ModuPlace: LLM-Assisted Modular PCB Placement via Preference-Optimized Constraint Graph Generation

Yaohui Han[1], Beichen Li[2], Mingyang Zhao[3], Rongliang Fu[1,*], Qunsong Ye[4], Tinghuan Chen[2,*], Bei Yu[1], Tsung-Yi Ho[1]
[1]CUHK, [2]CUHK-Shenzhen, [3]Central South University, [4]Index

Introduction

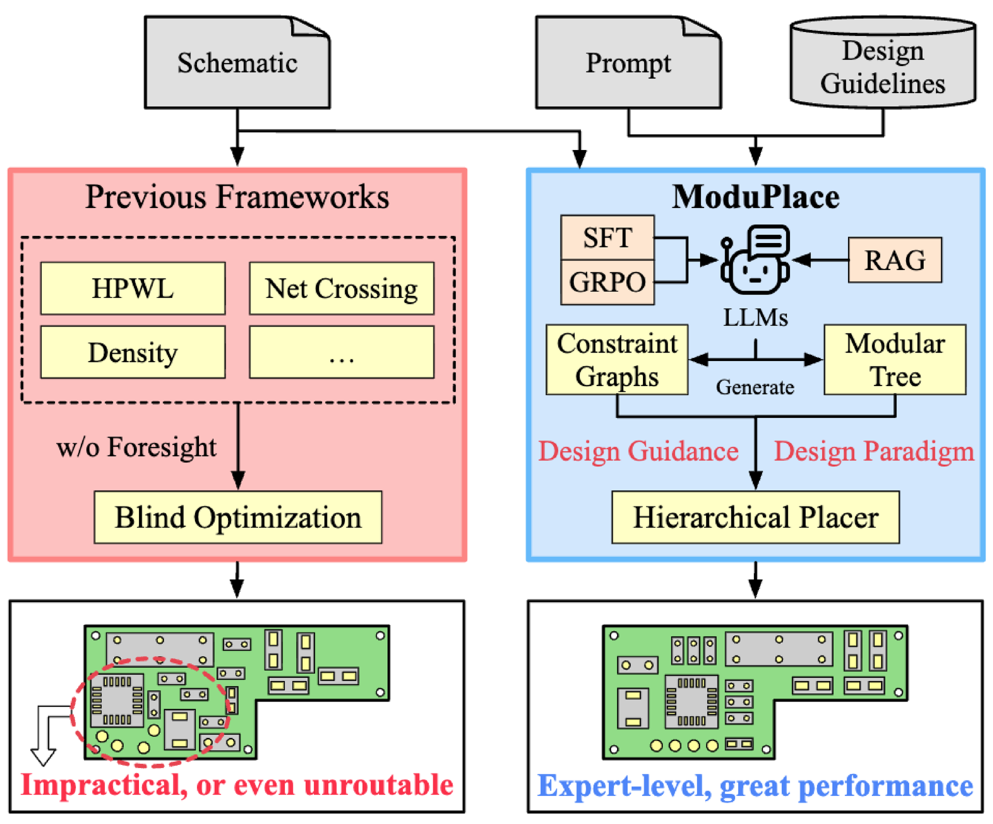
Placement is a crucial aspect of PCB design, determining the solution space of subsequent routing and thus playing a significant role in the overall design quality.

Previous Frameworks:

- **Heuristic**: SA-PCB [1], NS-Place [2].
- **AI-based**: DeepPCB [3], PCBAgent [4].

Existing Limitations:

- Lack of guidance from human engineers.
- Without considering modular information, the results are always unreasonable or even unusable [4, 5].
- Users cannot intervene the placement process through natural languages.

