Late Breaking Results: Customized Diffusion Model Empowered by Heterogeneous Graph Network for Effective Floorplanning

Xinglin Zheng[1,*], Hao Gu[1,*], Keyu Peng[1], Youwen Wang[1], Wenxing Zhu[2], Ziran Zhu[1,†]
[1]National ASIC System Engineering Research Center, Southeast University, Nanjing, China
[2]Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou, China
Email: {220236395, 230218727, kypeng, 220236428}@seu.edu.cn, wxzhu@fzu.edu.cn, zrzhu@seu.edu.cn

2025 62nd ACM/IEEE Design Automation Conference (DAC) | 979-8-3315-0304-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/DAC63849.2025.11133070

***Abstract*—Floorplanning is a critical phase in VLSI physical design, focusing on determining block positions while optimizing wirelength under specified area constraints. However, classical analytical-based floorplanners are highly sensitive to the quality of initial solutions and existing learning-based methods often suffer from high computational inefficiency and complexity. In this paper, we propose a customized diffusion model to directly generate high-quality initial floorplans. By leveraging a classical analytical-based floorplanner on top of this initial floorplan, the final floorplanning results are significantly improved. To enhance feature extraction, a heterogeneous graph neural network (HGNN) is developed to explicitly incorporate block-to-block and pin-to-block relationships from the netlist during the diffusion process. Additionally, a novel guidance sampling function is introduced to optimize both wirelength and overlap, effectively reducing the required sampling steps while maintaining competitive initial solutions. Experimental results demonstrate that integrating our proposed diffusion model with an advanced analytical-based floorplanner achieves at least 4.8% reduction in runtime and 3.0% reduction in HPWL compared to the original floorplanner and other diffusion-based methods.**

I. Introduction

Floorplanning plays a critical role in modern chip design, directly impacting circuit performance, wirelength, and area utilization. Traditional heuristic-based methods efficiently explore the optimized solution space but suffer from high computational cost and excessive runtime. More recently, learning-based methods [1] may lead to computational inefficiency and complexity, making them impractical for designs with thousands of blocks.

Analytical-based floorplanners [2] are efficient and scalable while highly sensitive to the quality of initial solutions. Conventional initialization strategies, such as fixed-center, random or quadratic programming (QP)-based floorplanning, are commonly employed due to their computational efficiency. However, they often fail to provide stable and high-quality initial solutions, leading to suboptimal floorplanning results. [3] is the first work to apply diffusion models for placement, demonstrating their potential in generating initial solutions. However, [3] fails to effectively capture relationships between movable blocks and fixed pins from the netlist, leading to a poor solution quality.

To address these limitations, we propose a customized diffusion model that leverages a heterogeneous graph neural network (HGNN) to capture block-to-block and pin-to-block relationships. Additionally, a guidance sampling function is introduced to optimize both wirelength and overlap, significantly reducing the required sampling steps while maintaining competitive initial solutions.

II. The Proposed Method

Fig. 1 illustrates our proposed method, comprising three stages: diffusion (forward), denoising (reverse), and floorplanning process. During the diffusion process, the model progressively injects noise into existing optimized floorplanning datasets to train a noise prediction model. In the denoising process, a specifically designed sampling function guides this model to reconstruct and refine the initial floorplan. Finally, analytical-based floorplanner [2] further optimizes the floorplan to achieve optimized results.

This work was partially supported by the National Key Research and Development Program of China under Grant 2022YFB4400500, and the National Natural Science Foundation of China under Grant 92373207. (*Xinglin Zheng and Hao Gu contributed equally to this work.) (†Corresponding author: Ziran Zhu.)

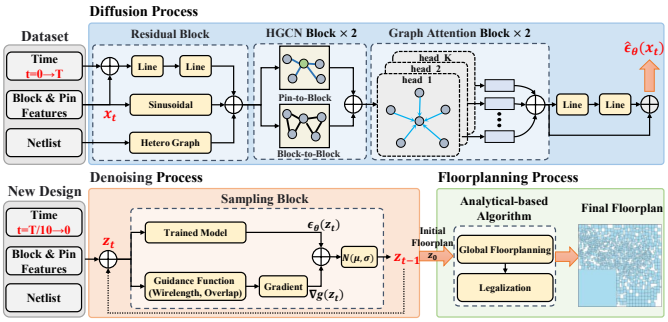


Fig. 1. Overview of our proposed method.

