Analytical Die-to-Die 3-D Placement With Bistratal Wirelength Model and GPU Acceleration

Peiyu Liao, Yuxuan Zhao, Dawei Guo, Yibo Lin, *Member, IEEE*, and Bei Yu, *Senior Member, IEEE*

***Abstract*—In this article, we present a new analytical 3-D placement framework with a bistratal wirelength model for face-to-face-bonded 3-D ICs with heterogeneous technology nodes based on the electrostatic-based density model. The proposed framework, enabling GPU acceleration, is capable of efficiently determining node partitioning and locations simultaneously, leveraging the dedicated 3-D wirelength model and density model. The experimental results on ICCAD 2022 contest benchmarks demonstrate that our proposed 3-D placement framework can achieve up to 6.1% wirelength improvement and 4.1% on average compared to the first-place winner with much fewer vertical interconnections and up to 9.8× runtime speedup. Notably, the proposed framework also outperforms the state-of-the-art 3-D analytical placer by up to 3.3% wirelength improvement and 2.1% on average with up to 8.8× acceleration on large cases using GPUs.**

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

I. Introduction

With technology scaling nearing its physical limits, the 3-D integrated circuit (3D-IC) has emerged as a promising solution for extending Moore's Law. Vertically stacking multiple dies enables 3D-IC to achieve higher-transistor density and replace long 2-D interconnects with shorter interdie connections, leading to improved circuit performance. Leveraging advanced packaging technology, chiplets with heterogeneous technology nodes can be integrated to achieve leading cost-effective performance. Prominent examples of such technology adoption are Intel's Meteor Lake [1] and AMD's Zen 4 [2], which have resulted in significant performance gains and cost savings.

Conventionally, 3D-ICs are fabricated using through-silicon vias (TSVs) with large pitches and parasitics, which may limit the total number of global interconnects to avoid performance degradation [3]. As an alternative approach, monolithic 3-D (M3D) integration has been proposed, where tiers are fabricated sequentially and connected using monolithic intertier vias (MIVs) [4], [5], [6], [7]. In contrast to TSVs with microscale pitches, MIVs exhibit nanoscale dimensions [5], allowing for higher-integration density with significantly reduced space requirements. Nevertheless, it is still necessary to allocate certain white space on placement regions to accommodate MIVs. Face-to-face (F2F) bonding is another approach that bonds ICs using face sides for both dies [8], [9], [10], [11]. F2F-bonded 3-D ICs do not require additional silicon area for 3-D connections [9], eliminating the need to reserve white space for vias and allowing much higher-integration density. The silicon-space overhead-free property of F2F-bonded 3-D ICs provides significant advantages in numerous applications [7].

The emergence of 3D-IC presents challenges to traditional 2-D electronic design automation methods in producing high-quality 3-D circuit layouts, and the heterogeneous technology nodes further complicates the problem. Placement plays a dominant role on the overall quality of physical design, and innovations of 3D-IC placement are required to fully benefit from the 3-D integration technologies. Within the context of 3-D placement, 3D-IC placers are responsible for solving the optimal 3-D node locations to optimize specific objectives. Such a very large-scale combinatorial optimization problem can be solved in either discrete or analytical algorithms. An analytical 3-D placement algorithm is characterized by employing "true-3D" flows that handle tier partitioning continuously and devise 3-D solutions directly.

Despite the various research achievements mentioned above, existing discrete and analytical 3-D flows are hardly applicable to F2F-bonded 3-D ICs with heterogeneous technology nodes. The discrete solutions typically fail to utilize the advantages of 3-D ICs sufficiently as most of them rely on the FM-mincut tier partitioning [12]. However, the total cutsize is not the primary placement objective in F2F-bonded 3-D ICs due to the silicon-space overhead-free property [9], resulting in suboptimal partitioning for discrete solutions. Conventional analytical 3-D placement algorithms adopt continuous optimization but they do not support heterogeneous technology nodes during global placement. Additionally, previous wirelength-driven analytical placement algorithms use inaccurate wirelength models [13] for numerical optimization, which is inconsistent with F2F-bonded scenarios. Some recent work [14] on wirelength models supports heterogeneous technology nodes in analytical placement. However, it still pays no attention to

Manuscript received 23 July 2023; revised 23 October 2023; accepted 4 December 2023. Date of publication 26 December 2023; date of current version 21 May 2024. This work was supported in part by The Research Grants Council of Hong Kong, SAR, under Project CUHK14208021. This article was recommended by Associate Editor V. Pavlidis. *(Peiyu Liao and Yuxuan Zhao contributed equally to this work.) (Corresponding author: Bei Yu.)*

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

Dawei Guo is with the School of Electronics Engineering and Computer Science, Center for Energy-Efficient Computing and Applications, Peking University, Beijing 100871, China.

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 214031, China.

Digital Object Identifier 10.1109/TCAD.2023.3347293

the wirelength reduction introduced by interdie connections, remaining unsolved inaccurate estimation in 3-D analytical placement.

In this article, we propose a new analytical 3-D placement framework for F2F-bonded 3-D ICs with heterogeneous technology nodes utilizing a novel and precise bistratal wirelength model. Based on the proposed placement framework, we efficiently determine the node locations along with partitioning in a single run. The main contributions are summarized as follows.

1) We design a *bistratal wirelength* model, including computation strategies of the wirelength objective and gradients, that significantly outperforms the widely used models for F2F-bonded 3-D ICs.
2) We propose an ultrafast analytical 3-D placement framework that leverages the *bistratal wirelength* model and eDensity-3D [13] with GPU acceleration, considering heterogeneous technology nodes.
3) Experimental results show that our results achieved the best results on the ICCAD 2022 Contest Benchmarks [15] with up to 6.1% wirelength improvement and 4.1% on average, compared to the first-place winner. Remarkably, we also outperform the state-of-the-art (SOTA) analytical 3-D placer [14] for heterogeneous F2F-bonded 3-D ICs by up to 3.3% wirelength improvement and 2.1% on average. The usage of vertical interconnects are also significantly reduced.

The remainder of this article is structured as follows. Section II provides some preliminaries, including previous works and foundations of analytical placement. Section III discusses the problem statement and problem formulation. Section IV presents the overall flow of the proposed placement framework for heterogeneous F2F-bonded 3-D ICs. Then, Section V depicts the theoretical details of the bistratal wirelength model. Section VI presents experimental results and some related analysis on the adopted benchmarks, followed by the conclusion in Section VII.

II. Preliminaries

A. Related Works

Conventional discrete solutions handle multiple tiers discretely. T3Place [16] transforms 2-D placement solutions into 3-D with several folding techniques and local refinement. Early TSV-based research on partition-based approaches [17], [18], [19], [20] first partitions the netlist to minimize specific targets, e.g., vertical connections, followed by a simultaneous 2-D placement on all tiers. The "pseudo-3D" flows utilize optimization techniques of existing 2-D engines to work with projected 3-D designs. Cascade2-D [21] implements an M3D design using 2-D commercial tools with a design-aware partitioning before placement. Recent partitioning-based approaches [6], [7], [21], [22] suggest that partitioning first may not sufficiently leverage physical information and thus perform partitioning-last strategies after 2-D preplacement. Shrunk-2D [6], [10] is a prominent example that performs partitioning according to a 2-D preplacement. Shrunk-2D requires geometry shrinking of standard cells and related interconnects by 50% during its 2-D preplacement for F2F-bounded 3-D ICs [10] or M3D [6]. Compact-2D [7] adopts placement contraction without geometry shrinking to obtain the 2-D preplacement, followed by a bin-based FM-mincut tier partitioning [12]. Pin-3D [23] proposes pin projection to incorporate interdie physical information by projecting pins to other dies with fixed locations and transparent geometries, which is first applicable to heterogeneous M3D ICs. Snap-3D [24] for F2F bonded 3-D ICs shrinks the height of standard cell layouts by one half and labels footprint rows top versus bottom to indicate partitioning. However, the bin-based min-cut partitioning algorithm lacks an understanding of the impact of partitioning on placement quality. TP-graph neural network (GNN) [25], an unsupervised graph-learning-based tier partitioning framework, is proposed to address this drawback for M3D ICs using GNNs. Considering that discrete algorithms are particularly sensitive to partitioning [26] and can potentially lead to performance degradation, analytical 3-D placement is considered to be more promising to produce solutions with higher quality.

Analytical 3-D solutions relax discrete tier partitioning and solve continuous 3-D optimization problems. Typical analytical approaches include quadratic programming [27], [28], nonlinear programming [29], and force-directed methods [30]. In addition, NTUPlace3-3D [31], [32] performs 3-D analytical placement based on a bell-shaped [33] smooth density considering TSV insertion, and mPL6-3D [34] utilizes a Huber-based local smoothing technique working with a Helmholtz-based global smoothing approach. Based on mPL6-3D [34], ART-3D [26] improves placement quality using reinforcement learning-based parameter tuning. The SOTA analytical placement is the ePlace family [13], [35], [36], [37] where the density constraint is modeled by an electrostatic field. Lu et al. [13] proposed a general 3-D eDensity model in ePlace-3D achieving analytically global smoothness along all dimensions in 3-D domain. Remarkably, the ePlace family has achieved substantial success in wirelength-driven analytical placement, and their adoption of fast Fourier transform (FFT) for solving the 3-D numerical solution has inspired quality enhancement [38] and GPU-accelerated ultrafast implementations [39], [40].

Unfortunately, the aforementioned previous works have difficulties in considering heterogeneous technologies and specific utilization constraints, and thus lead to poor performance when applied to heterogeneous 3-D placement problems. Recently, Chen et al. [14] proposed a 3-D analytical placement algorithm to optimize wirelength considering F2F-bonded 3-D ICs with multiple manufacturing technologies. They devise a multitechnologies weighted-average (MTWA) wirelength model using sigmoid-based functions for pin offset transition, and establish their framework based on ePlace-3D [13]. A 2-D analytical placement, considering the accurate wirelength, is employed after the 3-D global placement to further refine the solution. The aforementioned works, including [14], adopt the 3-D net bounding box as the wirelength model which is not capable of capturing enough information of the impact of partition on wirelength.

B. Analytical Placement

Global placement is performed on a netlist (V, E), where $V = \{c_1, \ldots, c_n\}$ and $E = \{e_1, \ldots, e_m\}$ are the node set and the net set, respectively. We are asked to determine the node locations $\boldsymbol{v} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ from scratch during global placement to minimize the total wirelength with little overlap allowed. A typical 3-D analytical global placement problem is formulated as the following unconstrained optimization problem:

$$\min_{\boldsymbol{v}} \sum_{e \in E} W_e(\boldsymbol{v}) + \lambda D(\boldsymbol{v}) \tag{1}$$

where $\boldsymbol{v} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ indicates the node location variables, $W_e(\cdot)$ is the net wirelength model of net $e \in E$, $D(\cdot)$ is the density model of the entire placement region evaluating the overall overlap, and λ is the density weight introduced as the Lagrangian multiplier of the density constraint. In analytical placement, we expect to make the objective differentiable and then apply numerical methods to solve (1).

The wirelength model $W_e(\cdot)$ in the above (1) is usually a differentiable approximation [31], [32], [41], [42] to the conventional net HPWL defined below.

