ComfyUI-Copilot: An Intelligent Assistant for Automated Workflow Development

Zhenran Xu[1,2]*, Xue Yang[1]†, Yiyu Wang[1], Qingli Hu[1], Zijiao Wu[1],
Longyue Wang[1], Weihua Luo[1], Kaifu Zhang[1], Baotian Hu[2]†, Min Zhang[2]

[1]Alibaba International Digital Commerce   [2]Harbin Institute of Technology (Shenzhen)

{xuzhenran.xzr,shali.yx,menshuan.wyy,qingli.hql,zijiao.wzj}@alibaba-inc.com
{wanglongyue.wly,weihua.luowh,kaifu.zkf}@alibaba-inc.com
{hubaotian,zhangmin2021}@hit.edu.cn

⊙ https://github.com/AIDC-AI/ComfyUI-Copilot

Abstract

We introduce **ComfyUI-Copilot**, a large language model-powered plugin designed to enhance the usability and efficiency of ComfyUI, an open-source platform for AI-driven art creation. Despite its flexibility and user-friendly interface, ComfyUI can present challenges to newcomers, including limited documentation, model misconfigurations, and the complexity of workflow design. ComfyUI-Copilot addresses these challenges by offering intelligent node and model recommendations, along with automated one-click workflow construction. At its core, the system employs a hierarchical multi-agent framework comprising a central assistant agent for task delegation and specialized worker agents for different usages, supported by our curated ComfyUI knowledge bases to streamline debugging and deployment. We validate the effectiveness of ComfyUI-Copilot through both offline quantitative evaluations and online user feedback, showing that it accurately recommends nodes and accelerates workflow development. Additionally, use cases illustrate that ComfyUI-Copilot lowers entry barriers for beginners and enhances workflow efficiency for experienced users. The ComfyUI-Copilot installation package and a demo video are available at https://github.com/AIDC-AI/ComfyUI-Copilot.

1 Introduction

Recent advancements in large language models (LLMs) and image generation methods have democratized AI-generated content (AIGC) production, with open-source frameworks like ComfyUI (comfyanonymous, 2023) emerging as pivotal tools for low-code AI workflow development. Serving over 4 million active users and backed by a vibrant community contributing 12K+ compo-

nents (e.g., SDXL (Podell et al., 2023), ControlNet (Zhang et al., 2023)), ComfyUI enables flexible workflow orchestration via drag-and-drop components for multimodal tasks such as text-to-image generation, face swapping, and video editing.

Despite its convenience, newcomers may face several potential barriers when starting with ComfyUI. These challenges include the installation of dependent nodes and models, scattered documentation across forums and GitHub issues. Even experienced users require substantial expertise to debug and construct a well-designed workflow (Gal et al., 2024). Recent research on automatic workflow construction has limitations, such as instability (i.e., generating unprocessable workflows) and a narrow focus primarily on text-to-image generation tasks (Xue et al., 2024; Sobania et al., 2024). Addressing these challenges and facilitating the onboarding process for ComfyUI is therefore crucial.

To this end, we introduce **ComfyUI-Copilot**, an LLM-empowered multi-agent framework designed to assist users in navigating ComfyUI. It provides the following key features: **(1) Automatic workflow generation:** Our copilot can identify user intent, retrieve or synthesize an appropriate workflow, and then integrate it into the ComfyUI canvas. An example of its functionality is shown in Figure 1. **(2) Node and model recommendation:** Our copilot can suggest suitable nodes based on user instructions, recommend relevant checkpoints and LoRA models. **(3) ComfyUI-related question answering:** Our copilot provides detailed tutorials on selected nodes and models, including usage guidelines, installation steps, and parameter explanations. It can also offer multiple feasible downstream subgraphs for selected nodes. Additionally, we introduce new features aimed at enhancing workflow debugging and optimization, including prompt writing and parameter search.

The framework of ComfyUI-Copilot is centered around an LLM-based assistant agent, which co-

---

(a) ComfyUI Workflow Generation　　　　　　　　(b) One-click Deployment

Figure 1: **An example of automated workflow generation in ComfyUI-Copilot**: the copilot suggests multiple workflows based on the user instruction and loads the selected one into the canvas with a single click.

ordinates with various specialized worker agents and knowledge bases (KBs). Depending on the query, the assistant agent may address user queries directly or delegate tasks to appropriate worker agents. We have developed three primary worker agents focused on workflow generation, node and model recommendation. To support these agents, we have constructed extensive KBs covering 7K nodes, 62K models, and 9K workflows. These KBs are enhanced through automated documentation generation by leveraging LLM's code comprehension capabilities, and are continuously expanded and updated daily. Unlike prior work (Gal et al., 2024; Sobania et al., 2024) which only targets text-to-image generation, the resources in our KBs extend to conditional multimodal generation tasks, ensuring that our system accommodates both diverse tasks and cutting-edge modules with accuracy.

Experiments show that ComfyUI-Copilot provides accurate assistance in node recommendation and workflow construction based on user instructions. The high recall rates for workflows and nodes (both exceeding 88.5%) validate the practical efficacy in automated workflow development and accurate node recommendation. Since its release on GitHub, online user feedback reflects a moderately high acceptance rate of 65.4% for recommended nodes and a notably high acceptance rate of 85.9% for proposed workflows. Use cases further highlight the system's capability to reduce entry barriers for beginners and enhance workflow efficiency for experienced ComfyUI users with multilingual support.

To the best of our knowledge, ComfyUI-Copilot is the first open-source project to develop a ComfyUI plugin for automating workflow creation and providing instant suggestions. As of the camera-ready date (May 29, 2025), it has already attracted a rapidly growing user base, accumulating over 1.6K GitHub stars and processing more than 85K queries from 19K users across 22 countries. In future work, we plan to incorporate feedback from the active open-source community and continuously update features to better address user needs.

