

SACPlace: Multi-Agent Deep Reinforcement Learning for Symmetry-Aware Analog Circuit Placement

Lei Cai^{2†}, Guojing Ge^{1,4†}, Guibo Zhu^{1,4,5*}, Jixin Zhang^{3*}, Jinqiao Wang^{1,4,5}, Bowen Jia² and Ning Xu²

¹Foundation Model Research Center, Institute of Automation, Chinese Academy of Sciences

²School of Computer Science and Artificial Intelligence, School of Information Engineering, Wuhan University of Technology

³School of Computer Science, Hubei University of Technology,⁴Wuhan AI Research,⁵University of Chinese Academy of Sciences

Abstract—The placement of analog Integrated Circuits (ICs) plays a critical role in their physical design. The objective is to minimize the Half-Perimeter Wire Length (HPWL) while satisfying complex analog IC constraints, such as symmetry. Unlike digital ICs, analog ICs are highly sensitive to parasitic effects, making device symmetry crucial for optimal circuit performance. However, existing methods, including both machine learning-based and analytical approaches, struggle to meet strict symmetry constraints. In machine learning-based methods, training a general model is challenging due to the limited diversity of the training data. In analytical methods, the difficulty lies in formulating symmetry constraints as a convex function, which is necessary for gradient-based optimization of the placement. To address the issue, we formulate the placement process as a Markov decision process and propose SACPlace, a multi-agent deep reinforcement learning method for Symmetry-Aware analog Circuit Placement. SACPlace initially extracts layout information and various constraints as the input information for placement refinement and evaluation. Subsequently, SACPlace constructs multi-agent policy networks for symmetry-aware placement by refining placement guided by the evaluation of optimal symmetry quality. Following this, SACPlace constructs multi-layer perceptron-based critic networks to embed placement information for evaluating symmetry quality. This evaluation reward will be used for guiding placement refinement. Experimental results from four public analog ICs instances demonstrate that our method achieves the lowest actual wirelength and area while fully satisfying symmetry and common constraints, outperforming state-of-the-art methods. Additionally, simulation results on real-world analog ICs show better performance than these methods and even manual designs.

Index Terms—Analog integrated circuit placement, Multi-agent deep reinforcement learning, Symmetry-Aware

I. INTRODUCTION

The placement of analog ICs is pivotal in analog design, yet the predominant method still relies heavily on manual execution. As the complexity and scale of analog ICs grow, design rules become more intricate, making optimization significantly more challenging and time-consuming, while also increasing the likelihood of errors. Therefore, automated placement techniques

that incorporate analog IC constraints are crucial for enhancing both design efficiency and quality.

Unlike digital ICs placement, as explored by many researchers, the automation of analog ICs placement is significantly hindered by complex constraints such as symmetry and matching. These constraints, combined with the varying design requirements across different technology nodes, make it challenging to fully automate. Contemporary methods are chiefly classified into two principal categories: analytical methods [1]–[4] and machine learning-based approaches [5]–[9]. However, both current analytical and machine learning-based placement methods struggle with satisfying constraints of analog IC placement. Analytical methods face difficulties in meeting strict symmetry constraints, which otherwise result in increased HPWL and layout area [1]. Machine learning approaches also face challenges in training models to generalize effectively for satisfying placement constraints, particularly when technology nodes change. Although previous studies have been conducted for circuit performance optimization, the existing literature faces a significant challenge: there is no uniform framework that addresses complex constraints, such as matching, particularly symmetry constraints, within a comprehensive analog IC placement framework.

To address the above problem, our solution focuses on transforming symmetry-aware placement optimization into a gradient-based optimization process, allowing for more efficient and precise refinement of placement outcomes.

- One of the key challenges is formulating symmetry-aware placement. To tackle this, we propose leveraging deep neural networks to encode the global information of the layout and netlist, as well as the local information with devices, enabling the network to provide gradient information that guides placement optimization.
- Another challenge lies in effectively utilizing gradient information while avoiding local optima. To address this, we introduce cumulative rewards from Markov chains incorporating both gradient information and sparse rewards in analog IC placement to optimize the placement process.

In this paper, we introduce SACPlace, a multi-agent reinforcement learning framework specifically designed for symmetry-

[†] Equal Contribution.