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

$$W_e(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = p_e(\boldsymbol{x}) + p_e(\boldsymbol{y}) + \alpha p_e(\boldsymbol{z}) \tag{2}$$

where $p_e(\boldsymbol{u}) = \max_{c_i \in e} u_i - \min_{c_i \in e} u_i$ denote the range or peak-to-peak function that evaluates the difference of maximum minus minimum in a net, and $\alpha \geq 0$ is a weight factor.

$p_e(\cdot)$ denotes partial HPWL along one axis. In real applications, it is approximated by a *differentiable* model, e.g., the weighted-average [32] model given a smoothing parameter $\gamma > 0$

$$p_{e,\mathrm{WA}}(\boldsymbol{u}) = \frac{\sum_{c_i \in e} u_i \mathrm{e}^{\frac{1}{\gamma} u_i}}{\sum_{c_i \in e} \mathrm{e}^{\frac{1}{\gamma} u_i}} - \frac{\sum_{c_i \in e} u_i \mathrm{e}^{-\frac{1}{\gamma} u_i}}{\sum_{c_i \in e} \mathrm{e}^{-\frac{1}{\gamma} u_i}}. \tag{3}$$

Other differentiable models [41], [42] are also applicable. Note that the z-dimension is usually defined manually, as tiers are discretely distributed in 3-D scenarios. The corresponding weight factor $\alpha \geq 0$ is determined in accordance with specific objectives in real applications.

The SOTA density model $D(\cdot)$ is the eDensity family [13], [35], [36], [37] based on electrostatics field, where every node $c_i \in V$ is modeled by an electric charge. We implement eDensity-3D [13] as our density model with GPU acceleration in the proposed framework.

The optimization formulation in (1) is general and thus can be applied in both 2-D and 3-D analytical global placement. In conventional 2-D cases, the variable $\boldsymbol{v} = (\boldsymbol{x}, \boldsymbol{y})$ is optimized to find planar cell coordinates [36], [39], [40]. In 3-D cases, the framework is well-established in ePlace-3D [13] where the z-direction coordinates is considered to optimize $\boldsymbol{v} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$.

III. Problem Formulation

A. Problem Statement

In this article, we focus on the 3-D placement problem with die-to-die (D2D) connections, specified in the ICCAD 2022 Contest [15]. The general requirement is to partition the given standard cells into two dies with different technologies, create vertical interconnections named hybrid bonding terminals (HBTs) for split nets, and determine the locations of all nodes, including standard cells and HBTs, so that the following constraints are satisfied.

1) *Utilization Constraints:* The utilization requirements of the top die and the bottom die are provided separately, leading to different area upper bound for two dies.
2) *Technology Constraints:* The cells may be fabricated using different technologies on different dies, i.e., the cell characteristic, cell height, cell width, and the cell layout would be different.
3) *Vertical Interconnection Constraints:* For any net e split to two dies, an HBT should be created to connect pins on two dies. All HBTs share the same size.
4) *Legality Constraints:* All standard cells on both dies should be placed without overlap and aligned to rows and sites. HBTs should be placed to satisfy the spacing constraint, i.e., the distance between each pair of HBTs and the distance to boundaries are lower bounded.

The objective of this 3-D placement problem is the total wirelength of all nets in the given design defined in Definition 3. In short, we focus on minimizing the sum of HPWL on the two dies. The center points of the HBTs are included in the HPWL calculation for each die. We will give rigorous mathematical formulations in Section III-B.

B. Problem Formulation

Consider a netlist (V, E) where $V = \{c_1, \ldots, c_n\}$ is the node set and $E = \{e_1, \ldots, e_m\}$ is the net set. A partition is determined by a 0-1 vector $\boldsymbol{\delta} \in \mathbb{Z}_2^n = \{0, 1\}^n$, where $\delta_i = 0$ indicates that cell $c_i \in V$ is placed on the bottom die, otherwise top die. In the 3-D placement with D2D vertical connections, the partition determines the total number of hybrid bonding terminals. In this section, we use $\boldsymbol{x}, \boldsymbol{y}$ to represent both node coordinates and corresponding pin coordinates ignoring pin offsets for simplicity.

Definition 2 (Net Cut Indicator): The cut indicator of a net $e \in E$ is a function of partition $\boldsymbol{\delta} \in \{0, 1\}^n$ defined by

$$C_e(\boldsymbol{\delta}) = \max_{c_i \in e} \delta_i - \min_{c_i \in e} \delta_i. \tag{4}$$

It is also a binary value in $\{0, 1\}$. If there exist two nodes incident to net e placed on two different dies, the cut $C_e(\boldsymbol{\delta}) = 1$, otherwise it is 0.

Given a partition $\boldsymbol{\delta} \in \{0, 1\}^n$, if a net $e \in E$ is a split net, i.e., $C_e(\boldsymbol{\delta}) = 1$, a HBT should be inserted for this net as a vertical connection. Otherwise, all nodes incident to $e \in E$ are placed on either the top or the bottom die. Different from TSVs and MIVs going through silicon substrates, HBTs do not require silicon space. If we have $C_{e_i}(\boldsymbol{\delta}) = 1$ for a net $e_i \in E$, one and only one HBT t_i should be assigned to e_i accordingly, otherwise t_i will be discarded. We denote the set of HBTs by $T = \{t_1, \ldots, t_m\}$ with planar coordinates $\boldsymbol{x}', \boldsymbol{y}'$.

Denote the top and bottom partial nets by $e^+(\boldsymbol{\delta}) = \{c_i \in e : \delta_i = 1\}$ and $e^-(\boldsymbol{\delta}) = \{c_i \in e : \delta_i = 0\}$, respectively. Correspondingly, the complete nets on top and bottom dies

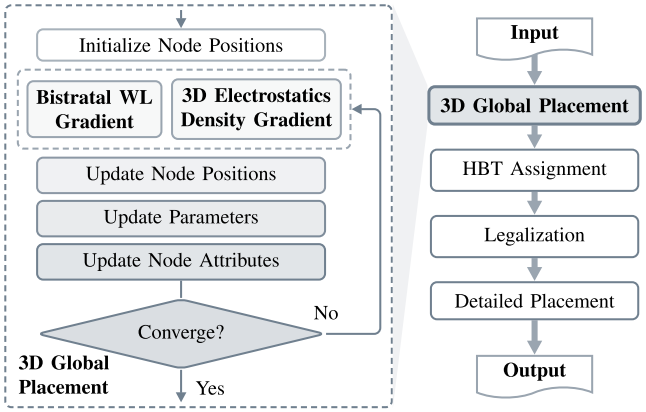


Fig. 1. Overall placement flow of our framework.

are $\tilde{e}_i^+ = e_i^+ \cup \{t_i\}$ and $\tilde{e}_i^- = e_i^- \cup \{t_i\}$, respectively, including HBTs. The D2D wirelength [15] of net $e \in E$ is defined as follows.

Definition 3 (D2D Net Wirelength): Given partition $\boldsymbol{\delta}$, the D2D wirelength of net e is defined by $W_e = W_{\tilde{e}_i^+} + W_{\tilde{e}_i^-}$. More specifically, we have

$$\begin{aligned} W_{\tilde{e}_i^+} &= \max_{c_j \in \tilde{e}_i^+} x_j - \min_{c_j \in \tilde{e}_i^+} x_j + \max_{c_j \in \tilde{e}_i^+} y_j - \min_{c_j \in \tilde{e}_i^+} y_j \\ W_{\tilde{e}_i^-} &= \max_{c_j \in \tilde{e}_i^-} x_j - \min_{c_j \in \tilde{e}_i^-} x_j + \max_{c_j \in \tilde{e}_i^-} y_j - \min_{c_j \in \tilde{e}_i^-} y_j. \end{aligned} \tag{5}$$

If $C_{e_i}(\boldsymbol{\delta}) = 0$, it degrades to the ordinary net HPWL without HBT considered.

The D2D net wirelength in Definition 3 simply sums up the half-perimeter wirelength on two dies, demonstrating equivalence to $p_{\tilde{e}_i^+}(\boldsymbol{x}) + p_{\tilde{e}_i^-}(\boldsymbol{x}) + p_{\tilde{e}_i^+}(\boldsymbol{y}) + p_{\tilde{e}_i^-}(\boldsymbol{y})$. Since the center point of HBT t_i is included, W_e is a function of node locations $\boldsymbol{x}, \boldsymbol{y}$, HBT locations $\boldsymbol{x}', \boldsymbol{y}'$, and partition $\boldsymbol{\delta}$. Our problem is formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{\delta},\boldsymbol{x}',\boldsymbol{y}'} \quad & \sum_{e \in E} W_e(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\delta}, \boldsymbol{x}', \boldsymbol{y}') \\ \text{s.t.} \quad & \sum_{i=1}^{n} \delta_i a_i^+ \le a_{\text{req}}^+ \\ & \sum_{i=1}^{n} (1 - \delta_i) a_i^- \le a_{\text{req}}^- \\ & \text{legality constraints} \end{aligned} \tag{6}$$

where a_i^+, a_i^- stand for the node area of c_i on the top and bottom die, respectively. The area requirements are set to $a_{\text{req}}^+, a_{\text{req}}^-$ correspondingly. Besides of the legality constraints of standard cells, all HBTs have a specific legality rule that the distance between each other is lower bounded. It worth mentioning that HBTs are on the top-most metal layer and thus would not occupy any placement resources on both dies.

IV. Overall Placement Flow

The overall placement flow of our proposed framework is illustrated in Fig. 1. We adopt a 3-D analytical global placement to find node locations with three dimensions. After global placement, we assign HBTs and legalize all nodes, including HBTs. At last, we perform detailed placement on each die to further refine the solution. The optimized circuit placement results will be output after detailed placement. Note that we do not apply 2-D placement after 3-D global placement and HBT assignment, as we are confident enough of our proposed 3-D global placement which effectively handles partitioning and planar placement together.

A. Global Placement

In 3-D placement, we assign coordinates $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^n$ to all nodes. Given the top die $[x_{\min}^+, x_{\max}^+] \times [y_{\min}^+, y_{\max}^+]$ and the bottom die $[x_{\min}^-, x_{\max}^-] \times [y_{\min}^-, y_{\max}^-]$, we have to make a necessary and realistic assumption that they differ very little so that the entire placement region is well-defined and the 3-D placement framework makes sense under this scenario.

Assumption 1: The die sizes of two dies are almost the same. Specifically, we have die width $x_{\min}^+ = x_{\min}^- = 0$, $x_{\max}^+ = x_{\max}^-$, and die height $y_{\min}^+ = y_{\min}^- = 0$, $|(y_{\max}^+ / y_{\max}^-) - 1| < \epsilon$, where $\epsilon > 0$ is a small tolerance.

Under Assumption 1, our 3-D global placement region is set to a cuboid $\Omega = [0, x_{\max}^+] \times [0, y_{\max}^+] \times [0, z_{\max}]$ by default, with a properly determined depth $z_{\max}$. For each node $c_i \in V$, along with its width and height provided by the input files, it will also be assigned a unified depth d.