2 Related Work

AI-generated content (AIGC) based on ComfyUI. Diffusion models have gained wide attention in AI research for image synthesis (Ho et al., 2020; Dhariwal and Nichol, 2021). As the field of text-to-image generation progresses, new tasks and models (Kumari et al., 2023; Ruiz et al., 2023; Li et al., 2023; Zhang et al., 2023) have been proposed to introduce controllable conditions in image generation. Therefore, researchers and practitioners are transitioning from simple text-to-image workflows to more sophisticated ones, where the open-source ComfyUI (comfyanonymous, 2023) offers great convenience. In ComfyUI, users can easily construct workflows by connecting a series of blocks, each representing specific models or parameter choices. Each ComfyUI workflow can be exported to a JSON file which outlines both the graph nodes and their connectivity.

Instead of relying on an end-to-end diffusion model for image generation, advanced workflows combine a variety of components to enhance image quality (Guo et al., 2024; Ye et al., 2023). These
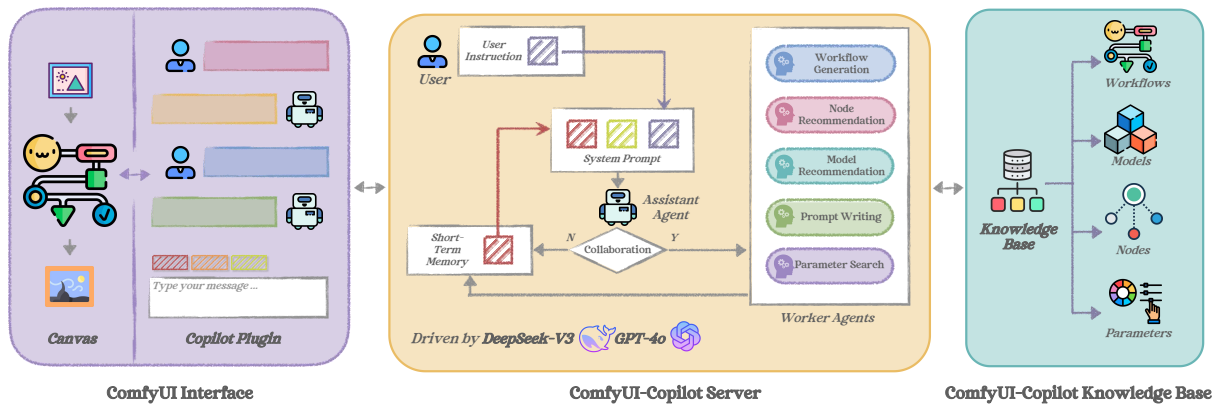
Figure 2: **Overview of the ComfyUI-Copilot framework**: The central LLM-based assistant agent can either respond directly to user instructions based on the conversation history (i.e., short-term memory), or collaborate with specialized worker agents. These agents are supported by our curated ComfyUI knowledge bases.

components may include fine-tuned versions of generative models, large language models (LLMs) for refining input prompts, LoRAs trained to introduce specific artistic styles, improved latent decoders for finer details, super-resolution blocks, and more (Hu et al., 2021; Mañas et al., 2024; Berrada et al., 2025; Ning et al., 2021). Importantly, effective workflows are prompt-dependent, with the selection of models and nodes often based on the user intent and the desired image content (Gal et al., 2024). Therefore, creating a well-designed workflow and selecting appropriate nodes and models require significant expertise, where our ComfyUI-Copilot comes into help.

LLM-based agents. Recent advancements in LLMs have demonstrated great improvements in reasoning abilities and adaptability to new content and tasks (Chen et al., 2024b; Zeng et al., 2024; Wang et al., 2025). Based on these emergent capabilities (Wei et al., 2022), various studies have utilized LLMs for agentic task completion using external tools, including hallucination detection (Cheng et al., 2024b), visual question answering (Cheng et al., 2024a; Yin et al., 2024), and web navigation (Agashe et al., 2025; Yang et al., 2025; Li et al., 2025). In addition to tools, LLM-based agents are often equipped with components such as memory mechanisms (Wang et al., 2024; Xu et al., 2025), retrieval modules (Asai et al., 2024; Kim et al., 2024) and reasoning strategies like self-reflection (Shinn et al., 2023; Xu et al., 2023), aimed at enhancing their overall performance.

Our work proposes a multi-agent framework for the automated development and deployment of ComfyUI workflows. In this framework, the LLM acts as the central planner, autonomously select-

ing suitable worker agents to address diverse user queries. Although recent research has shown increasing interest in workflow generation (Gal et al., 2024; Xue et al., 2024; Sobania et al., 2024), existing methods often face challenges such as instability, leading to unparseable output workflows, or are limited to text-to-image tasks. We broaden the scope to include various conditional image and video generation tasks, and address user queries with a high acceptance rate.

3 ComfyUI-Copilot

In this section, we provide a detailed description of ComfyUI-Copilot. As illustrated in Figure 2, the system utilizes a hierarchical multi-agent framework that includes a central assistant agent for task delegation and specialized worker agents for different usages. We first introduce our curated ComfyUI knowledge bases (Sec. 3.1), and the details of the multi-agent framework (Sec. 3.2). Following this, we present the interactive chat interface and provide usage examples in Section 3.3.

3.1 Knowledge Bases

We have constructed three KBs about nodes, models and workflows. The data is sourced from popular platforms for sharing generative resources, ComfyUI-related GitHub repositories, and the ComfyUI website, with NSFW content filtered out.

