

ComfyUI-R1: Exploring Reasoning Models for Workflow Generation

Zhenran Xu^{*1}, Yiyu Wang^{*2}, Xue Yang², Longyue Wang², Weihua Luo², Kaifu Zhang²,
Baotian Hu^{†1}, Min Zhang¹

¹Harbin Institute of Technology (Shenzhen) ²Alibaba Group
xuzhenran@stu.hit.edu.cn, wangyiyu18@mails.ucas.ac.cn
{shali.yx, wanglongyue.wly, weihua.luowh}@alibaba-inc.com
{hubaotian, zhangmin2021}@hit.edu.cn

Abstract

AI-generated content has evolved from monolithic models to modular workflows, particularly on platforms like ComfyUI, enabling customization in creative pipelines. However, crafting effective workflows requires great expertise to orchestrate numerous specialized components, presenting a steep learning curve for users. To address this challenge, we introduce **ComfyUI-R1**, the first large reasoning model for automated workflow generation. Starting with our curated knowledge bases, we construct long chain-of-thought (CoT) reasoning data, including node selection, workflow planning, and code-level workflow representation. ComfyUI-R1 is trained through a two-stage framework: (1) CoT fine-tuning for cold start, adapting models to the ComfyUI domain; (2) reinforcement learning for incentivizing reasoning capability, guided by a fine-grained rule-metric hybrid reward, ensuring format validity, structural integrity, and node-level fidelity. Experiments show that our 7B-parameter model achieves a 97% format validity rate, along with high pass rate, node-level and graph-level F1 scores, surpassing prior state-of-the-art methods that employ leading closed-source models such as GPT-4o and Claude series. Qualitative analysis further highlights our model’s strength in synthesizing intricate workflows with diverse nodes, aligning with human instructions, and generalizing to newly introduced nodes, underscoring the potential of long CoT reasoning in AI art creation¹.

1 Introduction

Recent advancements in large language models (LLMs) and image generation methods have democratized AI-generated content (AIGC) production (Wu and la Torre, 2023; Dhariwal and Nichol, 2021; Li et al., 2024b; Shi et al., 2025). Instead

of relying on an end-to-end diffusion model, recent AIGC production has evolved into building advanced workflows to further enhance image quality (Gal et al., 2024). Open-source frameworks such as ComfyUI (comfyanonymous, 2023) are emerging as important tools for low-code AI workflow development. Serving over 4 million users and backed by a vibrant community contributing 12K components (e.g., SDXL (Podell et al., 2023), LoRA (Ryu, 2023), ControlNet (Zhang et al., 2023), and IP-Adapter (Ye et al., 2023)), ComfyUI enables flexible workflow orchestration via drag-and-drop components for multimodal tasks.

Despite its convenient interactive interface, ComfyUI presents challenges for new users who may lack the knowledge and reasoning skills required to build effective workflows (Huang et al., 2025). The large number of nodes, along with their interdependencies and complex configurations, demands extensive documentation knowledge to go through. Moreover, the intricate connections within workflows require strong planning abilities to coordinate modules with diverse functionalities. As a result, constructing a well-designed workflow often requires considerable expertise, making automated end-to-end workflow generation and user onboarding crucial for broader adoption.

Recently, there has been increasing research on automated ComfyUI workflow generation. These approaches leverage state-of-the-art LLMs (e.g., GPT-4o) to generate valid JSON-formatted workflows and break down the task into sub-tasks for multi-agent collaboration (Xue et al., 2024; Huang et al., 2025). However, these methods still face several limitations: (1) Most studies focus only on text-to-image generation (Sobania et al., 2024; Gal et al., 2024), limiting their applicability to more conditional image and video generation tasks; (2) The generated JSON files often fail to transform into structured workflows or contain hallucinated nodes, possibly due to GPT-4o’s limited knowledge

^{*}Both authors contributed equally to this research.

[†]Corresponding author.