Different from the 2-D cases, the partition values $\boldsymbol{\delta}$ are restricted to take very discrete values in 3-D placement to determine node partition. More specifically, $\boldsymbol{\delta}$ must be constrained to take binary values in $\{0, 1\}^n$ in our placement problem, described in Section III-B, so that each node $c_i \in V$ has an assigned partition indicator. We equally split the placement cuboid Ω into two parts by the plane $z = (1/2)z_{\max}$, each of which represents a die

$$\begin{aligned} \Omega^+ &= \left[0, x_{\max}^+\right] \times \left[0, y_{\max}^+\right] \times \left[\frac{z_{\max}}{2}, z_{\max}\right] \\ \Omega^- &= \left[0, x_{\max}^+\right] \times \left[0, y_{\max}^+\right] \times \left[0, \frac{z_{\max}}{2}\right]. \end{aligned} \tag{7}$$

The unified node depth is $d = (1/2)z_{\max}$. Ideally, we expect every node $c_i \in V$ to be placed inside either the top part Ω^+ or the bottom part Ω^- at the end of 3-D global placement. Note that every node should not be placed out of boundary, therefore z_i, which stands for the corner point coordinate of node c_i, should take values within interval $[0, (1/2)z_{\max}]$. We determine the *tentative* node partition $\boldsymbol{\delta}$ as a function of z coordinates $P(\boldsymbol{z})$, by rounding the normalized value $(2/[z_{\max}])\boldsymbol{z}$ at every iteration, i.e., we have

$$\delta_i = \left\lceil \frac{2z_i}{z_{\max}} - \frac{1}{2} \right\rceil \tag{8}$$

for every $c_i \in V$.

An example of partition mapping $\boldsymbol{\delta} = P(\boldsymbol{z})$ is depicted in Fig. 2(a). Node c_i is partitioned to the bottom die, i.e., $\delta_i = 0$ as its corner coordinate $z_i < (1/4)z_{\max}$. The other node c_j in Fig. 2(a) is partitioned to the top die, i.e., $\delta_i = 1$ as its corner coordinate $z_j > (1/4)z_{\max}$. The exact value of the cuboid depth $z_{\max}$ should be determined properly to avoid ill-condition in numerical optimization.

We manually set the bin size of z dimension to the mean of bin sizes of the other two dimensions. More specifically,

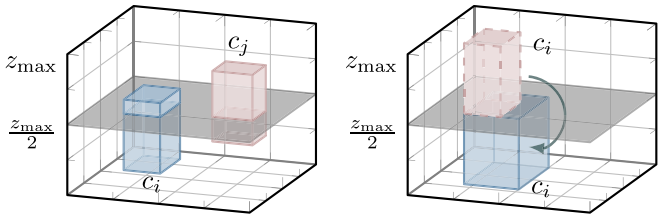


Fig. 2. Partition mapping $P(z) : [0, z_{\max}] \to \{0, 1\}$ and the update of node attributes for heterogeneous technologies. (a) At one iteration during 3-D global placement, node c_i has *tentative* partition $\delta_i = 0$ indicating the bottom die, while node c_j with $\delta_j = 1$ is assigned to the top die. (b) Node size and pin offset values of node $c_i \in V$ will change if moved to the other die.

suppose the placement region is uniformly decomposed into $N_x \times N_y \times N_z$ grids, then we set

$$z_{\max} = \frac{N_z}{2}\left(\frac{x_{\max}}{N_x} + \frac{y_{\max}}{N_y}\right) \tag{9}$$

in our analytical placement.

Heterogeneous Technologies: Different from ordinary analytical placement, we have to face a challenge of heterogeneous technologies that the node attributes, including node sizes and pin offset values, are different on the two dies.

Assume that each node $c_i \in V$ has width w_i^+ and height h_i^+ on the top die and w_i^-, h_i^- on the bottom die. At each iteration of 3-D global placement, we should determine the exact node size for every node according to tentative partition $\boldsymbol{\delta} = P(\boldsymbol{z})$. More specifically, if the tentative partition $\delta_i = 1$, w_i^+, h_i^+ will be adopted for node c_i, otherwise it will use w_i^-, h_i^-. In other words, the planar node size for node $c_i \in V$ is calculated as

$$\begin{aligned} w_i &= \delta_i w_i^+ + (1 - \delta_i) w_i^- \\ h_i &= \delta_i h_i^+ + (1 - \delta_i) h_i^- \end{aligned} \tag{10}$$

where the tentative partition δ_i determined by (8) is a binary value. The node depth remains $d = (1/2)z_{\max}$ in the entire process of 3-D global placement.

In addition to the node size, we also have two sets of pin offset values, although they are ignored for simplicity in previous wirelength notations. Denote all pins by $P = \{p_1, \ldots, p_l\}$, and P_i is the set of all pins on the node $c_i \in V$. Now, let $\boldsymbol{x}_{\text{offset}}, \boldsymbol{y}_{\text{offset}}, \boldsymbol{z}_{\text{offset}} \in \mathbb{R}^l$ be the pin offset vectors on three dimensions. For any $p_j \in P_i$, we have

$$\begin{aligned} x_{\text{offset},j} &= \delta_i x_{\text{offset},j}^+ + (1 - \delta_i) x_{\text{offset},j}^- \\ y_{\text{offset},j} &= \delta_i y_{\text{offset},j}^+ + (1 - \delta_i) y_{\text{offset},j}^- \end{aligned} \tag{11}$$

and $z_{\text{offset},j} = (1/4)z_{\max}$ is fixed. In other words, the pin offset values of every pin is determined by the tentative partition of the node it belongs to. Besides, we have a fact that, for any $p_j \in P_i$, node c_i's tentative partition $\delta_i = 1$ if and only if pin p_j is on the top part Ω^+: $z_i + z_{\text{offset},j} \geq (z_{\max}/2)$.

In accordance with (10) and (11), we update the *node attributes*, including node size and pin offset, at every iteration during 3-D global placement. An example of updating node attributes is illustrated in Fig. 2(b) where node c_i is moved from $z_i = (1/2)z_{\max}$ to $z_i = 0$.

Electrostatics-Based 3-D Density: As mentioned in Section II-B, eDensity [36] is the SOTA academic density model which analogizes every node c_i to a positive electric charge q_i. It expects an electric equilibrium so that movable objects can be evened out to reduce the overall node overlap. Extending the density model in [36], ePlace-3D [13] computes the potential map by solving the 3-D Possion's equation under Neumann boundary condition

$$\begin{aligned} \Delta\phi &= -\rho, \quad \text{in } \Omega \\ \hat{\boldsymbol{n}} \cdot \nabla\phi &= 0, \text{ on } \partial\Omega \end{aligned} \tag{12}$$

where $\rho = \rho(x, y, z)$ is the current density map in placement region $\Omega = [0, x_{\max}] \times [0, y_{\max}] \times [0, z_{\max}]$ computed using node locations. The second line in (12) is the boundary condition specifying that the electric force on the boundary is zero.

Suppose the placement region Ω is uniformly decomposed into $N_x \times N_y \times N_z$ grids, the solution to (12) under constraint $\int_\Omega \phi \, d\Omega = 0$ is given by

$$\phi = \sum_{j,k,l} \frac{a_{jkl}}{\omega_j^2 + \omega_k^2 + \omega_l^2} \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z) \tag{13}$$

where the tuple $(\omega_j, \omega_k, \omega_l) = ([j\pi/x_{\max}], [k\pi/y_{\max}], [l\pi/z_{\max}])$ stands for frequency indices. The density coefficients a_{jkl} is defined by

$$a_{jkl} = \frac{1}{N} \sum_{x,y,z} \rho \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z) \tag{14}$$

where the denominator $N = N_x N_y N_z$ denotes the total number of bins. Note that the DC component of density map ρ has been removed, i.e., $\int_\Omega \rho \, d\Omega = 0$ is satisfied by removing $a_{000} = (1/N) \sum_{x,y,z} \rho(x, y, z)$ which equals to the average density of all bins. The electric field $\boldsymbol{E}(x, y, z) = (E_x, E_y, E_z)$ can be directly derived from (13) by taking partial derivatives of ϕ

$$\begin{aligned} E_x &= \sum_{j,k,l} \frac{a_{jkl}\omega_j}{\omega_j^2 + \omega_k^2 + \omega_l^2} \sin(\omega_j x) \cos(\omega_k y) \cos(\omega_l z) \\ E_y &= \sum_{j,k,l} \frac{a_{jkl}\omega_k}{\omega_j^2 + \omega_k^2 + \omega_l^2} \cos(\omega_j x) \sin(\omega_k y) \cos(\omega_l z) \\ E_z &= \sum_{j,k,l} \frac{a_{jkl}\omega_l}{\omega_j^2 + \omega_k^2 + \omega_l^2} \cos(\omega_j x) \cos(\omega_k y) \sin(\omega_l z). \end{aligned} \tag{15}$$

Equations (13) and (15) are well-established in [13], demonstrating that these spectral equations can be solved efficiently using FFT with $O(N \log N)$ time complexity.

Different from the general scenarios in [13] where they may have multiple tiers, we only have two dies in our specific problem. To help the 3-D electrostatic filed even out the standard cells to different dies, the node depth is set to $d = (1/2)z_{\max}$ by default, as mentioned above. Through the numerical optimization of 3-D global placement, standard cells are expected to be roughly distributed within either Ω^+ or Ω^-, so that the tentative partition $\boldsymbol{\delta} = P(\boldsymbol{z})$ does not introduce significant wirelength degradation after 3-D global placement.

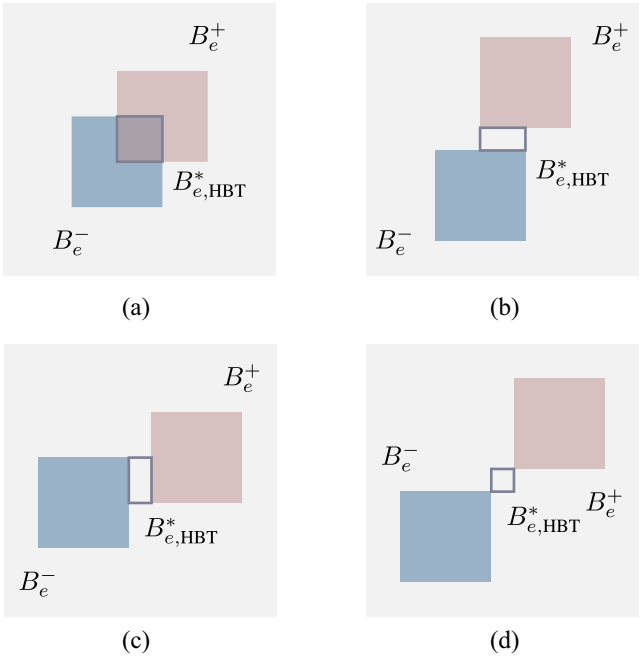


Fig. 3. Optimal region $B_{e,\mathrm{HBT}}^*$ of an HBT for a split net $e \in E$ with cut $C_e(\boldsymbol{\delta}) = 1$ under several different scenarios. (a) Top net bounding box B_e^+ and the bottom net bounding box B_e^- overlap on both the x dimension and the y dimension. (b) B_e^+ and B_e^- overlap only on the x dimension. (c) B_e^+ and B_e^- overlap only on the y dimension. (d) B_e^+ and B_e^- have no overlap on both two dimensions.