For nodes lacking structured documentation, we automatically generate detailed documentation by analyzing their GitHub repositories. As shown in Figure 3, the process begins by setting up a sandbox environment to run ComfyUI, cloning the GitHub repositories, and installing the necessary dependencies. After successfully importing the nodes
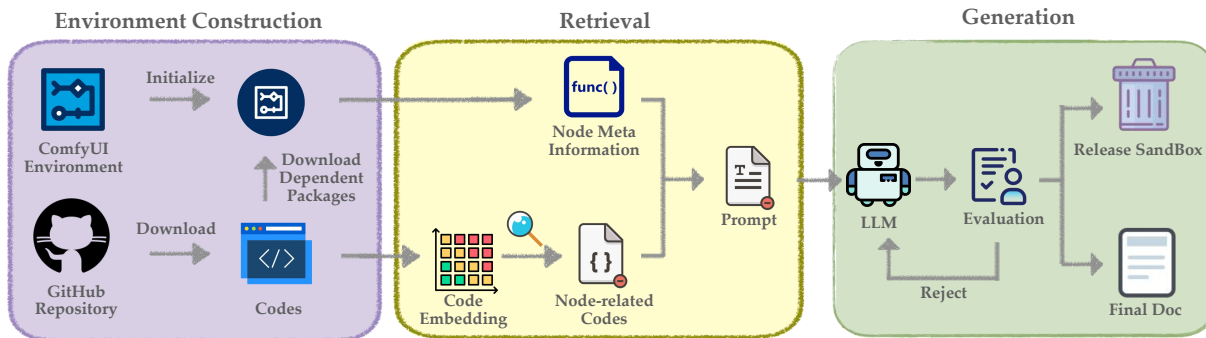
Figure 3: **The process of automatic node documentation generation.** Starting from GitHub repositories, the process involves constructing an executable ComfyUI environment, followed by code chunking and retrieval, and concludes with generating the final documentation.

within ComfyUI, we extract metadata, including the node class type, input and output parameters. The GitHub code is then segmented into chunks, which are embedded using the BGE-M3 embedding (Chen et al., 2024a), followed by retrieval to locate relevant code for each node. By combining the metadata with the code, we use an LLM to generate documentation on node usage and parameter meanings. The generated documentation undergoes quality reviews before finalization, with an example provided in Appendix A.

In addition, since community-sourced content tends to focus more on installation instructions, there is often a lack of detailed explanations of workflow and model functionalities. To address this, we leverage the multimodal understanding capabilities of GPT-4o, by prompting it with community-sourced texts, accompanying images that typically demonstrate the effects of the workflows or models, and any available workflow JSON files. This approach helps fill in the gaps in usage descriptions, which is essential for further developing effective recommendation worker agents.

In total, we have constructed extensive KBs covering 7K nodes, 62K models, and 9K workflows. These KBs are continuously expanded weekly, covering a wide range of conditional image and video generation tasks. This ensures that ComfyUI-Copilot remains adaptable to both widely used and cutting-edge modules.

3.2 Agents

The core of ComfyUI-Copilot is a well-instructed LLM-based assistant agent, serving as a planner. Depending on the user instruction, the assistant either responds to queries using the constructed KBs or delegates tasks to appropriate worker agents. We have created three worker agents for workflow,

nodes and models, which we collectively refer to as "modules" in this section. The recommendation process for each module follows a three-stage pipeline, progressing from coarse to fine granularity.

In the first stage, we employ an LLM or a large multimodal model (LMM), such as DeepSeek-V3 or GPT-4o, to expand vague user instructions into detailed task descriptions and noteworthy considerations. For example, when performing style transfer, if the LMM identifies the original image as a human portrait, the expanded user intent will highlight the importance of maintaining subject consistency. In the second stage, we represent the user intent as an embedding and calculate its cosine distance with modules in the KB, obtaining a semantic score sim_S based on OpenAI's `text-embedding-3-small`. Additionally, we compute a lexical similarity score sim_L based on the proportion of overlapping words. The overall retrieval score sim_O is calculated as:

$$\text{sim}_O = 0.7 \times \text{sim}_S + 0.3 \times \text{sim}_L \qquad (1)$$

The top 30 modules with the highest sim_O scores are then selected for further re-ranking. In the third stage, we use the GTE-Rerank model[1] to determine the top 3 modules from the above candidates. The re-ranking score is obtained by providing the re-ranker with the user intent and the description of each candidate module. These top 3 modules are further ranked by considering popularity factors such as upvotes, downloads, and star statistics.

For the workflow generation agent, in addition to the module recalling pipeline, we explore the possibility of generating workflows from scratch based on code LLMs. As illustrated in Figure 4,

---

[1] https://huggingface.co/Alibaba-NLP/gte-multilingual-reranker-base
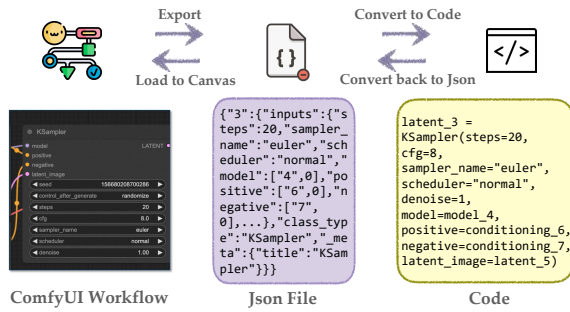
Figure 4: Different representations of ComfyUI workflows and their flexible conversions.

workflows can be represented in three common formats: ComfyUI flow graphs, JSON, and code. Following Xue et al. (2024), we enable mutual conversion between the JSON and code formats based on graph topology using Python-like syntax. We adopt code as the primary workflow representation due to its rich logical and semantic information, as well as its natural compatibility with LLMs' code generation capabilities. Given a user instruction, we prompt top-tier closed-source LLMs with retrieved nodes and code exemplars to generate workflows from scratch. Additionally, to investigate whether task-specific open-source models can replace closed-source LLMs, we fine-tune open-sourced Qwen2.5-Coder-7B (Hui et al., 2024) on workflows collected in our KB. Experimental results in Table 2 show that the fine-tuned model achieves performance comparable to Claude-3.7-Sonnet in terms of pass rate and node selection in generated workflows. More evaluation details are in Appendix B. However, due to the inherent complexity of workflow generation (Gal et al., 2024), there remains significant room for improvement in pass rates.