A. Diffusion Process

1) Heterogeneous Graph Construction: Firstly, we construct a heterogeneous graph from the optimized floorplanning dataset D to capture relationships between movable blocks (V_b) and fixed pins (V_p). This graph $G_{\text{Hetero}} = (V_b, V_p, E_{bb}, E_{pb})$ consists of two node types, with edge sets E_{bb} and E_{pb} representing their respective connections. For block-to-block (b2b) connections: an edge $E_{bb}^{i,j}$ is added between two movable blocks V_b^i and V_b^j if they share a net. Each block node is represented as $[x, y, w, h, A]$, where x, y denotes position, w, h represents width and height, and A is the block area. The edge set E_{bb} is characterized as $[x_i, y_i, x_j, y_j]$, where (x_i, y_i) and (x_j, y_j) denote the block coordinates. For pin-to-block (p2b) connections: an edge $E_{pb}^{i,j}$ is added between a fixed pin V_p^i and a block V_b^j if they share a net. Each pin node follows a simplified representation $[x, y, 0, 0, 0]$, and edge E_{pb} is also characterized as $[x_i, y_i, x_j, y_j]$. By explicitly modeling these relationships, our heterogeneous graph effectively captures b2b and p2b dependencies, enhancing the feature extraction of our diffusion model.

2) Noise Prediction Network: As illustrated in Fig. 1, noise prediction network f_θ is core component of diffusion process. It consists of three parts: residual, heterogeneous graph convolutional network (HGCN) and graph attention blocks. This network receives three inputs: t, node (block & pin) features and G_{Hetero}. First, noisy representation x_t undergoes sinusoidal encoding to enhance temporal representation, and t is embedded into x_t within a residual block to enrich feature expressiveness. Further, two HGCN layers perform message passing between nodes and edges, learning and updating node features as defined in II-A1. After each HGCN layer, pin node features are aggregated into block node features. Two independent graph attention blocks further refine node embeddings, capturing spatial structure and netlist information. Finally, the output will pass two linear layers with residual connections to get the final predicted noise $\hat{\epsilon}_\theta(x_t, t)$.

3) Model Training: We adopt DDPM [4] to train our proposed diffusion model, as detailed in Lines 1-7 of **Algorithm 1**. In Line 3, Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ is progressively injected into floorplan sample x_0 drawn from D according to randomly sampled diffusion timestep $t \sim \text{Uniform}(1, T)$. Line 4 defined the forward diffusion process of gradually perturbing x_0 to generate a noisy representation x_t. $\overline{\beta}_t = \prod_{i=1}^{t} \beta_i$ represents the cumulative noise schedule. In Line 5, noise prediction network f_θ detailed in Section II-A2, learns to estimate the added noise $\hat{\epsilon}_\theta(x_t, t)$, and its performance is optimized

Authorized licensed use limited to: University of Texas at Austin. Downloaded on September 18,2025 at 19:31:23 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Customized Diffusion Model for Floorplanning.

Input : Dataset D, max timestep T, noise schedules $\{\beta_t\}_{t=1}^{T}$, and $\{\gamma_t\}_{t=1}^{T}$, noise prediction model f_θ, total epoch number M, batch size N
Output: Trained model f_θ and the initial floorplan z_0

// **Diffusion Process**
1 **for** $i \leftarrow 1$ *to* M **do**
2 **for** $j \leftarrow 1$ *to* N **do**
3 Sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, floorplan $x_0 \sim D$, timestep $t \sim \text{Uniform}(1, T)$;
4 Compute $x_t \leftarrow \sqrt{\overline{\beta}_t} x_0 + \sqrt{1 - \overline{\beta}_t} \cdot \epsilon$;
5 Estimate the added noise: $\hat{\epsilon}_\theta(x_t, t) \leftarrow f_\theta(x_t, t, G_{\text{Hetero}})$;
6 Compute loss: $L(\theta) \leftarrow \|\epsilon - \hat{\epsilon}_\theta(x_t, t)\|^2$;
7 Update model parameters θ using Adam optimizer;

// **Denoising Process**
8 Sample initial random noise z_t;
9 **for** $t \leftarrow T/10$ *to* 1 **do**
10 Compute guidance sampling function $g(z_t)$ defined in Eq. (1);
11 Perform noise correction: $\epsilon_\theta(z_t, t) \leftarrow f_\theta(z_t, t, G_{\text{Hetero}}) + \sqrt{1 - \beta_t} \nabla_{z_t} g(z_t)$;
12 Compute: $z_{t-1} \leftarrow \frac{1}{\sqrt{\beta_t}} \cdot z_t + \gamma_t \cdot \epsilon_\theta(z_t, t)$;
13 **return** f_θ, z_0

by minimizing the loss function in Line 6. Line 7 updates the model parameters after each training iteration. By reducing the error between the estimated and actual noise, the diffusion model learns to reconstruct the original floorplan from noise.