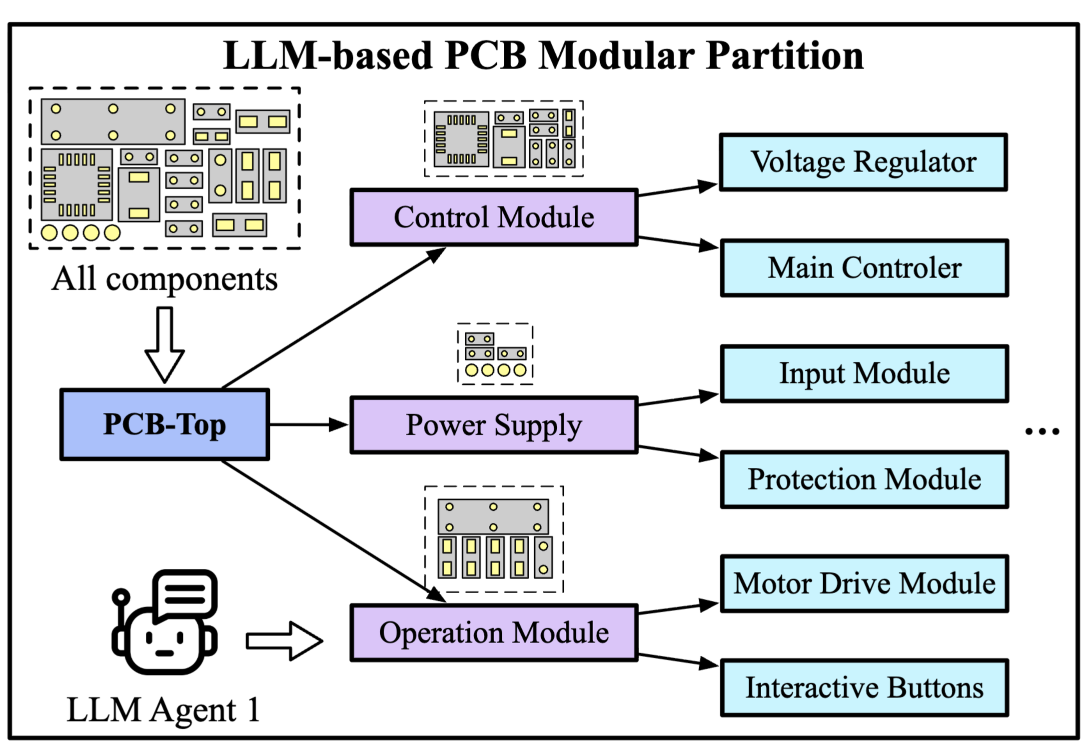


Comparison between existing frameworks and ModuPlace

Core Contributions:

- The first **modular-designed** PCB placement framework.
- **Comprehensive Constraint Graph** (CCG) is proposed to guide hierarchical placement.
- A **two-stage specialization** is utilized, including SFT and GRPO [6] processes.
- Experiments demonstrate that ModuPlace improves the overall performance compared with all baselines.

Methodology



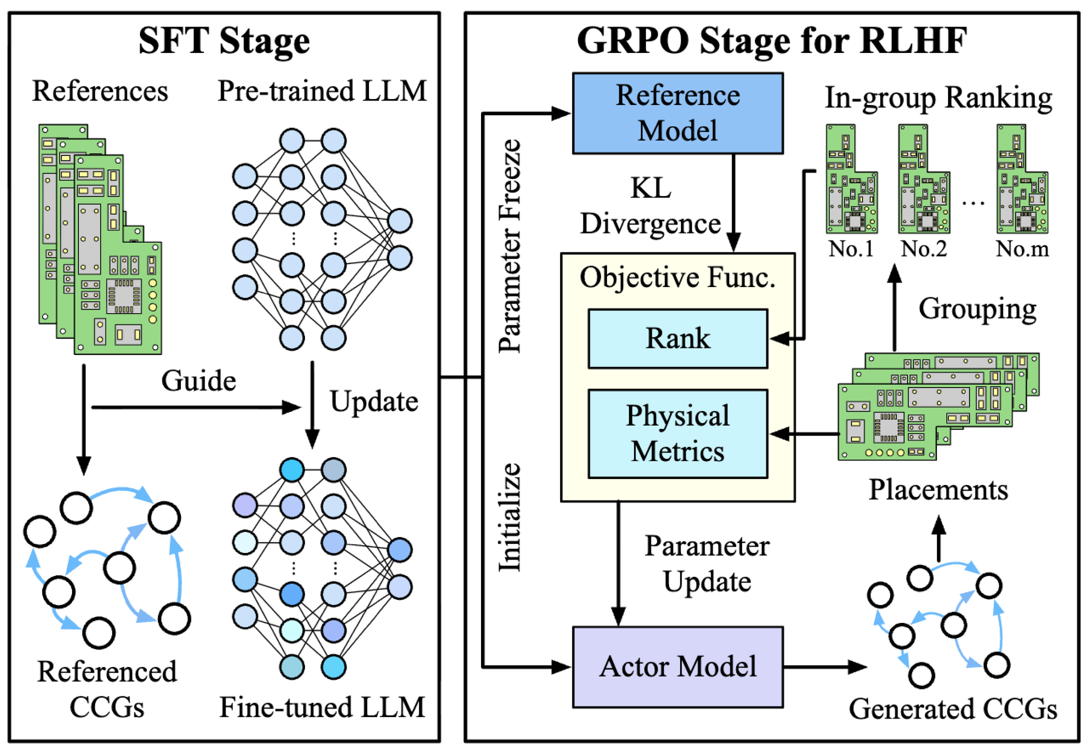
An example of the LLM-based PCB modular partition