Implemented with LangChain[2], our framework (Figure 2) equips the assistant agent to autonomously select appropriate worker agents based on user instructions and short-term memory (i.e., message history). The assistant then synthesizes responses by integrating outputs from these worker agents, enabling automated ComfyUI-related question answering, workflow generation, and module recommendation. For prompt writing and parameter search functionality, we provide illustrative examples in Section 3.3.

3.3 Interface

As shown in Figure 1, ComfyUI-Copilot is seamlessly integrated into the ComfyUI interface. Users can launch our service with a single click on the ComfyUI-Copilot icon in the left sidebar. Once activated, the chat box displays user inputs and our copilot's responses. Users can engage in multiple rounds of conversation and switch between underlying LLMs such as DeepSeek-V3 and GPT-4o.

Automatic Workflow Generation. As illustrated in Figure 1, ComfyUI-Copilot responds to user instructions by presenting the top three recalled workflows. By clicking "Accept", the selected workflow can be loaded onto the canvas. If ComfyUI-Copilot detects that any required nodes are missing, it provides an installation guide and directs the user to the official GitHub repositories for easy setup (See Figure 5 (d)).

ComfyUI-related Question Answering. Users can click on any node to ask shortcut questions about its usage, parameters, and recommended downstream nodes. Figure 5 (a) and (c) illustrate this feature: a user inquires about the input and output parameters of the "KSampler" node, and ComfyUI-Copilot not only explains them but also suggests relevant downstream nodes, such as subgraphs for face swapping and image upscaling, to streamline workflow construction. Additionally, as shown in Figure 5 (b), ComfyUI-Copilot supports multilingual queries and responses (e.g., Polish in the example), enhancing accessibility for users worldwide.

Node and Model Recommendation. Module recommendations in ComfyUI-Copilot are context-aware, taking into account dependencies between components in the workflow. For instance, certain LoRA models perform optimally with specific diffusion models. As shown in Figure 6 (a), when a user requests a LoRA model for text-to-image generation, ComfyUI-Copilot prompts the user to specify the diffusion model being used before suggesting compatible LoRA models. Figure 6 (b) demonstrates an example of node recommendation. The interface displays detailed descriptions and GitHub star counts for each recommended node, allowing users to add their preferred choice to the canvas with a single click.

In addition to the core features that lower entry barrier for beginners, we also provide new features to enhance productivity for experienced ComfyUI users. The prompt-writing functionality in Fig-
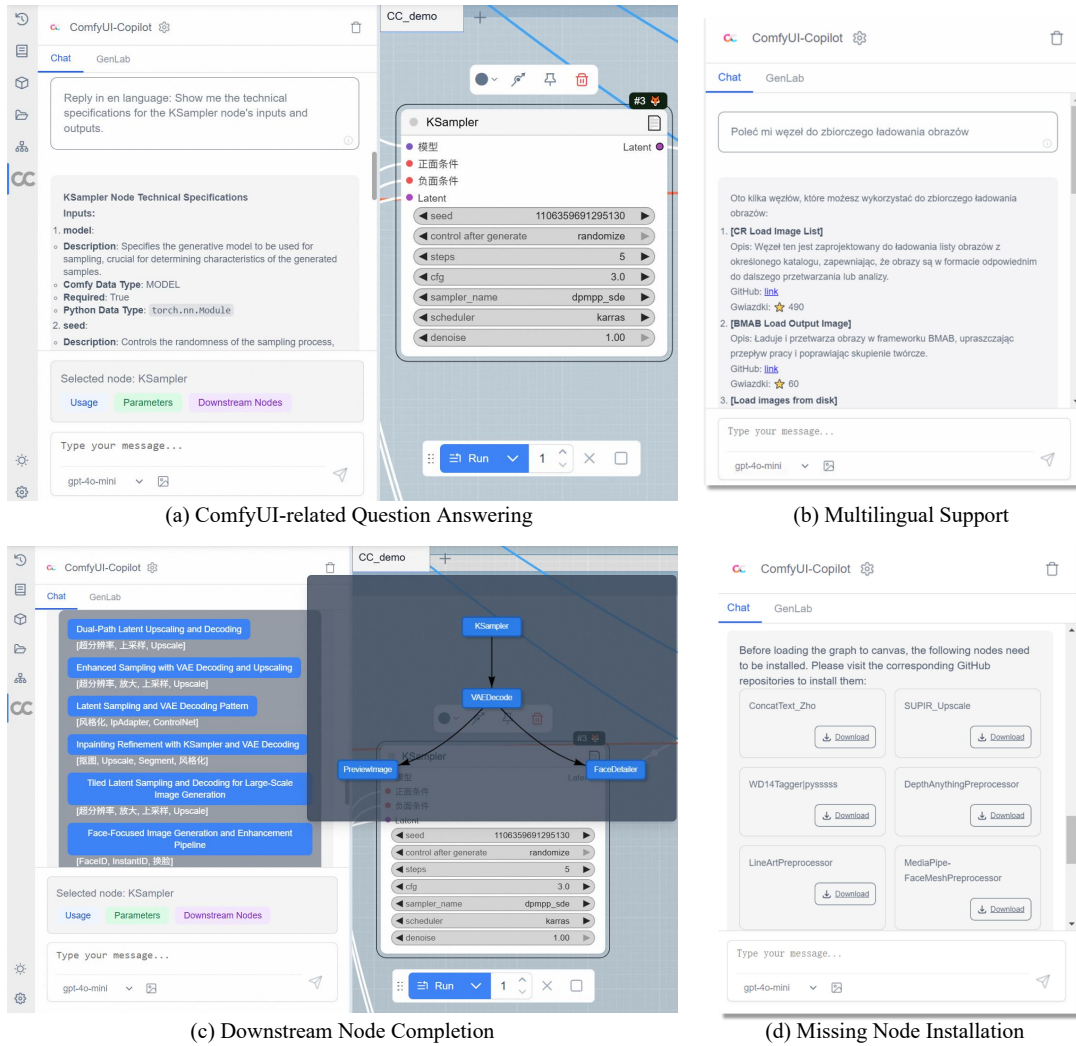
(a) ComfyUI-related Question Answering

