42.

[10] X. Zhao et al., "iPL-3-D: A novel bilevel programming model for die-to-die placement," in *Proc. ICCAD*, 2023, pp. 1–9.

[11] Y.-J. Chen, Y.-S. Chen, W.-C. Tseng, C.-Y. Chiang, Y.-H. Lo, and Y.-W. Chang, "Late breaking results: Analytical placement for 3-D ICs with multiple manufacturing technologies," in *Proc. DAC*, 2023, pp. 1–2.

[12] P. Liao, Y. Zhao, D. Guo, Y. Lin, and B. Yu, "Analytical die-to-die 3-D placement with bistratal wirelength model and GPU acceleration," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 43, no. 6, pp. 1624–1637, Jun. 2024.

[13] K.-S. Hu, I.-J. Lin, Y.-H. Huang, H.-Y. Chi, Y.-H. Wu, and C.-F. C. Shen, "2022 ICCAD CAD contest problem B: 3-D placement with D2D vertical connections," in *Proc. ICCAD*, 2022, pp. 1–5.

[14] K. Chang et al., "Cascade2-D: A design-aware partitioning approach to monolithic 3-D IC with 2-D commercial tools," in *Proc. ICCAD*, 2016, pp. 1–8.

[15] P. Vanna-Iampikul, C. Shao, Y.-C. Lu, S. Pentapati, and S. K. Lim, "Snap-3-D: A constrained placement-driven physical design methodology for face-to-face-bonded 3-D ICs," in *Proc. ISPD*, 2021, pp. 39–46.

[16] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "Shrunk-2-D: A physical design methodology to build commercial-quality monolithic 3-D ICs," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 36, no. 10, pp. 1716–1724, Oct. 2017.

[17] G. Luo, Y. Shi, and J. Cong, "An analytical placement framework for 3-D ICs and its extension on thermal awareness," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 4, pp. 510–523, Apr. 2013.

[18] M.-K. Hsu, V. Balabanov, and Y.-W. Chang, "TSV-aware analytical placement for 3-D IC designs based on a novel weighted-average wirelength model," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 4, pp. 497–509, Apr. 2013.

[19] J. Lu, H. Zhuang, I. Kang, P. Chen, and C.-K. Cheng, "ePlace-3-D: Electrostatics based placement for 3-D-ICs," in *Proc. ISPD*, 2016, pp. 11–18.

[20] C. M. Fiduccia and R. M. Mattheyses, "A linear-time heuristic for improving network partitions," in *Proc. DAC*, 1982, pp. 175–181.

[21] Y. Lin, S. Dhar, W. Li, H. Ren, B. Khailany, and D. Z. Pan, "DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement," in *Proc. DAC*, 2019, pp. 1–6.

[22] L. Liu, B. Fu, M. D. Wong, and E. F. Young, "Xplace: An extremely fast and extensible global placement framework," in *Proc. DAC*, 2022, pp. 1309–1314.

[23] J. Lu et al., "ePlace: Electrostatics-based placement using fast fourier transform and Nesterov's method," *ACM Trans. Design Autom. Electron. Syst.*, vol. 20, no. 2, pp. 1–34, 2015.

[24] J. Lu et al., "ePlace-MS: Electrostatics-based placement for mixed-size circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 5, pp. 685–698, May 2015.

[25] K.-S. Hu, H.-Y. Chi, I.-J. Lin, Y.-H. Wu, W.-H. Chen, and Y.-T. Hsieh, "2023 ICCAD CAD contest problem B: 3-D placement with macros," in *Proc. ICCAD*, 2023, pp. 1–6.

[26] Y.-J. Chen, C.-H. Hsieh, P.-H. Su, S.-H. Chen, and Y.-W. Chang, "Mixed-size 3-D analytical placement with heterogeneous technology nodes," in *Proc. DAC*, 2024, pp. 1–6.

[27] L. M. Milne-Thomson, *The Calculus of Finite Differences*. Providence, RI, USA: Am. Math. Soc., 2000.

[28] J.-M. Lin and Y.-W. Chang, "TCG: A transitive closure graph-based representation for non-slicing floorplans," in *Proc. DAC*, 2001, pp. 764–769.

[29] H.-C. Chen, Y.-L. Chuang, Y.-W. Chang, and Y.-C. Chang, "Constraint graph-based macro placement for modern mixed-size circuit designs," in *Proc. ICCAD*, 2008, pp. 218–223.

[30] Y. Lin et al., "MrDP: Multiple-row detailed placement of heterogeneous-sized cells for advanced nodes," *IEEE TCAD*, vol. 37, no. 6, pp. 1237–1250, Jun. 2018.

[31] D. Hill, "Method and system for high speed detailed placement of cells within an integrated circuit design," U.S. Patent 6 370 673, Sep. 2002.