B. HBT Assignment

During 3-D global placement, we do *NOT* insert HBTs as any HBT is allowed to have overlap with standard cells. After 3-D global placement, we first obtain a partition $\boldsymbol{\delta} = P(\boldsymbol{z}) \in \mathbb{Z}_2^n$ according to (8). The convergence of global placement implies a very low overflow indicating that z_i should be close to either 0 or $(1/2)z_{\max}$ to determine the partition solution. Since the partition $\boldsymbol{\delta}$ and $\boldsymbol{x}, \boldsymbol{y}$ is already determined, we proceed to the 2-D scenario with the top die layout and the bottom die layout. Every split net e should be assigned precisely one HBT.

Consider a split net $e \in E$. Ignoring pin offset values for simplicity, define x-dimension coordinates $x_{\mathrm{low}}^+ = \min_{c_i \in e^+} x_i$, $x_{\mathrm{high}}^+ = \max_{c_i \in e^+} x_i$ and vertical coordinates $y_{\mathrm{low}}^+, y_{\mathrm{high}}^+$ for the top partial net e^+, and similarly define corresponding variables for the bottom partial net e^-. Then, we denote the bounding box of partial nets e^+ and e^- by

$$\begin{aligned} B_e^+ &= \left[x_{\mathrm{low}}^+, x_{\mathrm{high}}^+\right] \times \left[y_{\mathrm{low}}^+, y_{\mathrm{high}}^+\right] \\ B_e^- &= \left[x_{\mathrm{low}}^-, x_{\mathrm{high}}^-\right] \times \left[y_{\mathrm{low}}^-, y_{\mathrm{high}}^-\right] \end{aligned} \tag{16}$$

respectively.

After 3-D global placement, the $\boldsymbol{x}, \boldsymbol{y}$ coordinates and partition $\boldsymbol{\delta} = P(\boldsymbol{z})$ of nodes are already determined, and thus B_e^+ and B_e^- are determined for every split net e. As illustrated in Fig. 3, for any split net e, its HBT has a specific *optimal region*, i.e., the net wirelength W_e is minimized only when its HBT is placed within this optimal region.

Theorem 1: For a split net $e \in E$, the optimal region of its HBT is defined by $B_{e,\mathrm{HBT}}^* = [x'_{\mathrm{low}}, x'_{\mathrm{high}}] \times [y'_{\mathrm{low}}, y'_{\mathrm{high}}]$ where

$$\begin{aligned} x'_{\mathrm{low}} &= \min\left\{\max\left\{x_{\mathrm{low}}^+, x_{\mathrm{low}}^-\right\}, \min\left\{x_{\mathrm{high}}^+, x_{\mathrm{high}}^-\right\}\right\} \\ x'_{\mathrm{high}} &= \max\left\{\max\left\{x_{\mathrm{low}}^+, x_{\mathrm{low}}^-\right\}, \min\left\{x_{\mathrm{high}}^+, x_{\mathrm{high}}^-\right\}\right\} \end{aligned} \tag{17}$$

and $y'_{\mathrm{low}}, y'_{\mathrm{high}}$ are defined similarly. Equivalently, coordinates $x'_{\mathrm{low}}, x'_{\mathrm{high}}$ are the two median numbers of $x_{\mathrm{low}}^+, x_{\mathrm{low}}^-, x_{\mathrm{high}}^+, x_{\mathrm{high}}^-$ and the same for $y'_{\mathrm{low}}, y'_{\mathrm{high}}$.

Theorem 1 enlightens us that the total net wirelength will be minimized when every split net e has its HBT placed within the optimal region $B_{e,\mathrm{HBT}}^*$. Therefore, we intuitively assign an HBT $t(e) \in T$ for each split net such that the center point of t locates exactly at the center point of $B_{e,\mathrm{HBT}}^*$.

Note that after this *HBT assignment* step, it is likely that HBTs may overlap with each other, requiring a subsequent legalization process. To control the total number of HBTs and mitigate potential wirelength degradation caused by legalization, we carefully regulate the weight α in the objective function described in Definition 1. This enables us to mitigate wirelength degradation while minimizing the number of HBTs.

C. Legalization

After the partitioning $\boldsymbol{\delta} = P(\boldsymbol{z})$ and the HBT assignment, the mission of 3-D global placement is completed. The rest is to legalize all nodes, including HBTs and further refine the solution from 2-D perspective. We legalize the standard cells on the top die and the bottom die separately with Tetris [43] and Abacus [44]. The HBTs are legalized similarly by treating them as ordinary standard cells with a specific terminal size.

Note that in our problem definition, HBTs share the same square size $w' \times w'$ and every pair of HBTs must satisfy the spacing constraint that the distance of boundaries should be no less than s'. Hence, we pad every HBT to a square with size $w' + s'$ and legalize them as ordinary standard cells with row height $w' + s'$.

D. Detailed Placement

We further improve the total wirelength by applying ABCDPlace [45] with several techniques, including global swap [46], [47], independent set matching [48], and local reordering [46], [48], die by die. When we are performing detailed placement on one die, all other nodes on the other die and HBTs remain fixed. After the detailed placement of two dies, the optimal regions of HBTs may get affected. Therefore, we can continue to map HBTs to their updated optimal regions, followed by a new round of HBT legalization and detailed placement. While this process can be iterated infinitely, we find that only the initial few rounds yield significant benefits. Therefore, we perform one additional round of this process during the detailed placement.

V. Bistratal Wirelength Model

The analytical wirelength model is critical to the numerical optimization of (1) in this problem. Previous works [13], [31], [32] use the 3-D HPWL model defined

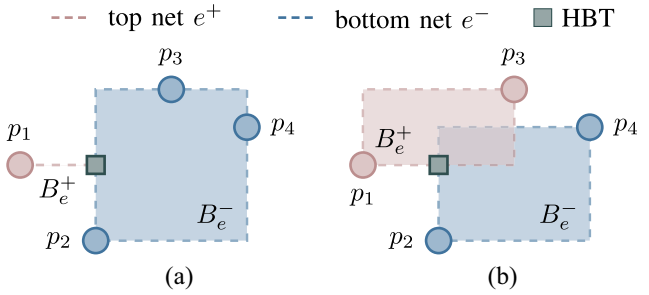


Fig. 4. Example where changing partition of one pin does not affect the net bounding box but increases the exact net wirelength. (a) Exact wirelength equals to the HPWL of the entire net. (b) Exact wirelength is strictly larger than the HPWL of the entire net.

in Definition 1 with the peak-to-peak function to describe the net wirelength. Chen et al. [14] proposed MTWA model to consider heterogeneous technologies, but it is still based on 3-D HPWL without considering the D2D wirelength. Note that $p_e(\boldsymbol{z})$ roughly reflects the cut size of net e and does not contribute to the planar net wirelength. The plain HPWL $\tilde{W}_e$ is defined as follows such that $W_e(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \tilde{W}_e(\boldsymbol{x},\boldsymbol{y}) + \alpha p_e(\boldsymbol{z})$.

Definition 4 (Plain HPWL): Given node positions $\boldsymbol{x},\boldsymbol{y}$, the plain HPWL of any net $e \in E$ is given by

$$\tilde{W}_e(\boldsymbol{x},\boldsymbol{y}) = \max_{c_i\in e} x_i - \min_{c_i\in e} x_i + \max_{c_i\in e} y_i - \min_{c_i\in e} y_i \tag{18}$$

which does not care node position $\boldsymbol{z}$ at all.

Obviously, (18) in the above definition is equivalent to the separable representation $\tilde{W}_e(\boldsymbol{x},\boldsymbol{y}) = p_e(\boldsymbol{x}) + p_e(\boldsymbol{y})$ using the peak-to-peak function defined in (2).

Unfortunately, 3-D HPWL model in Definition 1 based on the plain HPWL is *inaccurate* as the exact wirelength defined in Definition 3 and (5) sums up the HPWL on the top die and bottom die. Equation (18) only considers the entire bounding box with the top die and the bottom die together, neglecting the pin partition and the potential presence of HBTs. Additionally, the conventional 3-D HPWL wirelength model is *NOT* able to capture the wirelength variation resulting from different node partition.

Consider a net $e \in E$ connecting four pins p_1, p_2, p_3, p_4. Fix all planar locations of these pins and tentative partition of p_2, p_3, p_4. Fig. 4(a) shows e^+ and e^- when p_3 is on the bottom die and the corresponding HBT is placed optimally. It is clear that the total wirelength of net e is $W_e = \tilde{W}_{\tilde{e}^+} + \tilde{W}_{\tilde{e}^-}$ which exactly equals to the *plain* HPWL of the entire net e. By contrast, Fig. 4(b) shows the case when p_3 is on the top die. The HBT with the same coordinates preserves optimality, but the true wirelength $W_e = \tilde{W}_{\tilde{e}^+} + \tilde{W}_{\tilde{e}^-}$ is larger than the *plain* HPWL of net e.

Theorem 2: Given any partition $\boldsymbol{\delta}$ and any net $e \in E$, let $p_e(\boldsymbol{u}) = \max_{c_i\in e} u_i - \min_{c_i\in e} u_i$ be the peak-to-peak function defined in (2). Then, we always have

$$p_e \le \min_{x'} W_{e_x}(x') \le 2p_e \tag{19}$$

where $W_{e_x}(x')$ is the x-dimension part of the exact net wirelength defined in (5) with HBT coordinate x' under tentative partition $\boldsymbol{\delta}$.

The equality of the left part of (19) holds if and only if B_e^+ and B_e^- defined in (16) has no overlap on the x dimension. The equality of the right part holds if and only if B_e^+ and B_e^- are the same on the x dimension. The conclusion on the y dimension can be similarly established. We will give a more detailed representation of $\min_{x'} W_{e_x}(x')$ in Theorem 3.

Corollary 1: Given any partition $\boldsymbol{\delta}$ and any net $e \in E$, let the ordinary plain HPWL be $\tilde{W}_e$ defined in Definition 4. Then, we always have

$$\tilde{W}_e \le \min_{x',y'} W_e(x',y') \le 2\tilde{W}_e \tag{20}$$

where $W_e(\boldsymbol{\delta},x',y')$ is the exact net wirelength defined in (5) with HBT coordinate (x',y').

The equality of the left part of (20) holds if and only if B_e^+ and B_e^- defined in (16) has no overlap on both x and y dimensions. The equality of the right part holds if and only if B_e^+ and B_e^- are the same. Corollary 1 indicates that the HPWL model used in previous works [13], [31], [32] is just a lower bound of the exact bistratal wirelength in our problem. Apparently, optimizing $\tilde{W}_e$ does not necessarily benefit the exact wirelength as the error bound may get as large as the lower bound, according to (20). We will give a precise representation of $\min_{x',y'} W_e(x',y')$ for every net e in Theorem 3.

We propose a novel *bistratal* wirelength model that handles planar coordinates and partitioning together. Instead of optimizing $\tilde{W}_e$, we try to minimize $\min_{\boldsymbol{x}',\boldsymbol{y}'} W_e(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\delta},\boldsymbol{x}',\boldsymbol{y}')$ at every iteration according to the tentative partition. Besides of the wirelength estimation, the computation of gradients is more critical to the numerical optimization process. In this section, we will discuss the proposed model theoretically in detail.