(b) Multilingual Support

(c) Downstream Node Completion

(d) Missing Node Installation

Figure 5: Examples of ComfyUI-Copilot's different usages.

	DeepSeek-V3	GPT-4o
Node	0.885	0.894
Workflow	0.900	0.892

Table 1: Recall rates of nodes and workflows in ComfyUI-Copilot based on our constructed test set.

ure 7 helps users refine prompts for text-to-image generation, resulting in more vivid images. For example, given a simple instruction like "a cat", several detailed prompts are proposed, each leading to high-quality outputs. Figure 8 illustrates the parameter search functionality, which enables users to run parallel experiments by varying key parameters and batch-processing images for efficient comparison. In the given example, the image generated using the original workflow does not resemble the source sofa image. By experimenting with different combinations of parameters (specifi-

cally "cfg" and "denoise" in the KSampler node), the resulting images can be compared side by side, allowing users to easily identify the optimal parameters that best preserve the desired attributes.

4 Usage and Evaluation

To evaluate the performance of ComfyUI-Copilot, we designed 130 user instructions for workflow recall based on our workflow KB. These instructions are created by rewriting the usage descriptions of specific workflows, using the target workflow as the correct answer, Examples include "I need a workflow that is suitable for fast upscaling and image quality restoration". Similarly, We create 104 node recommendation instructions based on our node KB, such as "I want to enhance image aesthetics and resolution in AI art applications, recommend a suitable node".

As shown in Table 1, when recalling the top three

637

workflows and nodes, our framework achieves high recall rates (over 88.5%) with both GPT-4o and DeepSeek-V3. This demonstrates the robustness and effectiveness of our multi-agent framework. Error analysis on the unsuccessful workflow cases indicates that, even when the exact target workflow is not recalled, the suggested workflows often still fulfill the user's intended functions.

Since releasing ComfyUI-Copilot on GitHub on February 23, 2025, online user feedback has shown a moderately high acceptance rate of 65.4% for recommended nodes and a notably high acceptance rate of 85.9% for proposed workflows. As the first open-source project for a ComfyUI assistant plugin, ComfyUI-Copilot has quickly attracted a growing user base with active engagement, received over 1.6K Github stars, with 85K queries from 19K users across 22 countries. Thanks to the open-source community, we have gathered valuable feedback from GitHub issues and are actively updating features to better address user needs.

5 Conclusion

In this paper, we present ComfyUI-Copilot, an LLM-powered multi-agent framework designed to address ComfyUI-related queries and enable one-click workflow creation, thereby lowering the barriers of ComfyUI development. By leveraging an LLM as a core assistant agent and integrating specialized worker agents and extensive knowledge bases, ComfyUI-Copilot not only enhances the workflow generation process with a high recall rate, but also ensures that it stays current with the latest modules in multimodal generation. As the first project to explore a ComfyUI assistant plugin for providing instant suggestions, ComfyUI-Copilot has rapidly gathered over 1.6K stars, attracted 19K users across 22 countries and processed more than 85K queries. In future work, we plan to incorporate feedback from GitHub issues and actively update features to address user pain points, such as automatic workflow and parameter optimization.

Acknowledgments

References

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Eric Wang. 2025. Agent s: An open agentic framework that uses computers like a human. In *The Thirteenth International Conference on Learning Representations*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Tariq Berrada, Pietro Astolfi, Melissa Hall, Marton Havasi, Yohann Benchetrit, Adriana Romero-Soriano, Karteek Alahari, Michal Drozdzal, and Jakob Verbeek. 2025. Boosting latent diffusion with perceptual objectives. In *The Thirteenth International Conference on Learning Representations*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. 2024b. Agent-FLAN: Designing data and methods of effective agent tuning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9354–9366, Bangkok, Thailand. Association for Computational Linguistics.

Chuanqi Cheng, Jian Guan, Wei Wu, and Rui Yan. 2024a. From the least to the most: Building a plug-and-play visual reasoner via data synthesis. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4941–4957, Miami, Florida, USA. Association for Computational Linguistics.

Xiaoxue Cheng, Junyi Li, Xin Zhao, Hongzhi Zhang, Fuzheng Zhang, Di Zhang, Kun Gai, and Ji-Rong Wen. 2024b. Small agent can also rock! empowering small language models as hallucination detector. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14600–14615, Miami, Florida, USA. Association for Computational Linguistics.

comfyanonymous. 2023. Comfyui. https://github.com/comfyanonymous/ComfyUI.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc.

Rinon Gal, Adi Haviv, Yuval Alaluf, Amit H. Bermano, Daniel Cohen-Or, and Gal Chechik. 2024. Comfygen: Prompt-adaptive workflows for text-to-image generation. *Preprint*, arXiv:2410.01731.