* Corresponding authors: Guibo Zhu {gbzhu@nlpr.ia.ac.cn}, Jixin Zhang {zhangjx@hbut.edu.cn}

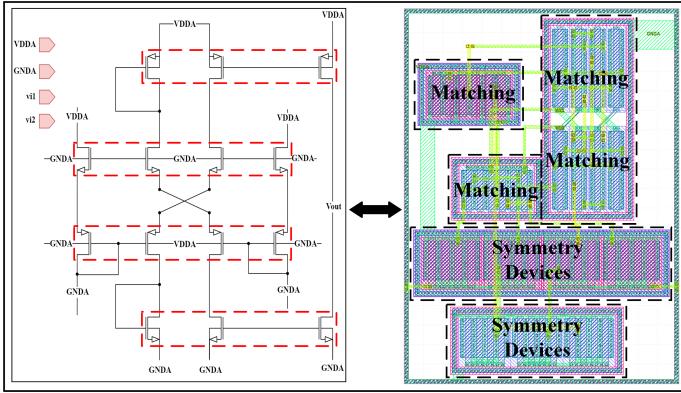


Fig. 1: An operational amplifier circuit with schematic and expert’s layout. The red dashed line shows the current mirror and differential pair, requiring symmetry and matching.

aware analog IC placement. We model the placement process as a Markov Decision Process (MDP). Firstly, we leverage multi-agent policy networks to encode layout information and complex constraints, providing gradient feedback for symmetry-aware placement refinement, and multi-layer perceptron-based critic networks to evaluate placement quality. Then, we develop a cumulative reward-based method using Markov chains, which incorporates both gradient information and sparse rewards to improve placement efficiency and avoid local optima while satisfying complex constraints. Our contributions are as follows:

- We propose SACPlace, which explicitly models various devices while incorporating multiple constraints such as symmetry and matching during the placement process.
- We develop a novel approach to address the challenge of symmetry-aware placement, encoding both global information and local information to provide critical gradient insights that guide placement optimization.
- We introduce a cumulative reward-based method incorporating both gradient information and sparse rewards in analog IC placement, enabling better optimization of complex constraints and avoiding local optima.
- To further improve the efficiency of placement, we utilize a multi-agent collaboration strategy along with the integration of a symmetrical action space within the DRL formulation.
- We conducted experiments on real-world analog IC instances and open-source datasets, achieving substantial improvements in both simulation results and HPWL, while fully satisfying symmetry and common constraints. This significantly enhances efficiency compared to other state-of-the-art (SOTA) methods.

II. RELATED WORKS

A. Analytical-based analog IC placement

Some analytical and optimization-based methodologies have been proposed for analog IC placement. Ou et al. (2016) introduced a layout-dependent, effects-aware placement approach [3], optimizing device placement by considering layout-induced effects. Xu et al. (2019) developed a device layer-aware placement method [2] that improves accuracy by incorporating

device layer impacts on circuit performance. Liu et al. (2020) proposed a Bayesian optimization-assisted hierarchical layout synthesis technique [10]. Chen et al. (2021) proposed MAGICAL [11], transforming the placement problem into a multi-objective optimization problem to optimize wire length and area. Lin et al. (2022) proposed ePlace-A [1], extending ePlace [12] with performance metrics from a pre-trained graph neural network to improve analog layout quality. Analytical methods reduce invalid searches and achieve rapid convergence, but face challenges in modeling symmetrical devices, affecting optimization gradients, and rely on pre-trained models for quality prediction.

B. Machine learning-based analog IC placement

Certain machine learning methods automate analog IC placement. Guerra et al. (2019) applied artificial neural networks for placement [13], while Li et al. (2020) introduced a customized graph neural network [8]. Gusmão et al. (2020, 2021) developed semi-supervised learning and DeepPlacer for OpAmp placement [6], [7]. Lu et al. (2023) used Bayesian optimization and autoencoders for automatic OpAmp generation [5]. Integrating DRL with placement methods addresses challenges without data scarcity. Ahmadi et al. (2021) proposed deep Q-Networks for automated FinFET placement [14]. Basso et al. (2024) used proximal policy optimization to manage topological constraints [9], optimizing area and HPWL. However, these DRL methods often overlook device symmetry. In general, training deep models for analog IC placement still remains challenging, especially when faced with the lack of diversity in existing training data, which restricts the models’ ability to generalize to placements with more complex constraints.

III. PROBLEM FORMULATION

Problem I. Given a *Netlist* which consists of nets $\mathbf{N} = \{net_i \mid 1 \leq i \leq |\mathbf{N}|\}$ and n devices $\mathbf{D} = \{D_i \mid 1 \leq i \leq n\}$, a set of constraints $\mathbf{C} = \{C_i \mid 1 \leq i \leq |\mathbf{C}|\}$, generate a placement for each device $D_i \in \mathbf{D}$ considering $C_j \in \mathbf{C}$ such that all devices are placed without any violations of constraints.

A. Optimization Objective

In our analog IC placement method, the objective is to minimize the optimization function defined in:

$$\begin{aligned} & \min \text{Area} + \text{Wirelength}, \\ & \text{s.t. } \mathbf{C} \end{aligned} \quad (1)$$

where *Area* is the total area of the analog IC, and *Wirelength* represents the sum of the HPWL calculated for the corresponding nets of the components. \mathbf{C} is the constraints for analog IC placement. The details of \mathbf{C} are shown in Section III-B.