¹GitHub: [AIDC-AI/ComfyUI-Copilot](https://github.com/AIDC-AI/ComfyUI-Copilot)

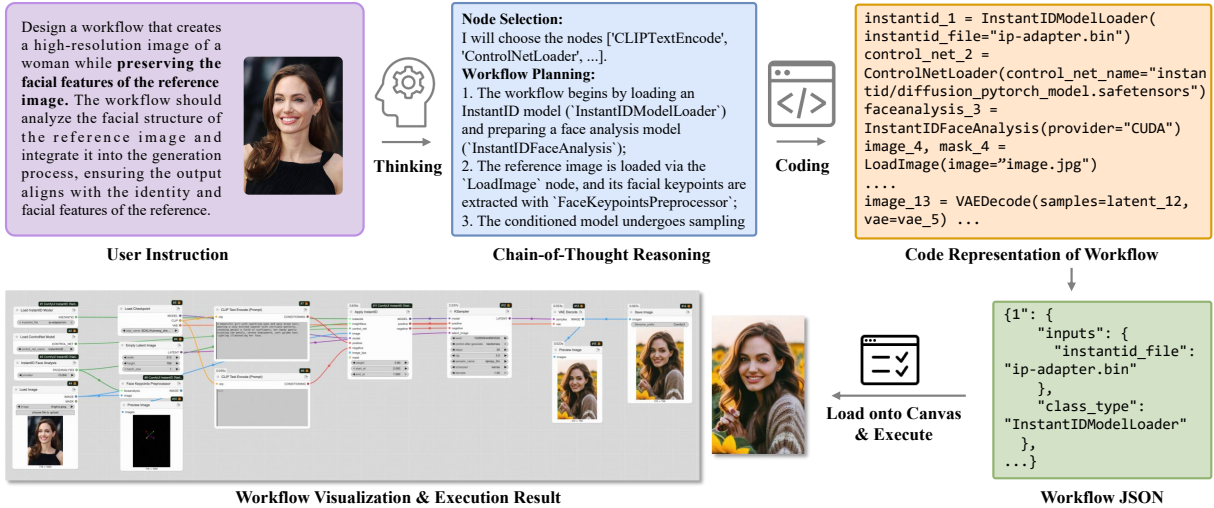


Figure 1: We introduce ComfyUI-R1, a large reasoning model for automated workflow generation. Given a user instruction, ComfyUI-R1 performs long chain-of-thought reasoning to generate a code representation of a ComfyUI workflow. The generated workflow adheres to the correct format, maintains structural integrity, executes successfully, and produces an image that aligns with the user’s instruction.

of newly introduced nodes and cumulative errors in multi-agent systems (Cemri et al., 2025; Wang et al., 2024). To address these issues, our work expands the scope to a wide range of multimodal tasks and focuses on post-training foundational LLMs rather than relying on commercial APIs.

The success of long chain-of-thought (CoT) reasoning has attracted widespread attention, exemplified by models such as OpenAI o1 (OpenAI, 2024), o3 (OpenAI, 2025b), and DeepSeek-R1 (DeepSeek-AI, 2025), which have achieved remarkable accuracy in solving Olympiad-level math problems. Inspired by this progress, we introduce **ComfyUI-R1**, the first large reasoning model designed for automated workflow generation. Specifically, we begin by collecting a substantial volume of workflows and node data from ComfyUI community websites, covering a diverse set of tasks including image, video, and 3D generation and editing. Then, we generate long CoT reasoning sequences with a simulated node retrieval process, node selection, and workflow planning, ultimately producing the code representation of the workflow.