A. Wirelength Objective

In the forward pass of numerical optimization [36], [39], we calculate the exact or approximated wirelength. Different from 3-D HPWL in Definition 1, we must consider partition for more precise wirelength estimation.

According to Corollary 1, we should approximate the exact wirelength W_e defined in (5) as precisely as possible. Considering that HBTs are not inserted in the 3-D global placement as the tentative partition $\boldsymbol{\delta}$ may vary at every iteration, we assume that each split net is assigned a dummy HBT placed within its optimal region according to the tentative partition. In other words, we target at optimizing $\min_{\boldsymbol{x}',\boldsymbol{y}'} W_e(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\delta},\boldsymbol{x}',\boldsymbol{y}')$ where the tentative partition $\boldsymbol{\delta} = P(\boldsymbol{z})$ is updated at every iteration. The following theorem reveals the explicit representation of our wirelength forward computation without any HBT inserted.

Theorem 3: The minimal precise net wirelength on the x dimension with respect to the HBT coordinate x' of net e is given by

$$\min_{x'} W_{e_x}(x') = \max\{p_e, p_{e^+} + p_{e^-}\} \tag{21}$$

as a function of node positions $(\boldsymbol{x})$ under partition $\boldsymbol{\delta}$, where the peak-to-peak function p_e is defined by $p_e(\boldsymbol{u}) = \max_{c_i\in e} u_i - \min_{c_i\in e} u_i$ for any $\boldsymbol{u}$.

Theorem 3 gives an accurate estimation of the minimum exact net wirelength on the x dimension for split nets at every iteration during 3-D global placement. Note that the right-hand side of (21) also indicates the exact wirelength for any nonsplit net e as either p_{e^+} or p_{e^+} is zero. The corresponding theorem on the y dimension can be similarly established.

Given any partition $\boldsymbol{\delta}$ and any split net $e \in E$, let $B_e^+ = [x_{\text{low}}^+, x_{\text{high}}^+] \times [y_{\text{low}}^+, y_{\text{high}}^+]$ and $B_e^- = [x_{\text{low}}^-, x_{\text{high}}^-] \times [y_{\text{low}}^-, y_{\text{high}}^-]$ be the bounding boxes of partial nets e^+, e^-, respectively, defined in (16). Define

$$W_{ex} = \begin{cases} \max_{c_i \in e} x_i - \min_{c_i \in e} x_i, & \text{if } x_{\text{high}} \le x_{\text{low}} \\ x_{\text{high}}^+ - x_{\text{low}}^+ + x_{\text{high}}^- - x_{\text{low}}^-, & \text{otherwise} \end{cases} \tag{22}$$

where $x_{\text{low}} = \max\{x_{\text{low}}^+, x_{\text{low}}^-\}, x_{\text{high}} = \min\{x_{\text{high}}^+, x_{\text{high}}^-\}$, and similarly define W_{ey}. Then, the minimal precise net wirelength considering both x, y dimensions with respect to the HBT coordinates (x', y') of net $e \in E$ defined in Theorem 3 is equivalent to

$$\min_{x', y'} W_e(x', y') = W_{ex} + W_{ey}. \tag{23}$$

More intuitively, (23) first checks whether the boxes B_e^+ and B_e^- overlap. If they overlap on one dimension, we optimize the HPWL of the top partial net e^+ and the bottom partial net e^- on this dimension separately, as we have

$$\begin{aligned} x_{\text{high}}^+ - x_{\text{low}}^+ &= p_{e^+}(\boldsymbol{x}) = \max_{c_i \in e^+} x_i - \min_{c_i \in e^+} x_i \\ x_{\text{high}}^- - x_{\text{low}}^- &= p_{e^-}(\boldsymbol{x}) = \max_{c_i \in e^-} x_i - \min_{c_i \in e^-} x_i \end{aligned} \tag{24}$$

otherwise the target degrades to the ordinary HPWL function $\tilde{W}_e$ on this dimension.

For a nonsplit net $e \in E$ with $C_e(\boldsymbol{\delta}) = 0$, i.e., it is completely within either the top or the bottom die, we treat it as an ordinary 2-D net and evaluate its ordinary plain wirelength $\tilde{W}_e$ with (18). Then, we propose the *bistratal wirelength* (BiHPWL) as follows.

Definition 5 (Bistratal Wirelength): Given 3-D node position $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, the bistratal half-perimeter wirelength of any net e is defined as

$$\begin{aligned} W_{e,\text{Bi}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = &\max\{p_e(\boldsymbol{x}), p_{e^+}(\boldsymbol{x}) + p_{e^-}(\boldsymbol{x})\} \\ &+ \max\{p_e(\boldsymbol{y}), p_{e^+}(\boldsymbol{y}) + p_{e^-}(\boldsymbol{y})\} \end{aligned} \tag{25}$$

where the peak-to-peak function $p_e(\cdot)$ is defined by $p_e(\boldsymbol{u}) = \max_{c_i \in e} u_i - \min_{c_i \in e} u_i$ for any $\boldsymbol{u}$. The partial nets $e^+(\boldsymbol{\delta})$ and $e^-(\boldsymbol{\delta})$ are determined by the tentative partition $\boldsymbol{\delta} = P(\boldsymbol{z})$.

Definition 5 gives a much accurate wirelength estimation in our problem. Combining the regularization of cut size, In our 3-D global placement, we use

$$W(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) := \sum_{e \in E} W_{e,\text{Bi}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) + \alpha \sum_{e \in E} p_e(\boldsymbol{z}) \tag{26}$$

as the wirelength objective where the bistratal net wirelength $W_{e,\text{Bi}}$ is defined by (25). Note that $W_{e,\text{Bi}}$ is also a function of $\boldsymbol{z}$ as the tentative partition $\boldsymbol{\delta}$ at every iteration is determined by $\boldsymbol{z}$. The second term with α weight is integrated to limit the total number of HBTs as we always expect fewer HBTs if possible. Moreover, a large number of HBTs would degrade the solution quality after legalization.

Optimizing (26) resolves the issue that 3-D HPWL approximates the true wirelength poorly when the top box B_e^+ and the bottom box B_e^- overlap, illustrated in Fig. 4(b). However, the objective in (26) is highly nondifferentiable. Therefore, we should establish the gradient approximation in detail to enable numerical optimization of 3-D global placement. In the following of this section, we will discuss the gradient computation, including the subgradient approximation, to the planar gradients and the finite difference approximation (FDA) to the depth gradient.

B. Gradient Computation

The optimization of (26) itself is difficult as it is nondifferentiable and even discontinuous with respect to $\boldsymbol{z}$. In this section, we will discuss our proposed strategy to find the "*gradients*" that percept the objective change with respect to variables.

Since the representations in (22) are always in a peak-to-peak form, we use the weighted-average model [31], [32] in (3) to approximate them, so that W_e in (21) is *differentiable* where $x_{\text{high}} \neq x_{\text{low}}$ and $y_{\text{high}} \neq y_{\text{low}}$ when calculating gradients. It is straight-forward to derive the closed-form representation of gradients of the WA model [31] described in (3)

$$\frac{\partial p_{e,\text{WA}}}{\partial u_i} = \frac{e^{\frac{u_i}{\gamma}}(\gamma + u_i - S_{\max})}{\gamma \sum_{c_i \in e} e^{\frac{u_i}{\gamma}}} - \frac{e^{-\frac{u_i}{\gamma}}(\gamma + S_{\min} - u_i)}{\gamma \sum_{c_i \in e} e^{-\frac{u_i}{\gamma}}} \tag{27}$$

where the smooth maximum $S_{\max} = S_{\max}(\boldsymbol{u})$ and the smooth minimum $S_{\min} = S_{\min}(\boldsymbol{u})$ are defined by

$$S_{\max} = \frac{\sum_{c_i \in e} u_i e^{\frac{1}{\gamma} u_i}}{\sum_{c_i \in e} e^{\frac{1}{\gamma} u_i}}, \quad S_{\min} = \frac{\sum_{c_i \in e} u_i e^{-\frac{1}{\gamma} u_i}}{\sum_{c_i \in e} e^{-\frac{1}{\gamma} u_i}} \tag{28}$$

such that $p_{e,\text{WA}} = S_{\max} - S_{\min}$. The variable $\boldsymbol{u}$ can be $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ to derive the detailed gradients of the smooth peak-to-peak on corresponding dimensions. More details of differentiable approximations are discussed in [31], [32], [41], and [42].

Adaptive Planar Gradients: In the numerical optimization, we are supposed to derive the "gradients" of (26) with respect to coordinates $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$. The gradients w.r.t. planar coordinates $\boldsymbol{x}, \boldsymbol{y}$ guide the optimizer to find optimal placement on each die, while the gradients w.r.t. $\boldsymbol{z}$ handle the partition correspondingly. It is clear that the planar gradients are determined by $\nabla_{\boldsymbol{x}} W_{e,\text{Bi}}$ and $\nabla_{\boldsymbol{y}} W_{e,\text{Bi}}$. Unfortunately, $W_{e,\text{Bi}}$ in (25) is nondifferentiable, forcing us to consider *subgradients* instead.

Without loss of generality, we focus on the x dimension. Consider function set $\mathcal{F} = \{p_e, p_{e^+} + p_{e^-}\}$ for a given tentative partition, then the x-dimension part of wirelength $W_{e,\text{Bi}}$ is $W_{e_x,\text{Bi}}(\boldsymbol{x}) = \max_{f \in \mathcal{F}} f(\boldsymbol{x})$. The corresponding active function set is

$$\mathcal{I}(\boldsymbol{x}) = \{f \in \mathcal{F} : f(\boldsymbol{x}) = W_{e_x,\text{Bi}}(\boldsymbol{x})\}. \tag{29}$$

According to the subgradient calculus rule, we know that the subdifferential of $W_{e,\text{Bi}}$ is a convex hull

$$\partial W_{e_x,\text{Bi}}(\boldsymbol{x}) = \text{conv} \bigcup_{f \in \mathcal{I}(\boldsymbol{x})} \partial f(\boldsymbol{x}). \tag{30}$$

We expect to legitimately take one subgradient $g \in \partial W_{e_x,\mathrm{Bi}}(\boldsymbol{x})$ for optimization.