B. Constraints

Device symmetry in analog ICs is critical, particularly due to the presence of current mirrors and differential pairs. In analog IC placement, specific device groups must follow symmetry and matching principles to ensure proper functionality. Violating these principles can result in significant parasitic effects, as shown in Fig. 1.

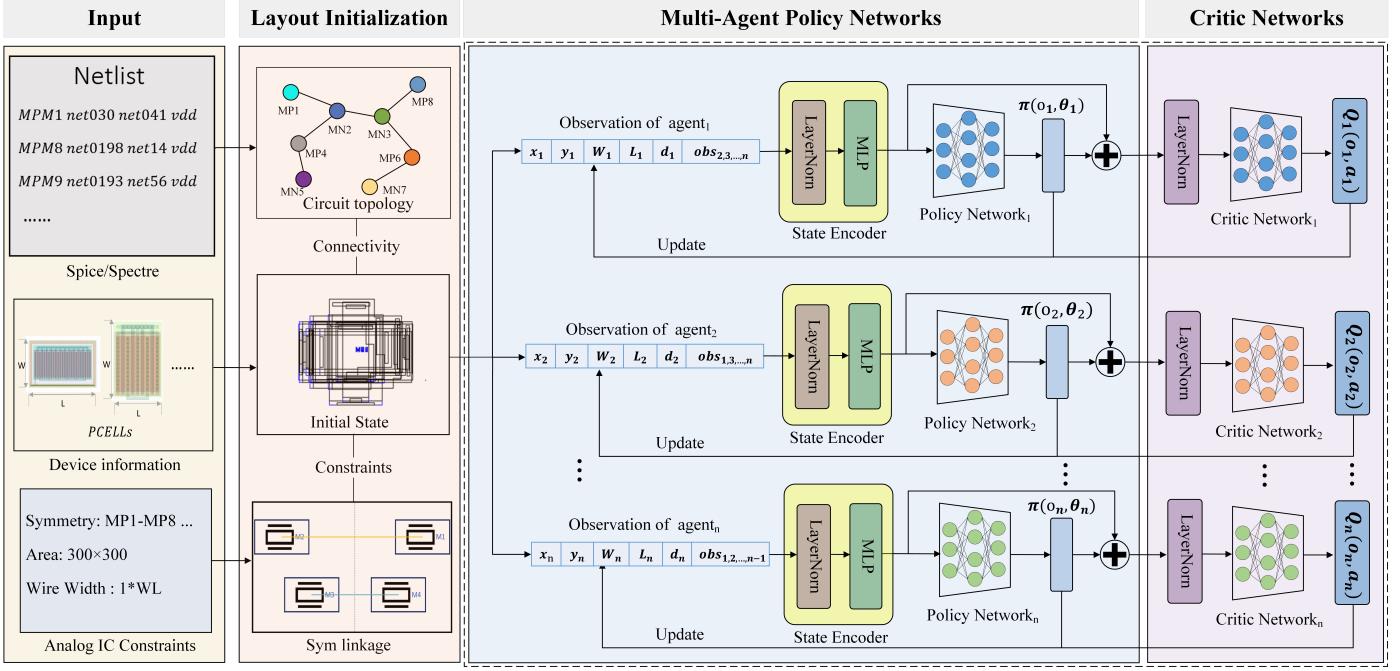


Fig. 2: Network architecture and placement process in SACPlace. Multi-Agent Policy Networks use state encoder to extract the feature of the layout and constraints, then refine the placement. Critic Networks evaluate the quality of the placement with symmetry. Each agent refines placement based on rewards from the environment and Q-values from the critic network.

Placement Constraints. The placement of devices \mathbf{D} must satisfy the following constraints \mathbf{C} :

- C_1 : **Symmetry Constraint** - Symmetric devices must be placed symmetrically to a designated axis.
- C_2 : **Proximity Constraint** - Certain devices must be within a specified distance to minimize parasitic effects.
- C_3 : **Matching Constraint** - Matched devices, like current mirrors or differential pairs, must be placed close together with similar surroundings for electrical matching.
- C_4 : **Overlap Constraint** - No two devices should overlap, ensuring each occupies a unique region.
- C_5 : **Boundary Constraint** - All devices must fit within the chip boundary, adhering to edge spacing rules.
- C_6 : **DRC Constraint** - Devices must meet spacing requirements under different technology nodes; any placement violating DRC will not be taped out.