Zinan Guo, Yanze Wu, Zhuowei Chen, Lang chen, Peng Zhang, and Qian HE. 2024. PuLID: Pure and lightning ID customization via contrastive alignment. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. 2024. Qwen2.5-coder technical report. *Preprint*, arXiv:2409.12186.

Minsoo Kim, Victor Bursztyn, Eunyee Koh, Shunan Guo, and Seung-won Hwang. 2024. RaDA: Retrieval-augmented web agent planning with LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13511–13525, Bangkok, Thailand. Association for Computational Linguistics.

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. 2023. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941.

Tianle Li, Max Ku, Cong Wei, and Wenhu Chen. 2023. Dreamedit: Subject-driven image editing. *arXiv preprint arXiv:2306.12624*.

Yunxin Li, Zhenyu Liu, Zitao Li, Xuanyu Zhang, Zhenran Xu, Xinyu Chen, Haoyuan Shi, Shenyuan Jiang, Xintong Wang, Jifang Wang, Shouzheng Huang, Xinping Zhao, Borui Jiang, Lanqing Hong, Longyue Wang, Zhuotao Tian, Baoxing Huai, Wenhan Luo, Weihua Luo, Zheng Zhang, Baotian Hu, and Min Zhang. 2025. Perception, reason, think, and plan: A survey on large multimodal reasoning models. *Preprint*, arXiv:2505.04921.

Oscar Mañas, Pietro Astolfi, Melissa Hall, Candace Ross, Jack Urbanek, Adina Williams, Aishwarya Agrawal, Adriana Romero-Soriano, and Michal Drozdzal. 2024. Improving text-to-image consistency via automatic prompt optimization. *Preprint*, arXiv:2403.17804.

Qian Ning, Weisheng Dong, Guangming Shi, Leida Li, and Xin Li. 2021. Accurate and lightweight image super-resolution with model-guided deep unfolding network. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):240–252.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *Preprint*, arXiv:2307.01952.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Dominik Sobania, Martin Briesch, and Franz Rothlauf. 2024. Comfygi: Automatic improvement of image generation workflows. *Preprint*, arXiv:2411.14193.

Jifang Wang, Xue Yang, Longyue Wang, Zhenran Xu, Yiyu Wang, Yaowei Wang, Weihua Luo, Kaifu Zhang, Baotian Hu, and Min Zhang. 2025. A unified agentic framework for evaluating conditional image generation. *Preprint*, arXiv:2504.07046.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2024. Agent workflow memory. *Preprint*, arXiv:2409.07429.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Zhenran Xu, Senbao Shi, Baotian Hu, Jindi Yu, Dongfang Li, Min Zhang, and Yuxiang Wu. 2023. Towards reasoning in large language models via multi-agent peer review collaboration. *Preprint*, arXiv:2311.08152.

Zhenran Xu, Jifang Wang, Baotian Hu, Longyue Wang, and Min Zhang. 2025. MeKB-sim: Personal knowledge base-powered multi-agent simulation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pages 393–403, Albuquerque, New Mexico. Association for Computational Linguistics.

Xiangyuan Xue, Zeyu Lu, Di Huang, Zidong Wang, Wanli Ouyang, and Lei Bai. 2024. Comfybench:

Benchmarking llm-based agents in comfyui for autonomously designing collaborative ai systems. *Preprint*, arXiv:2409.01392.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. 2025. Agentoccam: A simple yet strong baseline for LLM-based web agents. In *The Thirteenth International Conference on Learning Representations*.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. 2023. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*.

Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. 2024. Agent lumos: Unified and modular training for open-source language agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12380–12403, Bangkok, Thailand. Association for Computational Linguistics.

Fanghua Yu, Jinjin Gu, Zheyuan Li, Jinfan Hu, Xiangtao Kong, Xintao Wang, Jingwen He, Yu Qiao, and Chao Dong. 2024. Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild. *Preprint*, arXiv:2401.13627.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2024. AgentTuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3053–3077, Bangkok, Thailand. Association for Computational Linguistics.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *CVPR*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A Example of Node Documentation

Here we present an example of automatic node documentation generation using GPT-4o. The input GitHub repository is ComfyUI-SUPIR (Yu et al., 2024)[3]. The resulting documentation is as follows.

> ### SUPIR Upscale Documentation
>
> The SUPIR_Upscale node is designed to enhance image resolution using advanced upscaling techniques, leveraging the SUPIR and SDXL models for high-quality output. It allows for various configurations, including different upscaling methods and model parameters, to optimize the image enhancement process.
> ## Input types
>
> - **supir_model**
>   - Specifies the path to the SUPIR model, which is essential for the upscaling process, ensuring that the node can utilize the trained model for image enhancement.
>   - Type: `COMBO[STRING]`
>
> - **sdxl_model**
>   - Indicates the path to the SDXL model, which works in conjunction with the SUPIR model to improve the quality of the upscaled images.
>   - Type: `COMBO[STRING]`
>
> - **(More inputs omitted)**
>
> ## Output types
>
> - **upscaled_image**
>   - The resulting image after the upscaling process, enhanced in resolution and quality based on the input parameters.
>   - Type: `IMAGE`

---

[3] https://github.com/kijai/ComfyUI-SUPIR

B Automatic Workflow Generation Experiment

In this experiment, we randomly select 2K high-quality workflows from the KB for training and 100 for evaluation. The training data's input includes workflow usage, retrieved nodes, and code examples. The code representation of the target workflow is the desired output. We fine-tune Qwen2.5-Coder-7B (Hui et al., 2024) with LLaMA-Factory (Zheng et al., 2024). The fine-tuning process employs a learning rate of 1e-5 and a batch size of 16, with a sequence length of 16K.