A nonsplit net is trivial as $W_{e_x,\mathrm{Bi}}(\boldsymbol{x})$ degrades to $p_e(\boldsymbol{x})$ directly. Consider a split net $e \in E$. When B_e^+ and B_e^- have overlap on x dimension, i.e., $p_{e^+}(\boldsymbol{x})+p_{e^-}(\boldsymbol{x}) > p_e(\boldsymbol{x})$, $p_{e^+}(\boldsymbol{x})+p_e^-(\boldsymbol{x}) \in \mathfrak{I}(\boldsymbol{x})$ is active in (29) and we have $W_{e_x,\mathrm{Bi}}(\boldsymbol{x}) = p_{e^+}(\boldsymbol{x}) + p_{e^-}(\boldsymbol{x})$. According to (30), it is straight-forward to take any subgradient in $\partial p_{e^+}(\boldsymbol{x}) + \partial p_{e^-}(\boldsymbol{x})$ for numerical optimization. Empirically, differentiable approximations of p_e may be preferred to work with smooth optimizers, and thus we take $\nabla_{\boldsymbol{x}} p_{e^+,\mathrm{WA}} + \nabla_{\boldsymbol{x}} p_{e^-,\mathrm{WA}}$ as the "gradient" $\nabla_{\boldsymbol{x}} W_{e_x,\mathrm{Bi}}$, where we leverage the weighted-average model $p_{e,\mathrm{WA}}$ [31], [32] defined in (3). When B_e^+ and B_e^- do not overlap on x dimension, i.e., $p_{e^+}(\boldsymbol{x})+p_{e^-}(\boldsymbol{x}) < p_e(\boldsymbol{x})$, $p_e(\boldsymbol{x})$ is active in (29), so we have $W_{e_x,\mathrm{Bi}}(\boldsymbol{x}) = p_e(\boldsymbol{x})$ and treat e as a nonsplit net, then apply the approximation $p_{e,\mathrm{WA}}$. When $\mathfrak{I}(\boldsymbol{x})$ is not a singleton, i.e., $p_{e^+}(\boldsymbol{x}) + p_{e^-}(\boldsymbol{x}) = p_e(\boldsymbol{x})$, we can take any element in the convex hull in (30). Through this way, we define the "gradient" $\nabla W_{e,\mathrm{Bi}}$.

Definition 6 (Planar Gradient): Consider the bistratal wirelength $W_{e,\mathrm{Bi}}$. The planar gradient $\nabla_{\boldsymbol{x}} W_{e,\mathrm{Bi}} = \boldsymbol{g}$ is defined as follows:

$$\boldsymbol{g} = \begin{cases} \nabla p_{e^+,\mathrm{WA}} + \nabla p_{e^-,\mathrm{WA}}, & \text{if } p_{e^+} + p_{e^-} > p_e \\ \nabla p_{e,\mathrm{WA}}, & \text{otherwise} \end{cases} \tag{31}$$

which is an approximation of a subgradient, where the gradient of $p_{e,\mathrm{WA}}$ is given by (27).

The gradient $\nabla_{\boldsymbol{y}} W_{e,\mathrm{Bi}}$ can be defined similarly. Note that we still use $\nabla W_{e,\mathrm{Bi}}$ to denote such a subgradient approximation in (31) although $W_{e,\mathrm{Bi}}$ itself is nondifferentiable.

We consider (31) to be the *adaptive* planar gradients w.r.t. $\boldsymbol{x}, \boldsymbol{y}$ coordinates. The term "adaptive" is named after the overlap illustrated in Fig. 4. More specifically, we check whether B_e^+ and B_e^- overlap on x (and y) dimensions under tentative $\boldsymbol{\delta}$ for every net e at every global placement iteration. If they overlap on the x (or y) dimension, we have $p_{e^+} + p_{e^-} > p_e$ and use the first representation of $\nabla W_{e,\mathrm{Bi}}$ in (31) and the second otherwise. Equation (31) is applied in our numerical optimization during the 3-D global placement. With no doubt, it takes into account the physical information of pin coordinates on both dies, making it much more accurate than the 3-D HPWL model.

FDA of Depth Gradients: In addition to the planar gradients w.r.t. $\boldsymbol{x}$ and $\boldsymbol{y}$, we are also supposed to derive how to correctly define "gradients" w.r.t. $\boldsymbol{z}$ which is far more tricky. Finding a way to optimize $\boldsymbol{z}$ is critical to the entire optimization, as it directly determines the quality of partition.

The density gradient $\nabla_{\boldsymbol{z}} D(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ drives placer to separate nodes with depth $(1/2)z_{\max}$ to be distributed on two dies so that we can obtain a valid partition at last, neglecting wirelength optimization. The gradient $\sum_e \nabla_{\boldsymbol{z}} p_{e,\mathrm{WA}}(\boldsymbol{z})$ in (26) with the weighted-average model [32] tends to optimize the total cutsize of the design so that the total number of HBTs is limited, but there is no theoretical guarantee that a small cutsize would benefit the D2D wirelength. Hence, the most important task is to find how $W_{e,\mathrm{Bi}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ gets affected by $\boldsymbol{z}$ to evaluate the quality of partitioning. Considering that $W_{e,\mathrm{Bi}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is even discontinuous with respect to $\boldsymbol{z}$, the gradient $\nabla_{\boldsymbol{z}} W_{e,\mathrm{Bi}}$ does not exist at all. To tackle this problem, we leverage *finite difference* to approximate the impact of $\boldsymbol{z}$ on the bistratal wirelength.

Finite difference [49], [50], [51], [52] has been widely used in a large number of applications in numerical differentiation to approximate derivatives. We follow the definitions and notations in [52] and denote the *difference quotient* by:

$$\underset{h}{\Delta} f(x) = \frac{f(x+h) - f(x)}{h} \tag{32}$$

using the Nörlund's operator Δ_h [52], [53] for any function f on $\mathbb{R}$ and $x, h \in \mathbb{R}$. In the classical infinitesimal calculus, the first-order derivative of f is defined by $\lim_{h\to 0} \Delta_h f(x)$ if f is differentiable. Both difference and derivative estimate how the function value would change with its variables, but derivative is in a continuous view while difference depends on the step size h.

Taking a net $e \in E$ and $c_i \in e$, consider the impact of z_i to the bistratal wirelength $W_{e,\mathrm{Bi}}$. For simplicity, we use $W_{e,\mathrm{Bi}}(z_i)$ to represent the bistratal wirelength of net $e \in E$ as a function of z_i and fix all other variables. Given step size h, the difference quotient of $W_{e,\mathrm{Bi}}$ at z_i is

$$\underset{h}{\Delta} W_{e,\mathrm{Bi}}(z_i) = \frac{W_{e,\mathrm{Bi}}(z_i + h) - W_{e,\mathrm{Bi}}(z_i)}{h} \tag{33}$$

combining both the *forward/advancing difference* ($h > 0$) and the *backward/receding difference* ($h < 0$). Since $W_{e,\mathrm{Bi}}$ is discontinuous with respect to z_i, the limit $\lim_{h\to 0} \Delta_h W_{e,\mathrm{Bi}}(z_i)$ does not exist. However, we could consider (33) with a large h as we only have two dies. More specifically, we set $h = (1/4)z_{\max}$ if $\delta_i = P(z_i) = 0$ and $h = -(1/4)z_{\max}$ otherwise, so that the difference quotient in (33) will always be nonzero. Providing that $W_{e,\mathrm{Bi}}(z_i)$ is a step function that only takes two possible values $W_{e,\mathrm{Bi}}(0)$ and $W_{e,\mathrm{Bi}}([1/2]z_{\max})$, (33) can be summarized as follows.

Definition 7 (FDA): Consider the bistratal wirelength $W_{e,\mathrm{Bi}}$. The FDA $\nabla_{\boldsymbol{z}} W_{e,\mathrm{Bi}} = \boldsymbol{g}$ is defined by

$$\boldsymbol{g}_i = \underset{\frac{1}{4}z_{\max}}{\Delta} W_{e,\mathrm{Bi}}(z_i) = \frac{4}{z_{\max}} \left(W_{e,\mathrm{Bi}}\left(\frac{z_{\max}}{2}\right) - W_{e,\mathrm{Bi}}(0) \right) \tag{34}$$

where $z_{\max}$ is the total depth of our placement region, defined in (7).

Equation (34) is intuitive that it actually evaluates the wirelength change when moving a pin to the other die. It provides a local view of benefits we can obtain when changing node partition. Note that we still use the term $\nabla_{\boldsymbol{z}} W_{e,\mathrm{Bi}}$ to denote the FDA in Definition 7, although $W_{e,\mathrm{Bi}}$ itself is nondifferentiable.

From (34), any node $c_i \in V$ accumulates depth gradients $\nabla_{\boldsymbol{z}} W_{e,\mathrm{Bi}}$ from all related nets e, therefore the FDA locally evaluates the impact of every node to the total circuit wirelength. We apply $\sum_e \nabla_{\boldsymbol{z}} W_{e,\mathrm{Bi}}$ in (34) with cutsize gradient $\sum_e \nabla_{\boldsymbol{z}} p_{e,\mathrm{WA}}(\boldsymbol{z})$ and density gradient $\nabla_{\boldsymbol{z}} D(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ to numerical optimization in 3-D global placement to obtain a good partition with an acceptable number of HBTs. Combining with the adaptive planar gradients in Definition 6, we have defined the detailed gradient computation of the proposed bistratal wirelength model.

TABLE I
STATISTICS OF THE ICCAD 2022 CONTEST BENCHMARK SUITES [15] WHERE u^+, u^- STAND FOR THE UTILIZATION CONSTRAINTS ON THE TOP DIE AND THE BOTTOM DIE, RESPECTIVELY. RH^+ AND RH^- REPRESENT THE ROW HEIGHT ON THE TOP AND BOTTOM DIE. w' MEANS THE SIZE OF HYBRID BONDING TERMINALS

Bench.	#Nodes	#Nets	#Pins	u^+	u^-	RH^+	RH^-	w'
case2	2735	2644	8118	0.70	0.75	176	252	100
case2h	2735	2644	8118	0.79	0.79	252	252	114
case3	44764	44360	142246	0.78	0.78	115	115	50
case3h	44764	44360	142246	0.68	0.78	92	115	46
case4	220845	220071	773551	0.66	0.70	92	115	62
case4h	220845	220071	773551	0.66	0.76	103	115	66

VI. EXPERIMENTAL RESULTS

A. Experimental Setup

We conducted experiments on ICCAD 2022 contest benchmark suits [15]. The detailed design statistics are shown in Table I. Note that each HBT in a specific design has a size of $w' \times w'$, and the minimum spacing s' on x and y directions between each pair of HBTs is also equal to w'. Movable macros are not included in the benchmark suits.

We implemented the proposed 3-D analytical placement framework in C++ and CUDA based on the open-source placer DREAMPlace [39]. All the experiments were performed on a Linux machine with 20 Intel Xeon Silver 4210R cores (2.40GHz), 1 GeForce RTX 3090Ti graphics card, and 24 GB of main memory. We compared our framework with the SOTA placers from the top-3 teams in ICCAD 2022 contest and recent work [14], and the reported results were evaluated by the official evaluator provided by the contest.

B. Comparison With SOTA Placers

Table II shows the experimental results of the top-3 teams, SOTA analytical 3-D placer [14], and ours on the contest benchmark suites [15]. We compared the exact D2D wirelength (WL), the total number of HBTs, and runtime of each case with the baselines in Table II. The wirelength is evaluated using the provided official evaluator from the benchmark suites. For a fair comparison, we acquired their binary executable files and evaluated the end-to-end runtime of the baselines on our machine using their default settings.

It worths mentioning that the ICCAD 2022 Contest [15] evaluates WL as the final score. Hence, the contestants only target at optimizing WL and may not consider HBT costs explicitly. However, realistic requirements often expect to limit the HBT usage as well. Our framework explicitly considers the cutsize optimization as a secondary goal in the objective function in Definition 5 to find a "proper" cutsize, as simply reducing cutsize may also degrade the performance [54].

