ChatPattern: Layout Pattern Customization via Natural Language

Zixiao Wang[1*], Yunheng Shen[2*], Xufeng Yao[1], Wenqian Zhao[1], Yang Bai[1], Farzan Farnia[1], Bei Yu[1]

[1]The Chinese University of Hong Kong, [2]Tsinghua University

The Scope of ChatPattern

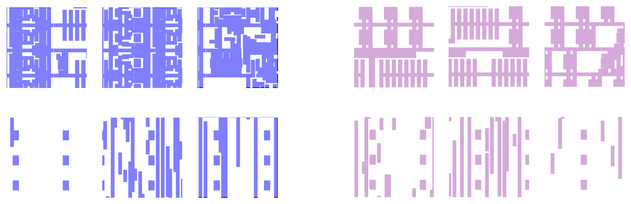
ChatPattern is an AI agent that offers a conversational interface, enabling users to use natural language to guide the creation of pattern libraries that meet their specific layout generation needs.

Highlights

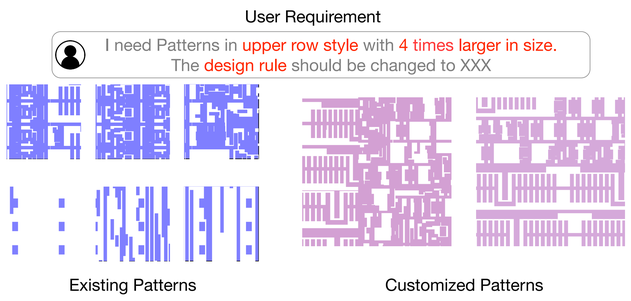
The contributions of this work are fourfold:

- Introduction of ChatPattern, the first LLM-powered tool for creating layout patterns.
- Integration of an expert LLM agent that builds pattern libraries from natural language inputs and uses tools automatically.
- Creation of a versatile model that surpasses current methods in generating patterns based on conditions, modifying layouts, and extending patterns of any size.
- Expansion of the layout pattern generation field, encouraging researchers to tackle more realistic and challenging tasks like generating layouts of any size.

From Generation to Customization



Pattern Generation VLSI layout patterns provide critical resources in various designs for manufacturability research. Pattern Generation task aims to mimic the distribution of existing patterns.



Pattern Customization. The requirements on layout pattern distributions can vary in real cases. Pattern Customization task aims to generate patterns to meet specialized requirements.

Overview of ChatPattern

ChatPattern seamlessly integrates a front-end powered by a Large Language Model with a back-end that employs a conditional discrete-diffusion model for layout pattern generation.

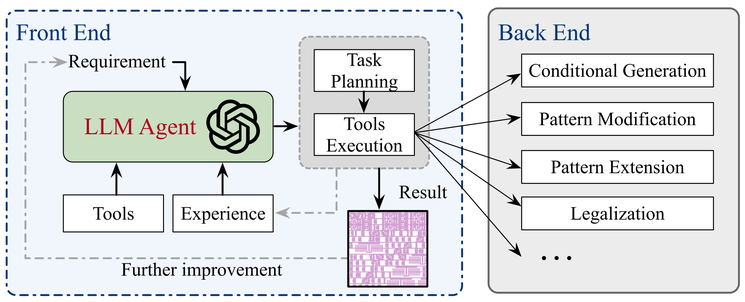


Figure 1. An illustration of **ChatPattern**.

Front End: Expert LLM Agent

This front-end LLM agent communicates with clients via natural language communication, adeptly understanding user requirements, and orchestrating scripts to efficiently generate a pattern library. The duties of LLM agent include:

- Requirement Auto-Formatting
- Task Planning and Execution
- Tool Function Learning and Application
- Learning from Documents and Experience

One key idea is that the LLM agent does not directly access generated patterns, which is outside the scope of pre-training. Instead, the LLM agent generates patterns via tools and gets feedback from evaluation metrics and the running log.