B. Denoising Process

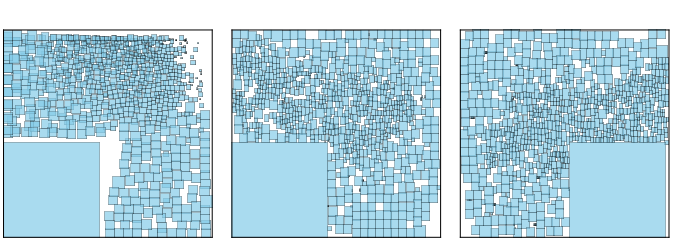
Our denoising process starts with Gaussian noise z_t and iteratively refines it over $T/10$ denoising steps to reconstruct the initial floorplan z_0, as detailed in Lines 8-13 of **Algorithm 1**. In Line 8, z_t is sampled from a standard normal distribution. In Line 9, the denoising process is conducted over $T/10$ steps, significantly reducing computational cost. To further enhance floorplan quality, a guidance sampling function is computed in Line 10. Additionally, we eliminate the injected noise and adopt a deterministic denoising approach in Line 11, allowing the total number of denoising steps to be reduced from T to $T/10$ without compromising output quality. Noise correction is also applied to refine the predicted noise, which will be detailed in Section II-B. Finally, the updated estimate z_{t-1} is computed in Line 12. After multiple denoising steps, the process reconstructs the initial floorplan z_0.

To improve initial floorplan quality, we incorporate a guidance sampling function $g(z_t)$ to optimize wirelength and overlap during the denoising process. Our method corrects the noise $\epsilon_\theta(z_t, t)$, as detailed in Line 11 of **Algorithm 1**. This correction guides the diffusion model to generate initial floorplans with minimized wirelength and reduced overlap. $g(z_t)$ is formulated as a weighted sum of wirelength $\mathcal{W}(z_t)$ and overlap $\mathcal{O}(z_t)$:

$$g(z_t) = \lambda_w \cdot \mathcal{W}(z_t) + \lambda_o \cdot \mathcal{O}(z_t), \tag{1}$$

where λ_w and λ_o are user-defined balance factors. $\mathcal{W}(z_t)$ represents the total estimated half-perimeter wirelength (HPWL) within the floorplan. $\mathcal{O}(z_t)$ quantifies the undesirable block overlaps:

$$\mathcal{O}(z_t) = \sum_{i,j \in S} \min\left(0, \max\left(H_o^{ij}(z_t), V_o^{ij}(z_t)\right)\right)^2, \tag{2}$$



(a) Initial floorplan (ours) (b) Final floorplan (ours) (c) Final floorplan (QP)

Fig. 2. Visualization of different floorplanning results on HB+01.

TABLE I
PERFORMANCE COMPARISON ON THE HB+ BENCHMARKS

Name	Blocks	Pins	QP-ICCAD'23[2] HPWL	QP-ICCAD'23[2] Time	Diff[3]-ICCAD'23 HPWL	Diff[3]-ICCAD'23 Time	Ours HPWL	Ours Time
HB+01	911	246	2.63 E+06	6.2 s	2.51 E+06	5.2 s	**2.41 E+06**	**4.7 s**
HB+02	1471	259	5.42 E+06	5.2 s	5.37 E+06	5.0 s	**5.09 E+06**	**4.5 s**
HB+03	1289	283	8.27 E+06	6.3 s	8.06 E+06	6.4 s	**7.39 E+06**	**6.2 s**
HB+04	1584	287	8.91 E+06	6.9 s	8.78 E+06	6.5 s	**8.42 E+06**	**6.2 s**
HB+06	749	166	7.61 E+06	6.4 s	7.88 E+06	6.8 s	**7.15 E+06**	**6.4 s**
HB+07	1120	287	1.39 E+07	7.7 s	1.37 E+07	7.6 s	**1.34 E+07**	**7.2 s**
HB+08	1269	286	1.67 E+07	10.9 s	1.60 E+07	10.7 s	**1.59 E+07**	**10.7 s**
HB+09	1113	285	1.48 E+07	9.1 s	1.44 E+07	8.1 s	**1.42 E+07**	**7.6 s**
HB+10	1595	744	3.84 E+07	16.2 s	3.72 E+07	16.4 s	**3.68 E+07**	**16.2 s**
HB+11	1497	406	2.39 E+07	11.7 s	2.34 E+07	10.0 s	**2.29 E+07**	**9.6 s**
HB+12	1233	637	4.72 E+07	10.6 s	4.62 E+07	9.7 s	**4.51 E+07**	**9.3 s**
HB+13	1254	490	3.13 E+07	12.4 s	3.10 E+07	12.1 s	**3.01 E+07**	**12.1 s**
HB+14	1635	517	6.06 E+07	16.3 s	5.69 E+07	16.0 s	**5.65 E+07**	**15.6 s**
HB+15	1412	383	7.23 E+07	27.6 s	7.06 E+07	26.6 s	**6.86 E+07**	**22.8 s**
HB+16	1091	504	8.87 E+07	25.4 s	8.62 E+07	25.0 s	**8.46 E+07**	**24.3 s**
HB+17	1442	743	**1.37 E+08**	14.4 s	1.38 E+08	14.3 s	1.39 E+08	**14.1 s**
HB+18	943	272	6.69 E+07	7.2 s	6.66 E+07	6.5 s	**6.66 E+07**	**6.5 s**
Average Ratio			1.051	1.100	1.030	1.048	**1.000**	**1.000**