As illustrated in Table II, our analytical 3-D placement framework consistently obtained the best-WL results for all the cases, demonstrating the significant advantage of our 3-D placement paradigm with the dedicated bistratal wirelength model. Compared to the top-3 teams, our placer achieved 4.1%, 5.7%, and 7.2% shorter wirelength on average, respectively.

Thanks to the global optimization view of our 3-D analytical approach, our placer utilized fewer HBTs and achieved better wirelength. Our framework reduced 52.3%, 21.1%, 2.0% number of HBTs on average compared to the top-3 contest winners. Our framework achieved up to 49.2% HBT number reduction than the first place on the large cases, making our framework more competitive to reduce the HBT fabrication cost for large designs in real scenarios. Leveraging the computation power of modern GPUs, our placer demonstrates better-runtime scalability than the baselines, achieving 4.300× and 5.320× speedup over the first place and the second place for end-to-end placement, and achieving up to 2.925× speedup over the third place on the large cases.

We also compared our proposed framework with the SOTA analytical 3-D placer [14] on the same ICCAD 2022 benchmarks [15]. Chen et al. [14] proposed an MTWA wirelength model based on 3-D HPWL in Definition 1, considering heterogeneous technologies with a weight factor α that correlates positively with net degrees. They aimed to guide the optimizer to split more low-degree nets for wirelength reduction, which resulted in notable improvements compared to the first-place winner. However, their wirelength model in 3-D analytical placement is still inaccurate and thus requires an additional 2-D placement to refine node locations. Moreover, MTWA [14] is not directly partitioning-aware. The experimental results in Table II show that we achieve up to 3.4% wirelength improvement and 2.1% on average compared to [14]. Remarkably, we are confident enough of our placement framework in numerical optimization, and thus do not require a 2-D placement to further refine node locations after 3-D placement. In addition, we require 25% fewer HBTs and can efficiently accomplish the placement task with GPU resources. In modern VLSI design, the performance on large cases is most critical. We considered two large cases containing more than 220K standard cells in the ICCAD 2022 Contest benchmark suites [15]. As shown in Table II, we significantly outperform the baseline by 1.8% and 2.5% wirelength improvement on the largest two cases case4 and case4h, respectively, with more than 8× runtime acceleration, proving the scalability of our framework.

C. 3-D Global Placement Analysis

Our 3-D analytical placement framework enables the simultaneous node partitioning and placement in the global placement stage, forming a larger solution space than previous separate partitioning and placement works [6], [7], [21], [22]. Unlike previous 3-D analytical placer [13] targets on multiple tiers and leverages subsequent 2-D placement to refine the placement solution, our framework assigns the nodes to exact two dies and place them in a single run. Our 3-D global placement is visualized in Fig. 5.

In Fig. 5, fillers, nodes on the top die, and nodes on the bottom die are denoted by gray, brown, and blue rectangles, respectively. The node depth is omitted for better visualization. All standard cells are randomly initialized around the center point of the design from a normal distribution, shown in Fig. 5(a). Note that fillers are already inserted according

TABLE II
EXPERIMENTAL RESULTS ON THE ICCAD 2022 CONTEST BENCHMARKS [15] COMPARED TO THE TOP-3 WINNERS AND THE SOTA ANALYTICAL 3-D PLACER [14]. *WL* INDICATES THE *exact* D2D WIRELENGTH EVALUATED BY THE PROVIDED OFFICIAL EVALUATOR. *HBTs* REPRESENTS THE CUT SIZE, I.E., THE TOTAL NUMBER OF HYBRID BONDING TERMINALS. *RT* (S) STANDS FOR THE TOTAL RUNTIME

Bench.	1st Place			2nd Place			3rd Place			[14]			Ours		
	WL	HBTs	RT	WL	HBTs	RT	WL	HBTs	RT	WL	HBTs	RT	WL	HBTs	RT
case2	2072075	1131	47	2080647	477	7	2097487	163	5	2011447	784	33	**1944656**	646	38
case2h	2555461	1083	45	2735158	687	8	2644791	151	5	2514597	891	32	**2462553**	345	40
case3	30580336	16820	342	30969011	11257	234	33063568	14788	68	30302643	8169	141	**30062713**	8017	92
case3h	27650329	16414	224	27756492	8953	243	28372567	11211	63	27135602	7727	155	**26727327**	8887	93
case4	281315669	84069	1324	274026687	51480	1675	281378049	46468	391	272327370	53264	1189	**267400694**	42763	135
case4h	301193374	84728	1096	308359159	59896	2040	307399565	58860	427	296655075	49616	1190	**289245472**	47712	146
Avg.	1.041	2.096	4.300	1.057	1.267	5.320	1.072	1.019	1.249	1.021	1.328	3.639	1.000	1.000	1.000

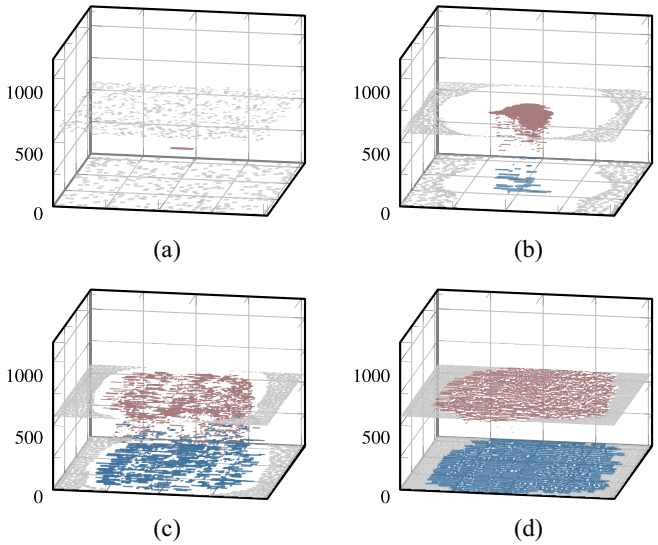


Fig. 5. 3-D global placement on case2 with heterogeneous technologies. Fillers, nodes on the top die, and nodes on the bottom die are denoted by gray, brown, and blue rectangles, respectively. Node depth is omitted for better visualization. Nodes are initialized at the center point, and the fillers are randomly distributed on the two dies as shown in (a). The 3-D density force combined with the wirelength force progressively drive all the nodes to the specific die, leading to a placement solution with almost perfect node partition as shown in (d). (a) Iteration 0, WL 4.20×10^5, cut size 1487, and overflow 0.96. (b) Iteration 800, WL 5.73×10^5, cut size 60, and overflow 0.91. (c) Iteration 1200, WL 1.48×10^6, cut size 652, and overflow 0.47. (d) Iteration 1769, WL 1.71×10^6, cut size 646, and overflow 0.07.

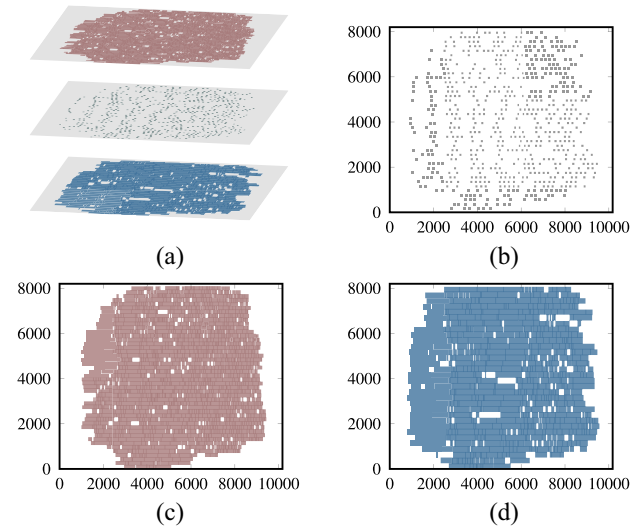


Fig. 6. (a) Placement results of case2 using our proposed method. (b) Illustrates how hybrid bonding terminals are placed to connect cells on different dies. (c) and (d) plot the top die placement and the bottom die placement, respectively.

to the given utilization requirements and uniformly initialized on two dies. During the 3-D global placement, the optimizer tends to move nodes according to the gradients of wirelength (including cutsize with weight α) and density. The tentative partition $\boldsymbol{\delta}$ is updated at every intermediate iteration of global placement, shown in Fig. 5(a) and (c), until the convergence is detected. At last, the placer will find a 3-D placement solution with optimized wirelength, shown in Fig. 5(d). When the convergence is attained, most standard cells c_i with coordinate z_i satisfying $|(2z_i/z_{\max}) - \delta_i| < \epsilon$ for a sufficiently small positive number $\epsilon > 0$, implying that our framework is confident enough to partition every standard cell. The 3-D global placement produces a solution with overflow 0.07, shown in Fig. 5(d), therefore we apply the tentative partition at the 1769th iteration as the final partition $\boldsymbol{\delta}$ and proceed to the later steps, including legalization and detailed placement.

In addition to the convergence visualization of our 3-D global placement in Fig. 5, we also plot the 2-D placement of two dies in Fig. 6, together with the hybrid bonding terminals. As shown in Fig. 6(b), our HBT placement is sparse after legalization and detailed placement.

D. Ablation Studies on Wirelength Models

We evaluated the effectiveness of our proposed bistratal wirelength model by using different wirelength models in our framework on the ICCAD 2022 Contest Benchmarks [15]. The detailed experimental results are shown in Table III.

Plain HPWL stands for the conventional HPWL model $\tilde{W}_e$ defined in Definition 4. It is integrated in 3-D HPWL adopted in many previous analytical placers [13], [31], [32], [34]. This wirelength model is very classical and has been proved to be effective in analytical 3-D placement. Notably, the gradients of differentiable approximations to $\tilde{W}_e$ w.r.t. $\boldsymbol{z}$ only focus on optimization on cutsize. Hence, it achieves the best results of cutsize, with only 55.5% HBTs of ours, shown in Table III. However, the wirelength reported by the evaluator is 16.9% larger than ours, as plain HPWL model could not comprehend the impact of partitioning on the exact

TABLE III
RESULTS OF ABLATION STUDY ON THE ICCAD 2022 CONTEST BENCHMARKS [15] USING DIFFERENT WIRELENGTH MODELS WITH THE SAME EXPERIMENTAL SETTINGS. *WL* INDICATES THE *exact* D2D WIRELENGTH EVALUATED BY THE PROVIDED OFFICIAL EVALUATOR. *HBTs* REPRESENTS THE CUT SIZE, I.E., THE TOTAL NUMBER OF HYBRID BONDING TERMINALS. BIHPWL IS THE BISTRATAL WIRELENGTH EQUIPPED WITH ADAPTIVE PLANAR GRADIENT IN DEFINITION 6. FDA INDICATES FDA OF DEPTH GRADIENTS IN DEFINITION 7

Bench.	Plain HPWL		BiHPWL w/o. FDA		Plain HPWL w/. FDA		BiHPWL w/. FDA	
	WL	HBTs	WL	HBTs	WL	HBTs	WL	HBTs
case2	2351813	459	2271554	454	2118450	708	**1944656**	646
case2h	2919815	236	2755549	245	3001905	441	**2462553**	345
case3	34776108	4396	33965431	4547	35577287	8086	**30062713**	8017
case3h	31093130	4770	30066866	4781	30748977	8544	**26727327**	8887
case4	309580785	19339	304667903	23261	288957440	51369	**267400694**	42763
case4h	330290736	18971	325610343	22195	324613980	54942	**289245272**	47712
Avg.	1.169	0.555	1.134	0.588	1.141	1.116	1.000	1.000

D2D wirelength. We now validate the effectiveness of the adaptive planar gradient defined in Definition 6 and the FDA of depth gradients in Definition 7.