1. LLM-based PCB Modular Partition:

- Hierarchical Modular Tree (HMT) is proposed to strengthen the effectiveness of modular partition through an LLM agent.
- We construct a netlist heterogenous graph [7] to generate a simplified format for the better understanding of LLM.
- The modular information in the HMT also enables subsequent hierarchical placement.

2. LLM-driven CCG Generation:

- Hierarchical Modular Tree (HMT) is proposed to strengthen the effectiveness of modular partition through an LLM agent.
- We construct a netlist heterogenous graph [7] to generate a simplified format for the better understanding of LLM.
- The modular information in the HMT also enables subsequent hierarchical placement.



The proposed two-stage specialization, including SFT and GRPO

2.1. SFT Stage

The loss function of SFT, where $\mathcal{N}$ is net crossing and $\mathrm{sim}(.,.)$ is the similarity between CCGs:

$$\mathcal{L}_j = \lambda_1 \cdot \mathrm{HPWL}_j + \lambda_2 \cdot \mathcal{N}_j - \lambda_3 \cdot \mathrm{sim}(g_j, g_{\mathrm{ref}}),$$

2.2. Customized GRPO Stage

The objective function:

$$\mathcal{J}(\theta) = \mathbb{E}[q \sim P(Q), \mathcal{G}_i^c \sim \pi_{\theta_{\mathrm{old}}}(O|q)]$$
$$\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}\left[\min\left(I_j\hat{A}_j, \mathrm{clip}(I_j, 1-\epsilon, 1+\epsilon)\hat{A}_j\right)\right] - \beta\mathbb{D}_{\mathrm{KL}_j},$$

The customized advantage function:

$$\hat{A}_j = \omega_1 \cdot \frac{\mu_{\mathrm{HPWL}} - \mathrm{HPWL}_j}{\sigma_{\mathrm{HPWL}}} + \omega_2 \cdot \frac{\mu_{\mathcal{N}} - \mathcal{N}_j}{\sigma_{\mathcal{N}}} + \omega_3 \cdot e^{-\hat{r_j}},$$