Back End: Flexible Generative Model

The back-end pattern generative model, providing API functions for LLM agent, is specifically designed for tasks involving free-size pattern generation. The provided functions include:

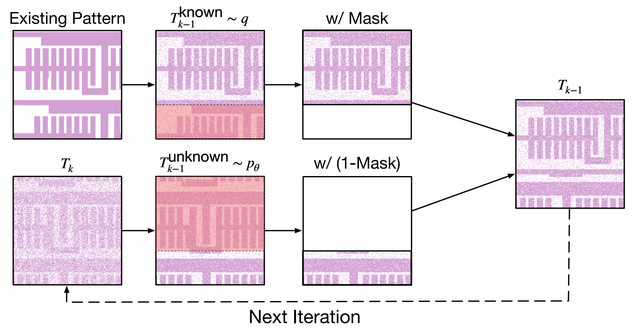
Property-Conditional Topology Generation. The condition design in pattern generation should consider the design rules, materials, and manufacturing process. In our conditional discrete diffusion model, a topology matrix with condition $\mathbf{c}$ can be generated by a K-step reverse process from the randomly-sampled noise $\mathbf{T}_K$,

$$p_\theta(\mathbf{T}_0|\mathbf{T}_K,\mathbf{c}) = p_\theta(\mathbf{T}_0|\mathbf{T}_1,\mathbf{c})\prod_{k=2}^{K} p_\theta(\mathbf{T}_{k-1}|\mathbf{T}_k,\mathbf{c}). \quad (1)$$

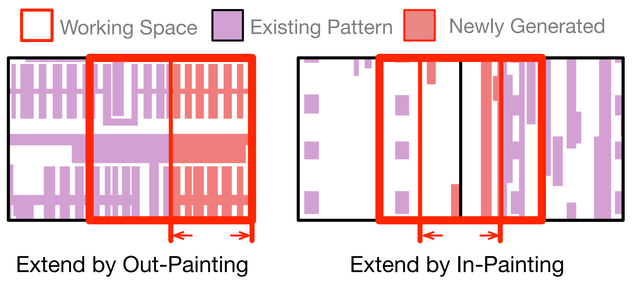
Pattern Modification. Given an existing pattern topology matrix $\mathbf{T}_0^{\text{known}}$, making modifications to any desired region on it can be useful when dealing with failed topology.

$$\begin{aligned}\mathbf{T}_{k-1}^{\text{known}} &\sim \mathbf{q}\left(\mathbf{T}_{k-1}|\mathbf{T}_0^{\text{known}}\right),\\ \mathbf{T}_{k-1}^{\text{unknown}} &\sim \mathbf{p}_\theta\left(\mathbf{T}_{k-1}|\mathbf{T}_k,\mathbf{c}\right),\\ \mathbf{T}_{k-1} &= \mathbf{M}\odot\mathbf{T}_{k-1}^{\text{known}} + (1-\mathbf{M})\odot\mathbf{T}_{k-1}^{\text{unknown}},\end{aligned} \quad (2)$$

where $\mathbf{T}_0^{\text{known}}$ shares the design rules with patterns in condition $\mathbf{c}$ and $\mathbf{M}$ denotes the mask.



Pattern Extension. Extending a given pattern to a larger one is a practical function since the model output usually takes a fixed size while the required patterns can vary among a large range.



Legalization. We utilize the non-linear legalization function proposed in DiffPattern[1] to legalize the generated patterns.

Numerical Results

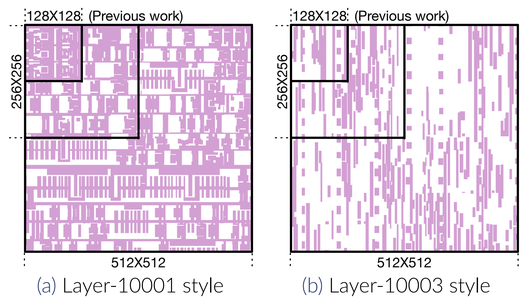
Table 1. Comparison on fixed-size and free-size pattern generation task. '/' refers to not applicable.

