ChatPattern: Layout Pattern Customization via Natural Language
Zixiao Wang\textsuperscript{1}, Yunheng Shen\textsuperscript{2}, Xufeng Yao\textsuperscript{1}, Wenyuan Zhao\textsuperscript{3}, Yang Bai\textsuperscript{1}, Farzan Farnia\textsuperscript{4}, Bei Yu\textsuperscript{5}
\textsuperscript{1}The Chinese University of Hong Kong, \textsuperscript{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 libraries from natural language inputs and uses tools automatically.
- 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.

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 $e$ can be generated by a $K$-step reverse process from the randomly-sampled noise $T_0$:

$$p_k(T_0|T_{K}, e) = p_k(T_0|T_{K}, e) \prod_{k=1}^{K} p_k(T_{k-1}|T_k, e).$$ (1)

Pattern Modification. Given an existing pattern topology matrix $T_0$ known, making modifications to any desired region on it can be useful when dealing with failed topology.

$$T_0 = M \odot T_0^{\text{known}} + (1 - M) \odot T_0^{\text{unknown}}, \quad T_0^{\text{known}}$$

where $T_0^{\text{known}}$ shares the design rules with patterns in condition $e$ and $M$ 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.

Legislation. We utilize the non-linear legalization function proposed in DiffPattern\textsuperscript{1} to legalize the generated patterns.

Table 1. Comparison on fixed-size and free-size pattern generation task. $\uparrow$ refers to not applicable.

Observation of TABLE 1. While all methods are trained on small-scale pattern datasets\textsuperscript{18}, the legibility 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.

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

- Requirement - subtask 1
- # Basic Part: Topology Size: [200, 200], Physical Size: [1500, 1500] mm, Style: Layer-10001, Count: 50000.

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

Figure 1. An illustration of ChatPattern.