2.3. CCG Equivalence Determination

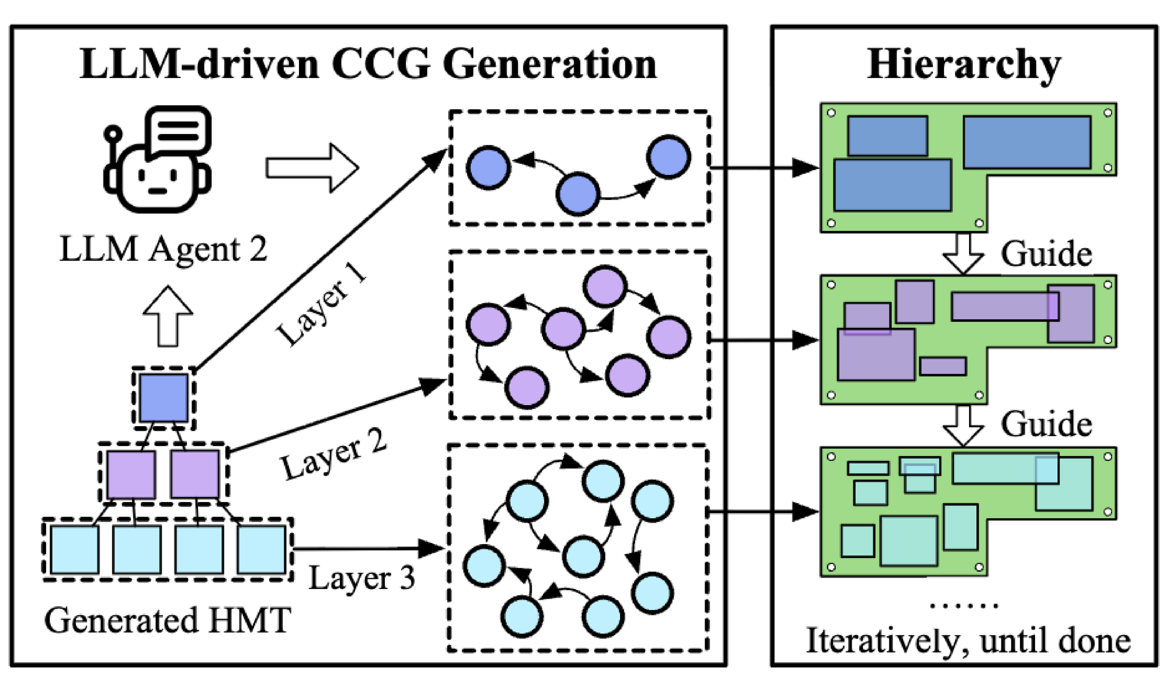
In actual situations, complete CCG consistency is extremely rare. Therefore, we proposed a CCG Equivalence Determination scheme to support the GRPO stage.

3. Hierarchical PCB Placement

Since the components have been organized into modules in the HMT, ModuPlace then performs hierarchical placement guided by the CCGs.

3.1. Place the modules hierarchically:

$$\min \sum_{i=1}^{|S_q|}\sum_{j=i}^{|S_q|} w_{ij}^c \cdot [\gamma_1 \cdot (|\hat{x_{ij}} - x_{ij}| + |\hat{y_{ij}} - y_{ij}|)$$
$$+ \gamma_2 \cdot |\hat{\theta_{ij}} - \theta_{ij}|] + \gamma_3 \cdot A_i^q + \gamma_4 \cdot \sum_{k=1}^{|net|} \mathrm{HPWL}(x, y, \theta),$$
$$\text{s.t. } \theta_i \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}, \ \forall i \in S_q,$$