IV. METHODOLOGY

A. Overview of Our Method

We propose a symmetry-aware multi-agent deep reinforcement learning framework, SACPlace. The framework begins by initializing the layout and incorporating both global and local information into the multi-agent policy networks. These networks extract layout details and complex constraints to evaluate and refine the placement. At each timestep, the policy networks generate optimal actions, while multi-layer perceptron-based critic networks assess the quality of these actions through Q-value outputs. The policy networks collaborate with critic networks to update the placement and produce gradient information. We further develop a cumulative reward-based

method that leverages this gradient information to optimize the placement process. The detailed architecture is shown in Fig. 2.

B. Input Information and Initialization

In this study, each device to be placed is treated as an individual agent, with its width and length denoted as $\mathbf{W} = \{W_i \mid 1 \leq i \leq n\}$ and $\mathbf{L} = \{L_i \mid 1 \leq i \leq n\}$, respectively. The circuit topology is derived from a given netlist N , used to compute the HPWL based on the connectivity between devices. Constraints include symmetry requirements alongside other common design constraints, and we add these constraints to the DRL environment. Initially, all devices are intentionally overlapped to facilitate the forward progress of DRL optimization.

C. Symmetry-Aware Multi-Agent DRL Definition

The state encoding of the i -th agent can be represented as $o_i = (x_i, y_i, W_i, L_i, d_i)$, where x_i and y_i denote the agent's coordinates, W_i and L_i represent the agent's width and length, and d_i signifies the distance between the current agent and other agents. The environmental information includes constraints \mathbf{C} that limit agent movement in the analog IC placement. The key constraint is **Symmetry Constraint** (C_1). Devices with symmetric pairs are considered a set. For devices $S = \{D_1, D_2, \dots, D_m\}$ requiring symmetry about a predefined axis, the action space \mathcal{A} ensures symmetrical actions on device pairs D_i and D_j .

a) Symmetrical Action Representation:

The symmetry constraint with respect to the y -axis can be formulated as:

$$x_j = 2x_{\text{sym}} - x_i, \quad y_j = y_i, \quad (2)$$

and this ensures that any movement of device D_i is mirrored by device D_j across the symmetry axis.

To formalize this within SACPlace, we define the action space \mathcal{A}_{sym} such that any action $a_i \in \mathcal{A}$ applied to a device D_i has a corresponding action a_j applied to its symmetric counterpart D_j . Therefore, for any action $a_i = (dx_i, dy_i)$, we enforce:

$$a_j = (dx_j, dy_j) = (-dx_i, dy_i). \quad (3)$$

This constrained action space reduces the dimensionality of the problem while ensuring that symmetry is maintained throughout the placement process.

b) Symmetrical Reward in DRL:

For the feedback calculation of symmetry constraints, we employ the following equation:

$$\text{Sym} = \sum_{D_i \in S} (|x_i - x_c| + |y_i - y_c|), \quad (4)$$

where x_c and y_c represent the coordinates of the symmetry center. The reward function for each agent to be minimized is defined as follows:

$$\text{Reward}_i = \sum_{D_i \in D} (\alpha Wl_i + \beta O_i + \mu \text{Sym}_i + \delta \text{Area}_i), \quad (5)$$

where Wl_i represents the normalized total wire length, estimated using the HPWL method, and O_i represents the overlap between agents. Area_i refers to the normalized utilization of the area. And $\alpha, \beta, \mu, \delta$ represent the weights of each variable, respectively.

Furthermore, to address the violation of the other common constraints ($C_2 - C_6$), a penalty is incorporated into the final reward, thereby compelling SACPlace to adhere to the established common constraints. The penalty is defined as:

$$P_i = c_2 P(p_i) + c_3 M(p_i) + c_4 O(p_i) + c_5 B(p_i) + c_6 D(p_i), \quad (6)$$

where $c_2 - c_6$ are coefficients, which are the weights of different constraints. The final sum of our symmetry-aware DRL rewards is calculated as:

$$r = \sum_{D_i \in D} (\text{Reward}_i - P_i) \quad (7)$$

where r is used for multi-agent policy networks to refine placement and multi-layer perceptron-based critic networks to evaluate the quality of layout.

D. Multi-Agent Policy Networks

To formalize placement constraints, we use a state encoder to extract features. The encoder includes Layer Normalization (LN) followed by a Multi-Layer Perceptron (MLP) with three submodules. Each submodule has a linear layer, a ReLU activation, and a LN layer. As shown in Fig. 3, we apply a concat function to concatenate the state encodings of all agents as input to the state encoder, enabling agents to use the global state of other agents and achieve collaborative placement:

$$s = \text{concat}(o_1, o_2, \dots, o_n), \quad (8)$$

$$\mathbf{h}_i = \text{MLP}(\text{LN}(s)), \quad (9)$$

where \mathbf{h}_i represents the hidden vector generated from the state encoder. The policy network we used is implemented as a three-layer MLP with ReLU, to express agent actions through gradient