We compare the fine-tuned Qwen2.5-Coder-7B with the retrieval-augmented method based on closed-source models such as GPT-4o and Claude-3.7-Sonnet. Evaluation metrics include the pass rate (i.e., whether the generated workflow can be executed within the ComfyUI canvas), the average number of nodes, and the precision, recall, and F1 score for node selection. Results in Table 2 show that our fine-tuned model performs comparably to Claude-3.7-Sonnet, achieving the highest F1 score for node selection (0.95). Although GPT-4o achieves the highest pass rate, a closer examination reveals that it tends to generate overly simplistic workflows (an average of 8 nodes). 83% of the workflows produced by GPT-4o contain fewer nodes than the target workflows, leading to low node recall rates. Despite the promising performance of our fine-tuned model and Claude-3.7-Sonnet, there remains significant room for further improvements in workflow generation.

Model	Pass	#Nodes	Node		
			P	R	F1
GPT-4o	**0.92**	8	0.91	0.65	0.75
Claude 3.7	0.73	13	0.90	0.88	0.88
Ours	0.74	**14**	**0.96**	**0.94**	**0.95**

Table 2: Performance comparison of LLMs for workflow generation across evaluation metrics. #Nodes means the number of nodes. Precision (P), recall (R), and F1-score at node level are reported.
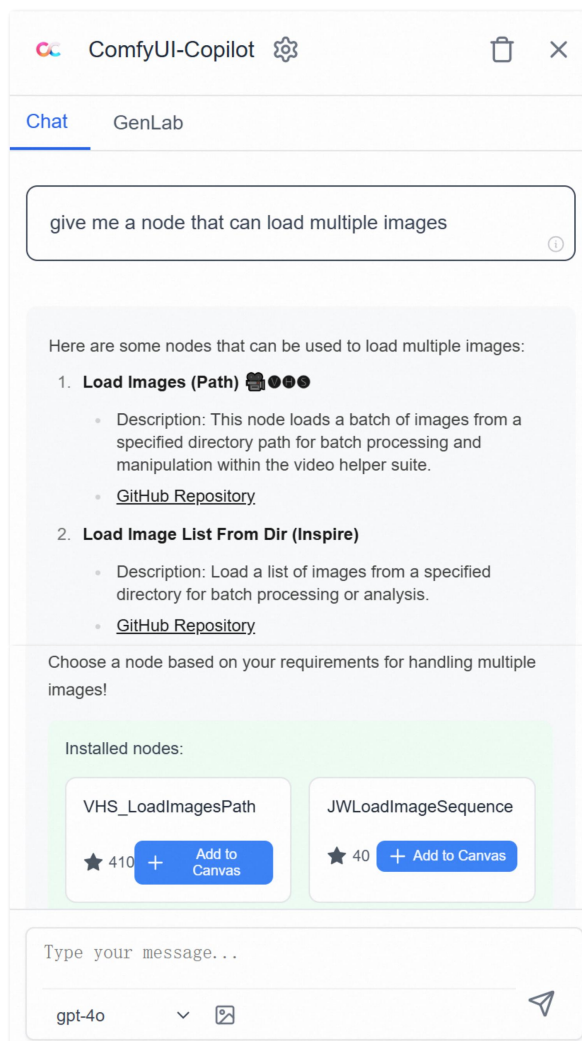
C More Examples of ComfyUI-Copilot

Due to page limit, we demonstrate the remaining functionalities in Figure 6, 7 and 8, including node and model recommendation, prompt writing assistance, and parameter search.
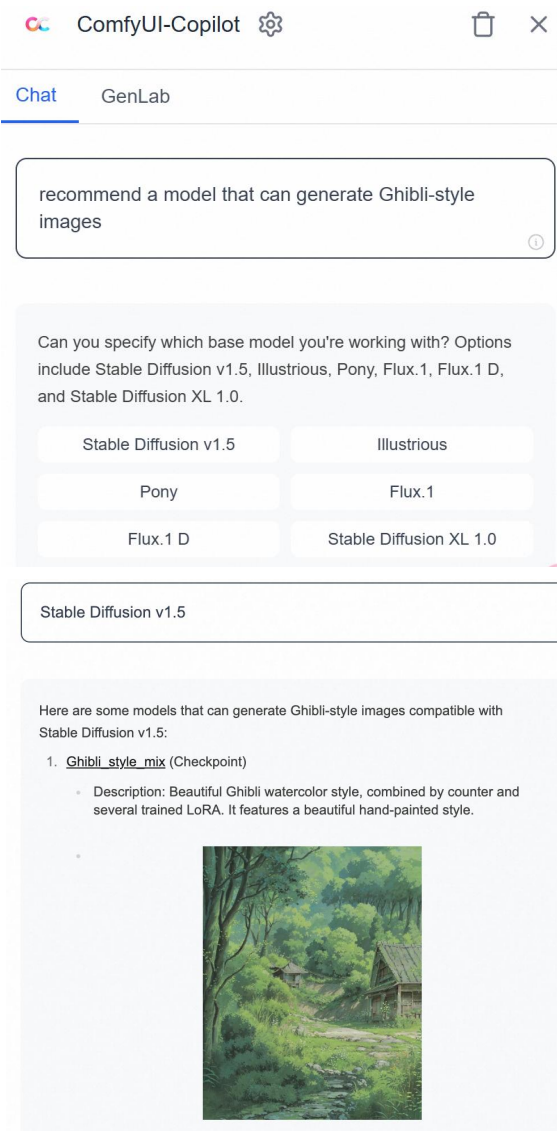
(a) Model Recommendation   (b) Node Recommendation