where S denotes the set of block pairs, $H_o^{ij}(z_t)$ and $V_o^{ij}(z_t)$ represent the horizontal and vertical overlap measures between blocks V_b^i and V_b^j. By computing the gradient of $g(z_t)$ and correcting the predicted noise over multiple denoising steps, we can ultimately achieve the initial floorplan z_0.

C. Floorplanning Process

Finally, the analytical-based floorplanner further refines the initial floorplan to achieve optimized results. As illustrated in Fig. 1, the generated z_0 is used to replace QP stage in initial floorplanning of [2]. Subsequent global floorplanning and legalization are then performed, ultimately producing the final optimized floorplan.

III. EXPERIMENTAL RESULTS

The proposed diffusion model is trained on $FloorSet$ [5] and evaluated on the HB+ benchmark [2], including fixed pins, movable hard and soft blocks. Detailed statistics are provided in Table I. For ease of processing, we assume soft blocks have equal length and width when generating initial floorplan. Our method is implemented in $C++$ and $Python$ and runs on a $Ubuntu$ 20.04 $Linux$ machine equipped with two 64-core CPUs and five 4090 GPUs. T is set to 1000.

To evaluate performance, we integrate our model with advanced analytical-based floorplanner ICCAD'23 [2] and compare it against original QP-ICCAD'23 [2], as well as ICCAD'23 with diffusion [3] initialization (Diff-ICCAD'23). As [1] exhibits excessive runtime or fails to generate initial floorplans, we exclude it from the comparison. As shown in Table I, ours achieves average improvements of 10.0% and 4.8% in floorplanning runtime, and 5.1% and 3.0% in HPWL, compared to QP-ICCAD'23 and Diff-ICCAD'23. These results indicate that our trained diffusion model can efficiently generate high-quality initial floorplans, leading to significant improvements when integrated with classical analytical-based floorplanner. Besides, Fig. 2 compares floorplanning results. Fig. 2(a) shows the initial floorplan generated by our diffusion model, Fig. 2(b) presents the final floorplan of ours, and Fig. 2(c) depicts the final floorplan of original QP-ICCAD'23.

REFERENCES

[1] Wenbo Guan, Xiaoyan Tang, Hongliang Lu, Jingru Tan, Jinlong Wang, Yuming Zhang, and Yimen Zhang. LSTM-characterized approach for chip floorplanning: Leveraging HyperGCN and DRQN. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 44(2):709–722, 2024.

[2] Fuxing Huang, Duanxiang Liu, Xingquan Li, Bei Yu, and Wenxing Zhu. Handling orientation and aspect ratio of modules in electrostatics-based large scale fixed-outline floorplanning. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–9, 2023.

[3] Vint Lee, Chun Deng, Leena Elzeiny, Pieter Abbeel, and John Wawrzynek. Chip placement with diffusion. *arXiv preprint arXiv:2407.12282*, 2024.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33(574):6840–6851, 2020.

[5] Uday Mallappa, Hesham Mostafa, Mikhail Galkin, Mariano Phielipp, and Somdeb Majumdar. FloorSet-a vlsi floorplanning dataset with design constraints of real-world socs. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–9, 2024.

Authorized licensed use limited to: University of Texas at Austin. Downloaded on September 18,2025 at 19:31:23 UTC from IEEE Xplore. Restrictions apply.