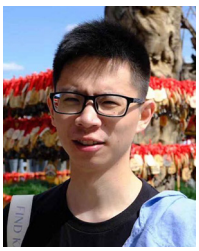
[32] P. Spindler, U. Schlichtmann, and F. M. Johannes, "Abacus: Fast legalization of standard cell circuits with minimal movement," in *Proc. ISPD*, 2008, pp. 47–53.

[33] Y. Lin, W. Li, J. Gu, H. Ren, B. Khailany, and D. Z. Pan, "ABCDPlace: Accelerated batch-based concurrent detailed placement on multithreaded CPUs and GPUs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 12, pp. 5083–5096, Dec. 2020.

[34] S. Popovych, H.-H. Lai, C.-M. Wang, Y.-L. Li, W.-H. Liu, and T.-C. Wang, "Density-aware detailed placement with instant legalization," in *Proc. DAC*, 2014, pp. 1–6.
[35] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang, "NTUplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 7, pp. 1228–1240, Jul. 2008.
[36] M. Pan, N. Viswanathan, and C. Chu, "An efficient and effective detailed placement algorithm," in *Proc. ICCAD*, 2005, pp. 48–55.
[37] Z. Guo, J. Mai, and Y. Lin, "Ultrafast CPU/GPU kernels for density accumulation in placement," in *Proc. DAC*, 2021, pp. 1123–1128.
[38] B. Fu, L. Liu, Y. Sun, W.-H. Lau, M. D. Wong, and E. F. Young, "CoPlace: Coherent placement engine with layout-aware partitioning for 3-D ICs," in *Proc. ASPDAC*, 2024, pp. 65–70.
[39] M. Martins et al., "Open cell library in 15-nm FreePDK technology," in *Proc. ISPD*, 2015, pp. 171–178.
[40] "NanGate 45 nm." Accessed: Sep. 10, 2023. [Online]. Available: https://si2.org/open-cell-library/
[41] "Rocket chip." 2022. [Online]. Available: https://github.com/chipsalliance/rocket-chip
[42] "SweRV." 2020. [Online]. Available: https://github.com/westerndigitalcorporation/swerv_eh1
[43] "Ariane RISC-V CPU." 2024. [Online]. Available: https://github.com/openhwgroup/cva6
[44] "BlackParrot." 2020. [Online]. Available: https://github.com/black-parrot/black-parrot
[45] T. Ajayi et al., "Toward an open-source digital flow: First learnings from the OpenROAD project," in *Proc. DAC*, 2019, pp. 1–4.
[46] "Gurobi Optimizer reference manual." Gurobi. 2023. [Online]. Available: https://www.gurobi.com/

Yuxuan Zhao received the B.S. degree in information engineering from Zhejiang University, Hangzhou, China, in 2021. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

His current research interests include high performance computing in physical design and machine learning in EDA.



Peiyu Liao received the B.S. degree from the School of Mathematical Sciences, Zhejiang University, Hangzhou, China, in 2017, and the M.S. degree from the School of Engineering, The Hong Kong University of Science and Technology, Hong Kong, in 2019. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

His current research interests include high performance computing and numerical optimization in physical design.

Siting Liu received the B.S. degree from the Department of Computer Science, Huazhong University of Science and Technology, Wuhan, China, in 2020. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

She is a visiting student with the School of Integrated Circuits, Peking University, Beijing, China. Her current research interests include deep learning applications and GPU acceleration in physical design.

Ms. Liu received the Best Paper Award from DATE 2022 and the Best Paper Award Nomination from DATE 2021.

Jiaxi Jiang received the B.E. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2023. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

His current research interests include physical design and machine learning applications in EDA.

Yibo Lin (Member, IEEE) received the B.S. degree in microelectronics from Shanghai Jiaotong University, Shanghai, China, in 2013, and the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, Austin, TX, USA, in 2018, advised by Prof. David Z. Pan.

He was a Postdoctoral Researcher with the University of Texas at Austin from 2018 to 2019. He currently is an Assistant Professor with the School of Integrated Circuits, Peking University, Beijing, China. His research interests include physical design, machine learning applications, and heterogeneous computing in VLSI CAD.

Dr. Lin is a recipient of the Best Paper Awards at Premier EDA/CAD journals/conferences like IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, DAC, DATE, and ISPD.

Bei Yu (Senior Member, IEEE) received the Ph.D. degree from The University of Texas at Austin, Austin, TX, USA, in 2014.

Dr. Yu received ten Best Paper Awards from IEEE TSM 2022, DATE 2022, ICCAD 2021 and 2013, ASPDAC 2021 and 2012, ICTAI 2019, Integration, The VLSI Journal in 2018, ISPD 2017, SPIE Advanced Lithography Conference 2016, and six ICCAD/ISPD contest awards. He is currently an Associate Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. He has served as a TPC Chair for ACM/IEEE Workshop on Machine Learning for CAD, and in many journal editorial boards and conference committees. He is an Editor of IEEE TCCPS Newsletter.