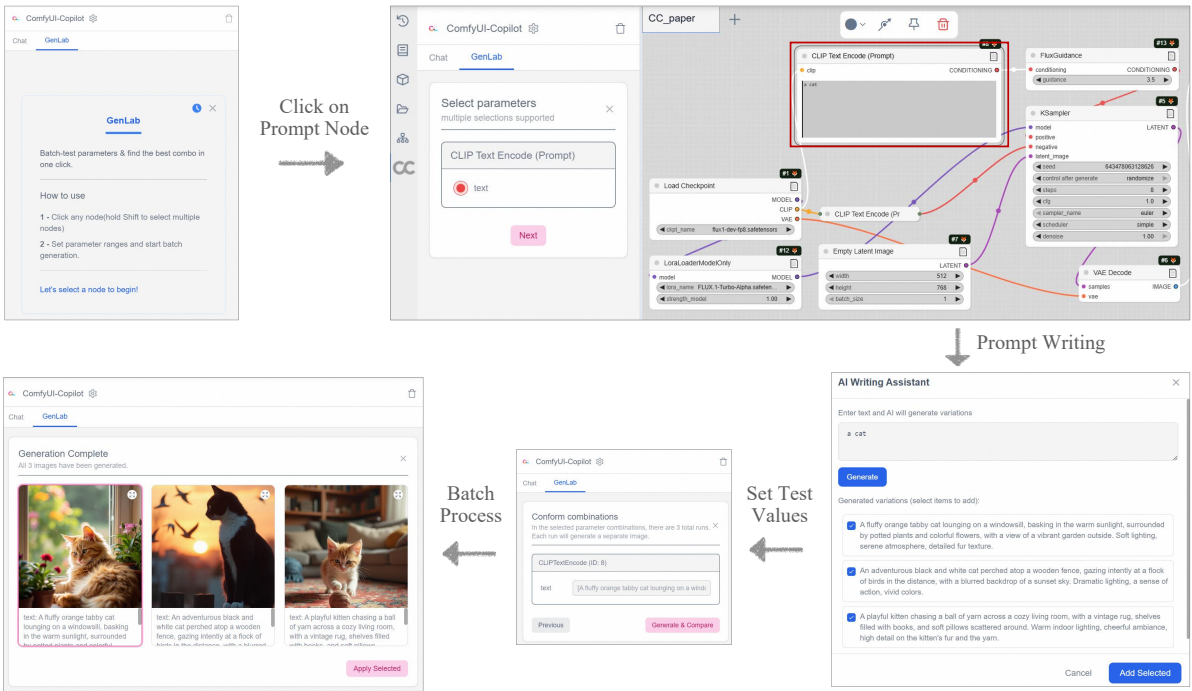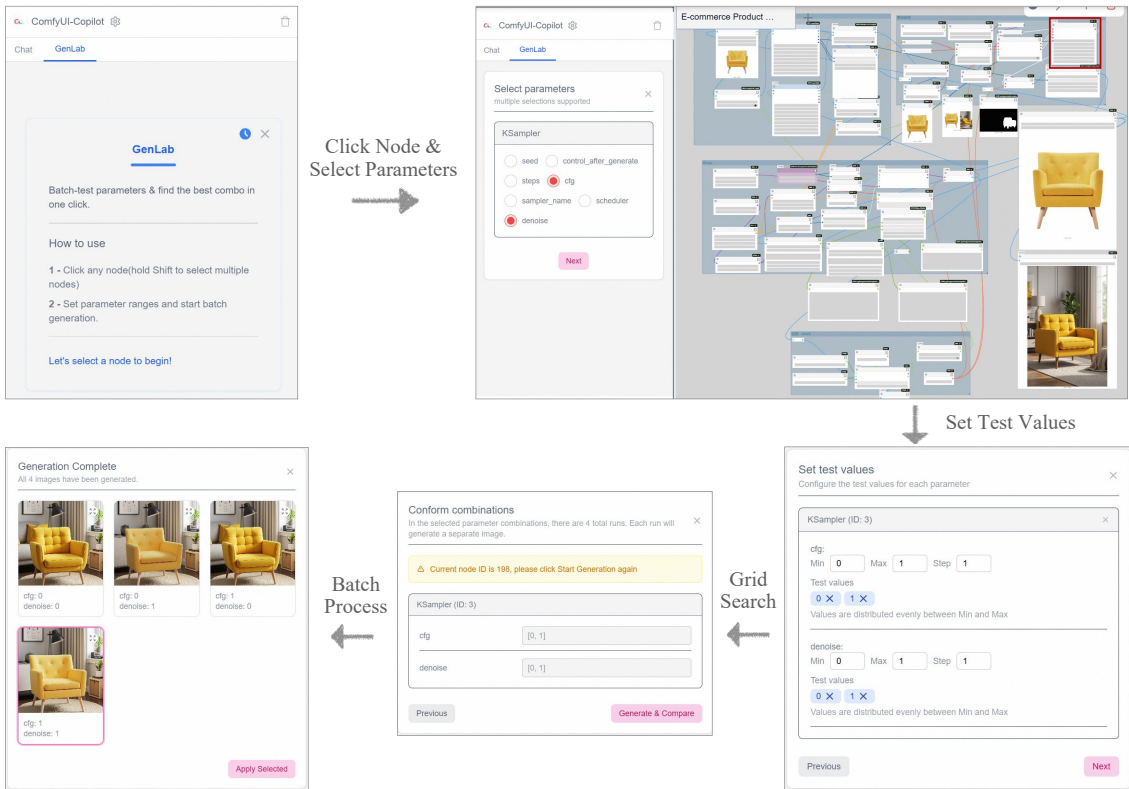Figure 6: Model and node recommendation in ComfyUI-Copilot.

Figure 7: Prompt writing in ComfyUI-Copilot.

Figure 8: Parameter search in ComfyUI-Copilot.