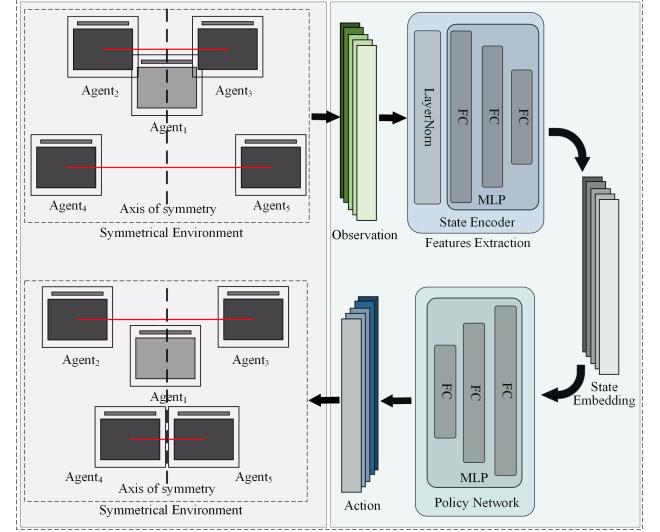


Fig. 3: Symmetry-Constrained Action Variation

optimization and map the \mathbf{h}_i into a probability distribution for actions:

$$\pi(a|s) = \text{softmax}(\text{MLP}(\mathbf{h}_i)). \quad (10)$$

The policy $\pi(a|s)$ selecting actions a based on state s must ensure symmetrical placements for all device pairs:

$$\pi_{\text{sym}}(a|s) = \prod_{(i,j) \in \text{sym}} \pi(a_i|s) \times \pi(a_j|s), \quad (11)$$

and this ensures that the placement decisions are not only optimizing for objectives but also maintaining symmetry throughout the placement process. The rewards can be estimated by computing discounted cumulative reward as MDP [15]:

$$R_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}, \quad (12)$$

where t is the timestep in $[0, T]$, and R_T can be regarded as the final reward at the end of the episode, γ is a discount factor. The method we use to update the policy network is:

$$L^P(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) \hat{A}_t, r_t^{\text{clip}} \hat{A}_t \right) \right] - \lambda H(\pi_\theta), \quad (13)$$

where $r_t^{\text{clip}} = \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)$, θ is the parameter of the policy model, $r_t(\theta)$ is the probability ratio between the old policy and the current policy, ϵ is set to 0.2 [16], $\hat{A}_t = R_t - Q_\phi(s_t, a_t)$, and $Q_\phi(s_t, a_t)$ is calculated by the Critic Network. And $H(\pi_\theta)$ is the entropy of the strategy.

E. Multi-Layer Perceptron-Based Critic Networks

We concatenate the output of the policy network and the embedding of the state encoder into a vector, then we apply MLP to map the vector to the Q-value which evaluates the quality of actions. More specifically,

$$Q_\phi(s_t, a_t) = \text{MLP}(\text{LN}(\text{concat}(\pi(a|s), \mathbf{h}_i))), \quad (14)$$

where $Q_\phi(s_t, a_t) \in \mathbb{R}^1$ is the estimated expectation of discounted cumulative reward drawn from the policy model based

on the current state. Then the loss function of the critic network is defined by the mean square error:

$$L^C(\phi) = \mathbb{E}_t \left[(Q_\phi(s_t) - R_t)^2 \right], \quad (15)$$

where ϕ is the parameter of the critic network. We update the parameters of the policy and critic networks by minimizing these two loss functions with the Adam optimizer [17].

V. EXPERIMENTAL RESULTS

A. Experimental Setup

The computational setup for our experiments includes an Intel Xeon CPU with 32 cores, complemented by eight NVIDIA A100 GPUs (each with 80 GB of memory) and 1024 GB of system RAM. The software environment is based on Python 3.10 (64-bit) running on a Linux operating system. For the training process, we utilized the following hyperparameters: a batch size of 1024, a learning rate of 0.0005 for both the policy and critic networks, a discount factor for rewards of 0.99, and a target network update coefficient of 0.01.

B. DataSet

- MAGICAL Dataset [11]: This dataset includes three operational transconductance amplifier (OTA) circuits and one comparator (COMP) circuit, all implemented using the TSMC40 technology node. Without access to the Process Design Kit (PDK), simulation is unfeasible, so the evaluation focuses on wire length and area. To assess the layout quality after routing, SAGERoute [18], a routing tool compatible with this dataset's process, is employed.
- SMIC 130nm and 180nm Dataset: This dataset includes OTA and Operational Amplifier (OpAmp) circuits. After the layouts are completed, Cadence Virtuoso's automatic routing function is utilized for post-processing. We then perform post-layout simulation with Cadence Spectre to evaluate the quality of placement solutions. Calibre PEX is used to extract parasitic resistance, parasitic capacitance, and coupling capacitance (R+C+CC).