BiHPWL in the second main column of Table III represents the bistratal wirelength in Definition 5 equipped with the adaptive planar gradient. "BiHPWL model without FDA" is equivalent to "plain HPWL with adaptive planar gradient" in terms of gradient computation. As shown in Table III, the BiHPWL model without FDA achieves 3.5% wirelength improvements on average with little degradation of cutsize, compared to plain HPWL. It is intuitively rational as the adaptive planar gradient tries to figure out when the plain HPWL is inaccurate compared to the exact D2D wirelength and switches a different strategy accordingly. However, it is still far inferior to the results with FDA, as the adaptive planar gradient in Definition 6 focuses on optimizations of planar coordinates $\boldsymbol{x}, \boldsymbol{y}$ without comprehension of partitioning.

In the third main column of Table III, the plain HPWL is equipped with FDA, which means that we use $\tilde{W}_e$ to replace $W_{e,\mathrm{Bi}}$ in Definition 7. However, the plain HPWL $\tilde{W}_e$ is irrelevant to $\boldsymbol{z}$ and thus insensitive to different partitioning. Therefore, nonzero gradients occur only because of changes of node attributes given heterogeneous technologies, resulting in less than 3% wirelength improvements with significant cutsize degradation. By contrast, BiHPWL is evidently sensitive to partitioning, leading to 13.4% wirelength improvement when FDA is enabled, as shown in the last main column in Table III. Note that we utilize much more resources of vertical interconnects to optimize wirelength versus plain HPWL, fully taking advantage of the benefits of F2F-bonded 3-D ICs. Meanwhile, our framework still significantly outperforms the first-place winner on cutsize, preserving advantages on wirelength.

E. Runtime Breakdown

Fig. 7 plots the runtime breakdown on case4h for our 3-D analytical placement framework. The GPU-accelerated 3-D global placement takes 82.59% of the total runtime, while the GPU-accelerated detailed placement takes 15.07%.

Similar to [39], the density and its gradients are computed with a GPU-accelerated implementation of 3-D FFT in the 3-D global placement. Given the ultrafast density computation, we set the number of bins $N_z = 32$ by default for

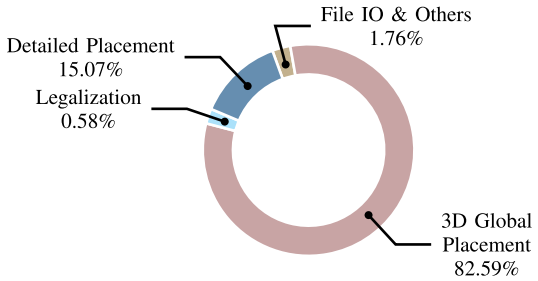


Fig. 7. Runtime breakdown of our proposed analytical 3-D placement framework on the ICCAD 2022 contest benchmark case4h [15]. Our global placement and detailed placement are both GPU-accelerated.

all nontrivial cases in [15] so that the discrete grids can model the 3-D electric field more precisely and thus produce better results. The proposed bistratal wirelength model is also implemented based on weighted-average [31], [32] with GPU-acceleration techniques in [39]. The computation of wirelength and density with their gradients take up the main part of runtime in global placement. It is worth mentioning that we can achieve 9.807× and 7.506× runtime speedup for the largest two designs case4 and case4h over the first-place winner, demonstrating that our placement framework is scalable.

VII. CONCLUSION

This article proposes a new analytical 3-D placement framework for F2F bonded 3-D ICs with heterogeneous technologies, incorporating a novel bistratal wirelength model. The proposed framework leverages high-performance GPU-accelerated implementations of both the wirelength model and the electrostatic-based density model. The experimental results on ICCAD 2022 Contest benchmarks demonstrate that our framework significantly surpass the first-place winner and the SOTA analytical 3-D placer by 4.1% and 2.1% on wirelength, respectively, with much fewer vertical interconnections and conspicuous acceleration. The 3-D placement framework accomplishes partitioning and placement in a single run, proving that true 3-D analytical placement can effectively handle partitioning with respect to wirelength optimization for F2F-bonded 3-D ICs and thus inspire more explorations and studies on 3-D analytical placement algorithms.

REFERENCES

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

[2] B. Munger et al., "'Zen 4': The AMD 5nm 5.7 GHz x86-64 microprocessor core," in *Proc. ISSCC*, 2023, pp. 38–39.

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

[4] P. Batude, T. Ernst, J. Arcamone, G. Arndt, P. Coudrain, and P.-E. Gaillardon, "3-D sequential integration: A key enabling technology for heterogeneous co-integration of new function with CMOS," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 2, no. 4, pp. 714–722, Dec. 2012.

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

[6] 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. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 10, pp. 1716–1724, Oct. 2017.

[7] 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 *Proc. ISPD*, 2018, pp. 90–97.

[8] P. Morrow, C.-M. Park, S. Ramanathan, M. J. Kobrinsky, and M. Harmes, "Three-dimensional wafer stacking via Cu-Cu bonding integrated with 65-nm strained-Si/low-*k* CMOS technology," *IEEE Electron Device Lett.*, vol. 27, no. 5, pp. 335–337, May 2006.

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

[10] S. A. Panth, K. Samadi, Y. Du, and S. K. Lim, "Design and CAD methodologies for low power gate-level monolithic 3D ICs," in *Proc. ISLPED*, 2014, pp. 171–176.

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

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

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

[14] 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 3D ICs with multiple manufacturing technologies," in *Proc. DAC*, 2023, pp. 1–2.

[15] 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: 3D placement with D2D vertical connections," in *Proc. ICCAD*, 2022, pp. 1–5.

[16] J. Cong, G. Luo, J. Wei, and Y. Zhang, "Thermal-aware 3D IC placement via transformation," in *Proc. ASPDAC*, 2007, pp. 780–785.

[17] Y. Deng and W. P. Maly, "Interconnect characteristics of 2.5-D system integration scheme," in *Proc. ISPD*, 2001, pp. 171–175.

[18] S. Das, A. Chandrakasan, and R. Reif, "Design tools for 3-D integrated circuits," in *Proc. ASPDAC*, 2003, pp. 53–56.

[19] B. Goplen and S. Spatnekar, "Placement of 3D ICs with thermal and interlayer via considerations," in *Proc. DAC*, 2007, pp. 626–631.

[20] D. H. Kim, K. Athikulwongse, and S. K. Lim, "A study of through-silicon-via impact on the 3D stacked IC layout," in *Proc. ICCAD*, 2009, pp. 674–680.

[21] K. Chang et al., "Cascade2D: A design-aware partitioning approach to monolithic 3D IC with 2D commercial tools," in *Proc. ICCAD*, 2016, pp. 1–8.

[22] H. Park, B. W. Ku, K. Chang, D. E. Shim, and S. K. Lim, "Pseudo-3D approaches for commercial-grade RTL-to-GDS tool flow targeting monolithic 3D ICs," in *Proc. ISPD*, 2020, pp. 47–54.

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

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

[25] Y.-C. Lu, S. S. K. Pentapati, L. Zhu, K. Samadi, and S. K. Lim, "TP-GNN: A graph neural network framework for tier partitioning in monolithic 3D ICs," in *Proc. DAC*, 2020, pp. 1–6.

[26] G. Murali, S. M. Shaji, A. Agnesina, G. Luo, and S. K. Lim, "ART-3D: Analytical 3D placement with reinforced parameter tuning for monolithic 3D ICs," in *Proc. ISPD*, 2022, pp. 97–104.

[27] I. Kaya, M. Olbrich, and E. Barke, "3-D placement considering vertical interconnects," in *Proc. SOCC*, 2003, pp. 257–258.

[28] R. Hentschke, G. Flach, F. Pinto, and R. Reis, "Quadratic placement for 3D circuits using z-cell shifting, 3D iterative refinement and simulated annealing," in *Proc. Symp. Integr. Circuits Syst. Design (SBCCI)*, 2006, pp. 220–225.

[29] T. Tanprasert, "An analytical 3-D placement that reserves routing space," in *Proc. ISCAS*, vol. 3, 2000, pp. 69–72.

[30] B. Goplen and S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," in *Proc. ICCAD*, 2003, pp. 86–89.

[31] M.-K. Hsu, Y.-W. Chang, and V. Balabanov, "TSV-aware analytical placement for 3D IC designs," in *Proc. DAC*, 2011, pp. 664–669.

[32] 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.

[33] A. B. Kahng and Q. Wang, "Implementation and extensibility of an analytic placer," in *Proc. ISPD*, 2004, pp. 18–25.

[34] 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.

[35] J. Lu et al., "FFTPL: An analytic placement algorithm using fast fourier transform for density equalization," in *Proc. ASICON*, 2013, pp. 1–4.

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

[37] 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.

[38] C.-K. Cheng, A. B. Kahng, I. Kang, and L. Wang, "RePlAce: Advancing solution quality and routability validation in global placement," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 9, pp. 1717–1730, Sep. 2018.

[39] 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.

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

[41] W. C. Naylor, R. Donelly, and L. Sha, "Non-linear optimization system and method for wire length and delay optimization for an automatic electric circuit placer," U.S. Patent 6 301 693, 2001.

[42] P. Liao, H. Liu, Y. Lin, B. Yu, and M. Wong, "On a Moreau envelope wirelength model for analytical global placement," in *Proc. DAC*, 2023, pp. 1–6.

[43] 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.

[44] 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.

[45] 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.

[46] M. Pan, N. Viswanathan, and C. Chu, "An efficient and effective detailed placement algorithm," in *Proc. ICCAD*, 2005, pp. 48–55.

[47] 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.

[48] 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.

[49] B. Taylor, *Methodus Incrementorum Directa & Inversa*. London, U.K.: Gul Innys, 1715.

[50] G. Boole and J. F. Moulton, *A Treatise on the Calculus of Finite Differences*. New York, NY, USA: Dover, 1872.

[51] C. Jordan, *Calculus of Finite Differences*. New York, NY, USA: Chelsea, 1956.

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

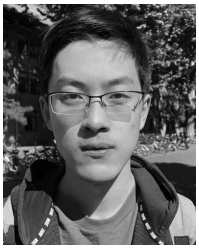
[53] N. E. Nörlund, *Vorlesungen über Differenzenrechnung*. Berlin, Germany: Springer, 1924.
[54] S. Pentapati and S. K. Lim, "Metal layer sharing: A routing optimization technique for monolithic 3D ICs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 30, no. 9, pp. 1355–1367, Sep. 2022.

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.

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.



Dawei Guo received the B.S. degree in computer science from the School of Electronics Engineering and Computer Science associated with the Center for Energy-Efficient Computing and Applications, Peking University, Beijing, China.

His research interests include algorithm in physical design and high performance computing.

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.

He is currently an Associate Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

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 has served as a TPC Chair of 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.