Task	Set/Method	Training Set*	Size	Layer-10001 Legality (↑)	Layer-10001 Diversity (↑)	Layer-10003 Legality (↑)	Layer-10003 Diversity (↑)	Total† Legality (↑)	Total† Diversity (↑)
Fixed-size	Real Patterns	/	128^2	/	10.731	/	8.769	/	10.625
	CAE+LegalGAN [3]	Layer-10001		3.74%	5.814	/	/	/	/
	VCAE+LegalGAN [3]	Layer-10001		84.51%	9.867	/	/	/	/
	LayouTransformer [2]	Layer-10001		89.73%	10.527	/	/	/	/
	DiffPattern [1]	Layer-10001/10003		**99.97%**	10.711	99.98%	8.578	**99.98%**	10.633
	ChatPattern	Layer-10001/10003		**99.97%**	**10.796**	**99.99%**	**8.625**	**99.98%**	**10.650**
Free-size	Real Patterns	/	256^2	/	12.702	/	10.696	/	12.695
	[1] w/ Concatenation	Layer-10001/10003		57.78%	10.719	93.69%	10.511	75.74%	11.706
	ChatPattern	Layer-10001/10003		**87.36%**	**11.154**	**99.78%**	**10.556**	**93.57%**	**11.830**
	Real Patterns	/	512^2	/	13.435	/	12.139	/	13.787
	[1] w/ Concatenation	Layer-10001/10003		0.29%	5.714	40.83%	11.555	20.56%	11.359
	ChatPattern	Layer-10001/10003		**36.42%**	**10.401**	**98.86%**	**11.620**	**67.64%**	**12.133**
	Real Patterns	/	1024^2	/	13.573	/	12.644	/	14.109
	[1] w/ Concatenation	Layer-10001/10003		0.00%	0.000	0.64%	6.926	0.32%	6.926
	ChatPattern	Layer-10001/10003		**1.19%**	**6.438**	**94.96%**	**11.981**	**47.80%**	**11.992**

* All training datasets are the 128×128 version.
† We collected generated samples from both Layer-10001/10003 and evaluated them together.

Observation of TABLE 1. While all methods are trained on small-scale pattern datasets 128^2, the legality of patterns generated by **ChatPattern** can be **100**× higher than baseline methods when the size reaches 512^2 or larger.

Pattern Extension. We illustrate some instances of Pattern Extension in Fig. 2.



(a) Layer-10001 style (b) Layer-10003 style

Figure 2. 512×512 topology matrix generated by **ChatPattern**.

Requirement Auto-formatting. An example of the requirement list is following.

```
# Requirement - subtask 1

## Basic Part: Topology Size: [200, 200], Physical Size: [1500, 1500] nm, Style: Layer-10001, Count: 50000,

## Advanced Part: Extension Method: Out (Default: Out), Drop Allowed: True (Default: True), Time Limitation: None (Default: None).
```

Conclusion

We introduced **ChatPattern**, a novel framework for pattern generation utilizing a LLM. **ChatPattern** provides a user-friendly interface that accepts natural language inputs to tailor the pattern library to specific needs.

References

[1] Zixiao Wang, Yunheng Shen, Wenqian Zhao, Yang Bai, Guojin Chen, Farzan Farnia, and Bei Yu. Diffpattern: Layout pattern generation via discrete diffusion. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2023.

[2] Liangjian Wen, Yi Zhu, Lei Ye, Guojin Chen, Bei Yu, Jianzhuang Liu, and Chunjing Xu. Layoutransformer: Generating layout patterns with transformer via sequential pattern modeling. In *ICCAD*, 2022.

[3] Xiaopeng Zhang, James Shiely, and Evangeline FY Young. Layout pattern generation and legalization with generative learning models. In *ICCAD*, pages 1–9, 2020.