C. Methods for Comparison

In this paper, we compare different SOTA DRL approaches for IC placement tasks. The Decision Transformer (DEC) [19] method uses transformers to predict actions, represented by Chipformer [20] for digital IC placement, which is our baseline for analog IC placement. The Multi-Agent Transformer (MAT) [21] improves multi-agent reinforcement learning with centralized, sequential policy search. MAPPO [22], an actor-critic on-policy algorithm, combines centralized training with decentralized execution. Its extension, RMAPPO [22], uses recurrent networks to capture state sequence correlations. HATRPO [23] adds hierarchical structures and temporal rewards for optimizing policies in complex environments with varied agents. We also analyze MAGICAL [11], an open-source placement tool for a custom TSMC 40nm simplified PDK. Finally we evaluate Virtuoso Automatic Placer, a commercial tool using SMIC 130nm and 180nm technology nodes.

D. Experimental Results and Analysis

1) Evaluation on SMIC Technology Node

We first verify our SACPlace in commercial datasets by simulating the performance of the analog IC layout. The comparison results of a 5-stage OpAmp under SMIC 130nm technology node are listed in Table I:

- SACPlace achieves the best Power Supply Rejection Ratio (PSRR), Common Mode Rejection Ratio (CMRR), and Bandwidth (BW) while handling the symmetry constraints. It reflects that our SACPlace facilitates the satisfaction of constraints to improve the performance of analog IC.
- SACPlace improves Unity Gain Frequency (UGF) by 5.3% at the expense of 0.8% degradation in Phase Margin (PM) compared to manual design. It shows that our SACPlace achieves placement results with performance metrics comparable to manual designs.

Similar levels of improvement are seen in other cases. Our method yields outcomes comparable to manual layout techniques in terms of performance metrics. Specifically, Table II summarizes the performance of various methods for OTA and OpAmp circuits using the SMIC 180nm technology node:

- SACPlace achieves the highest PSRR and CMRR in OpAmp. It indicates that the placement results of SACPlace have excellent noise rejection capabilities.
- SACPlace maintains competitive power consumption while achieving high gain and UGF in OTA. The BW and PM metrics also show significant improvements. It highlights the robustness and stability of our SACPlace results.
- The commercial placement tool, Virtuoso, struggles to adequately meet symmetry constraints, resulting in suboptimal performance across several crucial metrics.
- These SOTA DRL methods [19], [21]–[23] exhibit sub-optimal performance in placement tasks, attributed to constraints in their model scale and fitness.

2) Evaluation on TSMC 40nm Technology Node

Table III presents the experimental results on the MAGICAL dataset. The focus of this evaluation is on actual routing WL and area utilization, and the routing result is generated by [18]:

- SACPlace demonstrates superior performance in symmetry-aware placement compared to MAGICAL.
- Our method surpasses other SOTA DRL approaches by achieving the shortest WL and smallest area utilization across all tested circuits. It highlights that SACPlace is more

TABLE I: Experimental results of a 5-stage OpAmp under SMIC 130nm technology node

	5-stage OpAmp				
	PSRR (dB) \uparrow	CMRR (dB) \uparrow	PM (degree) \uparrow	UGF (MHz) \uparrow	BW (MHz) \uparrow
Schematic	42.731	105.891	89.995	5.152	1.636
Manual	40.598	98.035	76.412	4.772	1.562
MAT [21]	36.285	90.709	70.793	4.385	1.342
MAT-DEC [19]	35.803	88.453	69.046	4.259	1.303
MAPPO [22]	37.012	90.684	71.532	4.371	1.392
R-MAPPO [22]	37.126	91.245	72.318	4.592	1.421
HATRPO [23]	38.671	93.512	75.214	4.812	1.519
Virtuoso	39.047	95.476	76.169	4.683	1.539
SACPlace (ours)	40.692	98.317	75.803	5.022	1.594

TABLE II: Experimental results of OTA and OpAmp under SMIC 180nm technology node

	OTA						OpAmp		
	PSRR (dB) \uparrow	CMRR (dB) \uparrow	Power (uW) \downarrow	Gain (dB) \uparrow	UGF (MHz) \uparrow	BW (MHz) \uparrow	PSRR (dB) \uparrow	CMRR (dB) \uparrow	PM (degree) \uparrow
Schematic	41.765	62.357	69.748	36.394	136.353	9.390	84.821	83.392	70.507
Manual	39.991	44.391	71.551	35.945	123.501	8.535	83.855	83.037	67.407
MAT [21]	38.958	39.226	75.953	34.320	119.112	9.210	81.325	78.733	60.325
MAT-DEC [19]	37.803	42.427	78.582	34.653	119.538	9.130	81.956	79.201	62.941
MAPPO [22]	38.693	41.252	76.952	34.597	119.855	8.894	82.053	79.541	62.825
R-MAPPO [22]	38.921	42.975	78.753	35.391	119.539	9.010	81.592	78.904	63.578
HATRPO [23]	39.311	40.799	82.826	35.803	121.981	8.344	82.127	79.238	65.439
Virtuoso	39.949	49.995	81.231	35.855	118.732	8.599	83.931	83.035	67.352
SACPlace (ours)	39.975	50.329	72.953	35.967	122.503	9.232	84.385	83.173	67.531