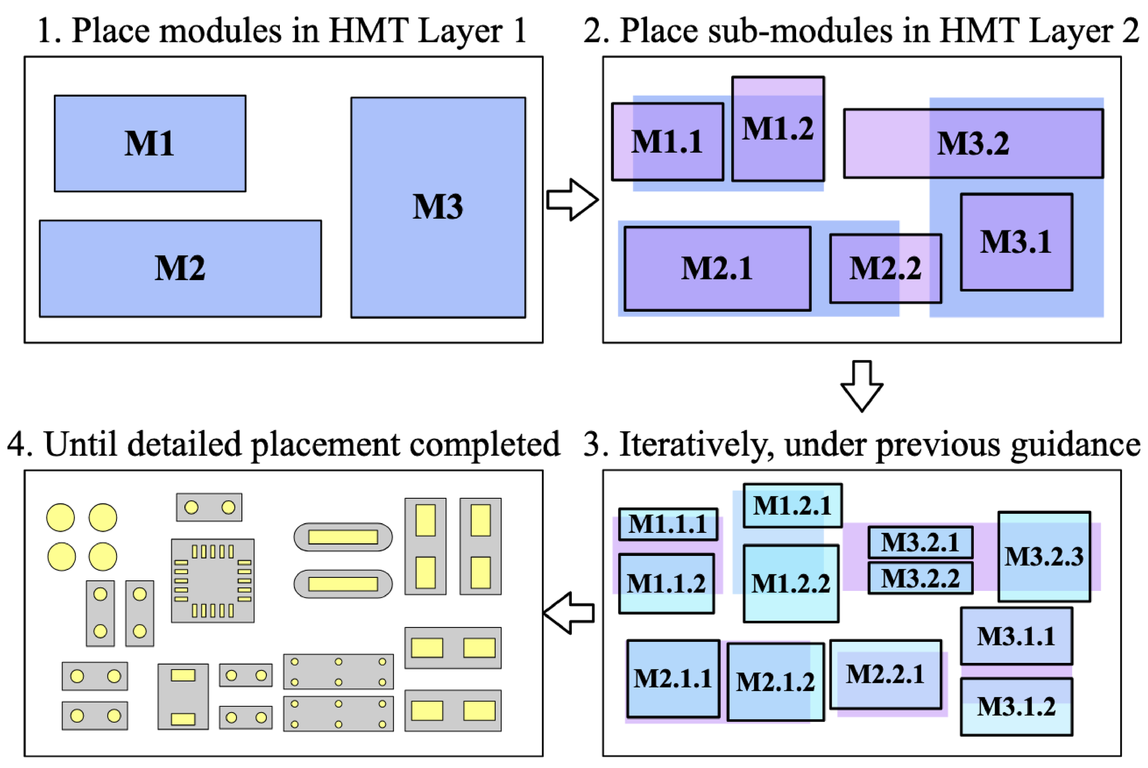
LLM agent takes the HMT as input, and generates CCGs layer by layer to support the subsequent hierarchical placement.

All the modules act as **guidance** for the placement of sub-modules in the next layer of HMT.

3.2. Detailed Placement

$$\min \ (t_1 \cdot \mathrm{BoundingBox}(C_q) + t_2 \cdot \sum_{k=1}^{|net|} \mathrm{HPWL}(x, y, \theta)),$$
$$\text{s.t. } \theta_i \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}, \ \forall i \in C_q,$$

With the guidance from the last layer of HMT, the detailed placement is performed to finish placement.



An example of hierarchical PCB Placement

ModuPlace learned from expert design experience and preferences, and completed the placement in a modular method.

Experimental Results

- Compared with the classical baseline, SA-PCB [1], ModuPlace shows 17.14% wirelength and 40.51% via count reduction.
- Even compared with the SOTA baseline, DeepPCB [3], ModuPlace still shows significant lower wirelength and via count.
- Compared to pre-trained LLMs, ModuPlace demonstrates better performance on Qwen-14B, illustrating the effectiveness of the framework.
- ModuPlace is **flexible** to adapt to users' **varying requirements** in different applications.

Conclusion

- **ModuPlace**: **the first** modular PCB placement approach, integrating expert experience via LLM.
- CCG is proposed to provide a standard format to guide hierarchical placement, alleviate hallucinations, and improve overall performance.
- ModuPlace utilizes a two-stage specialization method, with the implementation of SFT and GRPO for CCG generation enhancement.
- Experimental results demonstrate that ModuPlace can significantly improve the overall performance compared with all baselines.

References

[1] SA-PCB. [Online]. Available: https://github.com/The-OpenROAD-Project/SA-PCB.
[2] Cheng, Chung-Kuan, Chia-Tung Ho, and Chester Holtz. "Net separation-oriented printed circuit board placement via margin maximization." 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2022.
[3] DeepPCB. [Online]. Available: https://app.deeppcb.ai/.
[4] Chen, Lin, et al. "PCBAgent: An agent-based framework for high-density printed circuit board placement." Proceedings of the 30th Asia and South Pacific Design Automation Conference. 2025.
[5] Li, Shanyi, et al. "HiePlace: Efficient Hierarchical PCB Placement." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2025).
[6] Shao, Zhihong, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).