TABLE III: Experimental results of MAGICAL dataset

	OTA1		OTA2		OTA3		COMP	
	WL (um) \downarrow	Area (um 2) \downarrow	WL (um) \downarrow	Area (um 2) \downarrow	WL (um) \downarrow	Area (um 2) \downarrow	WL (um) \downarrow	Area (um 2) \downarrow
MAT [21]	19869.42	1820.97	46617.37	4470.81	70894.21	9063.51	4741.82	161.82
MAT-DEC [19]	23518.27	1838.51	46847.46	4429.85	73225.79	9094.42	4486.93	162.86
MAPPO [22]	25106.91	1803.82	47365.56	4245.31	67544.49	8973.44	4831.82	162.94
R-MAPPO [22]	23743.49	1820.12	46827.53	4348.89	85830.28	9021.32	5012.82	163.57
HATRPO [23]	23628.25	1792.82	46887.02	4021.86	69717.32	9008.58	4621.93	160.89
MAGICAL [11]	19025.83	1820.45	46402.15	4470.26	64892.09	9063.50	4419.28	161.70
SACPlace (ours)	18829.34	1780.29	46012.75	3762.89	62874.53	9001.73	4206.13	158.65

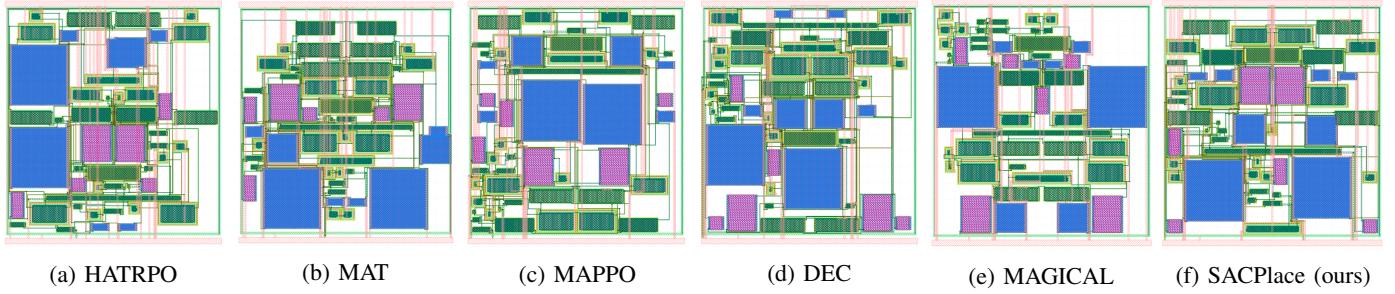


Fig. 4: Placement results of MAGICAL OTA2

efficient in achieving higher placement quality compared to SOTA DRL methods.

- Although MAGICAL achieves considerable outcomes in terms of symmetry, it only adheres to hard symmetry constraints during its legalization process, resulting in relatively suboptimal solutions.

E. Case Studies

Fig. 4 presents the visualization results of the open-source case from MAGICAL, which involves approximately 50 devices requiring symmetry and features a complex net relationship alongside other constraints. Our method effectively satisfies all complex analog IC constraints, achieving the shortest wirelength and optimal area utilization. In contrast, SOTA DRL methods struggle to deliver competitive results when confronted with such complex constraints, while MAGICAL leaves room for improvement in placement.

Overall, the experimental results demonstrate that SACPlace outperforms existing methods in analog IC placement, delivering significant improvements in key performance metrics while also minimizing wire length and area utilization. These outcomes

validate the effectiveness and efficiency of our proposed approach in addressing the challenges of analog IC placement design.

VI. CONCLUSION

In this paper, we present a symmetry-aware multi-agent reinforcement learning method for analog IC layout, named SACPlace, which uses multi-agent policy networks and multi-layer perceptron-based critic networks to handle complex constraints. This approach converts placement optimization into gradient optimization, assessing placement quality with symmetry efficiently. Experimental results show that our method outperforms manual layouts and other baselines on open source datasets and real-world analog IC designs. Future work will expand the SACPlace framework for larger-scale designs.

ACKNOWLEDGMENT

This work was supported by National Science and Technology Major Project (2021ZD0114600), National Natural Science Foundation of China (No. 62276260, 62076235, 62002106, 62441233).

REFERENCES

- [1] Y. Lin, Y. Li, D. Fang, M. Madhusudan, S. S. Sapatnekar, R. Harjani, and J. Hu, “Are analytical techniques worthwhile for analog ic placement?” in *2022 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2022, pp. 154–159.
- [2] B. Xu, S. Li, C. W. Pui, D. Liu, and D. Z. Pan, “Device layer-aware analytical placement for analog circuits,” in *the 2019 International Symposium*, 2019.
- [3] H. C. Ou, K. H. Tseng, J. Y. Liu, I. P. Wu, and Y. W. Chang, “Layout-dependent effects-aware analytical analog placement,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 8, pp. 1243–1254, 2016.
- [4] M. Liu, K. Zhu, X. Tang, B. Xu, and D. Z. Pan, “Closing the design loop: Bayesian optimization assisted hierarchical analog layout synthesis,” in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020.
- [5] J. Lu, L. Lei, J. Huang, F. Yang, L. Shang, and X. Zeng, “Automatic op-amp generation from specification to layout,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 12, pp. 4378–4390, 2023.
- [6] A. Gusmo, R. Póvoa, N. Horta, N. Loureno, and R. Martins, “Deepplacer: A custom integrated opamp placement tool using deep models,” *Appl. Soft Comput.*, vol. 115, p. 108188, 2021.
- [7] A. Gusmão, F. Passos, R. Póvoa, N. Horta, N. Lourenço, and R. Martins, “Semi-supervised artificial neural networks towards analog ic placement recommender,” in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.
- [8] Y. Li, Y. Lin, M. Madhusudan, A. K. Sharma, W. Xu, S. S. Sapatnekar, R. Harjani, and J. Hu, “A customized graph neural network model for guiding analog ic placement,” in *ICCAD ‘20: IEEE/ACM International Conference on Computer-Aided Design*, 2020.
- [9] D. Basso, L. Bortolussi, M. Videnovic-Misic, and H. Habal, “Fast ml-driven analog circuit layout using reinforcement learning and steiner trees,” *arXiv preprint arXiv:2405.16951*, 2024.
- [10] M. Liu, K. Zhu, X. Tang, B. Xu, W. Shi, N. Sun, and D. Z. Pan, “Closing the design loop: Bayesian optimization assisted hierarchical analog layout synthesis,” in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.
- [11] H. Chen, M. Liu, B. Xu, K. Zhu, X. Tang, S. Li, Y. Lin, N. Sun, and D. Z. Pan, “Magical: An open- source fully automated analog ic layout system from netlist to gdsii,” *IEEE Design Test*, vol. 38, no. 2, pp. 19–26, 2021.
- [12] J. Lu, H. Zhuang, P. Chen, H. Chang, C.-C. Chang, Y.-C. Wong, L. Sha, D. Huang, Y. Luo, C.-C. Teng, and C.-K. Cheng, “eplace-ms: Electrostatics-based placement for mixed-size circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 5, pp. 685–698, 2015.
- [13] D. Guerra, A. Canelas, R. Póvoa, N. Horta, N. Loureno, and R. Martins, “Artificial neural networks as an alternative for automatic analog ic placement,” *IEEE*, 2019.
- [14] M. Ahmadi and L. Zhang, “Analog layout placement for finfet technology using reinforcement learning,” in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.
- [15] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, “A survey of monte carlo tree search methods,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.
- [16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *ArXiv*, vol. abs/1707.06347, 2017.
- [17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.
- [18] H. Zhang, X. Gao, H. Luo, J. Song, J. Liu, X. Tang, Y. Lin, R. Wang, and R. Huang, “Sageroute: Synergistic analog routing considering geometric and electrical constraints with manual design compatibility,” in *IEEE/ACM Proceedings Design, Automation and Test in Europe (DATE)*, 2023.
- [19] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, “Decision transformer: Reinforcement learning via sequence modeling,” in *Neural Information Processing Systems*, 2021.
- [20] Y. Lai, J. Liu, Z. Tang, B. Wang, J. Hao, and P. Luo, “Chipformer: Transferable chip placement via offline decision transformer,” in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 2023, pp. 18346–18364.
- [21] M. Wen, J. Kuba, R. Lin, W. Zhang, Y. Wen, J. Wang, and Y. Yang, “Multi-agent reinforcement learning is a sequence modeling problem,” in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 16509–16521.
- [22] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, “The surprising effectiveness of PPO in cooperative multi-agent games,” in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.
- [23] J. G. Kuba, R. Chen, M. Wen, Y. Wen, F. Sun, J. Wang, and Y. Yang, “Trust region policy optimisation in multi-agent reinforcement learning,” in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.