With the data prepared, ComfyUI-R1 is trained using a two-stage pipeline: (1) Cold-start CoT fine-tuning to adapt models to the ComfyUI domain, addressing the *knowledge* gap by incorporating retrieved node documentation; (2) reinforcement learning (RL) for incentivizing *reasoning* capability in LLMs. Unlike math tasks, which typically have a single correct answer and have been extensively studied in RL settings (Li et al., 2025b;

de Winter et al., 2024), the reward design of workflow generation introduces multiple layers of complexity: the generated code should contain no hallucinated nodes, and conform to a valid format that can form a directed acyclic graph (DAG). Here we propose a fine-grained rule-metric hybrid reward. If there is anything wrong with the format validity, structural integrity, or node hallucinations, it will receive a negative reward to penalize such errors. Only when the above rules are satisfied, indicating that the workflow is well-formed, a positive reward will be given based on the correctness of node selection. We empirically demonstrate the effectiveness of this reward mechanism using Group Relative Policy Optimization (Shao et al., 2024).

Experiments on our test set show that, building upon Qwen2.5-Coder (Hui et al., 2024), our 7B ComfyUI-R1 model outperforms previous state-of-the-art methods that rely on top-tier closed-source LLMs such as GPT-4o and the Claude series. It achieves a high format validity rate of 97%, a substantial improvement over the original Qwen2.5-Coder’s 41%, showing the effectiveness of our two-stage training. We further evaluate end-to-end retrieval and generation performance on ComfyBench (Xue et al., 2024), where ComfyUI-R1 surpasses o3-mini, Gemini-2.5-pro, and the previous state-of-the-art ComfyAgent on both pass rate and solve rate. Additional analysis underscores the importance of the reasoning process, the benefit of fine-grained reward components, and the advantage of using code-based workflow representations

over JSON. Qualitative comparisons demonstrate ComfyUI-R1’s ability to synthesize complex workflows with diverse nodes that align with human instructions, and to generalize to newly introduced nodes. These results showcase the potential of long CoT reasoning in AI-driven content creation.

In summary, our contributions are as follows:

- We present ComfyUI-R1, the first large reasoning model for workflow generation to the best of our knowledge, incentivizing long CoT reasoning capability for complex workflow planning.
- We curate a comprehensive workflow and node knowledge base and develop a two-stage training framework that combines supervised fine-tuning for cold start and reinforcement learning with our proposed rule-metric hybrid reward.
- Extensive experiments show the superior performance of the 7B ComfyUI-R1 over previous state-of-the-art methods based on GPT-4o and the Claude series, underscoring the potential of long CoT reasoning in AI art creation.

2 Related Work

2.1 General Workflow Generation

General workflow generation has seen significant advancements with the advent of AI systems, such as Deep Research (Zheng et al., 2025; OpenAI, 2025a), OpenAI o3 (OpenAI, 2025b), which are capable of converting natural language descriptions into structured execution plans. Early efforts focus on human-designed workflows (Guo et al., 2024), where multiple models can collaborate and demonstrate collective intelligence in various domains, ranging from reasoning tasks (Wang et al., 2025b; Liu et al., 2025; Xu et al., 2023) and software development (Hong et al., 2024; Qian et al., 2023), to long-document translation (Wu et al., 2024) and video generation (Xu et al., 2025a; Li et al., 2024b). However, these manually crafted workflows are time-consuming, heavily dependent on domain expertise, and often lack flexibility.

To address this challenge, recent studies have focused on training models or developing language agents to automatically generate workflows (Shang et al., 2025; Niu et al., 2025; Hu et al., 2025; Xu et al., 2024; Zhang et al., 2025; Qiao et al., 2025). For example, WorkflowLLM (Fan et al., 2025) proposes a data-centric framework to enhance the workflow orchestration of LLMs by fine-tuning on complex, real-world workflows. These studies reflect a shift towards building systems that can man-

age more intricate and realistic workflow demands. The ComfyUI platform (comfyanonymous, 2023) stands out as a particularly challenging testbed in this context, due to the large number of components involved in AI art creation workflows and the intricate connections between different modules with diverse functionalities (Xue et al., 2024).

2.2 ComfyUI-based Workflow Generation

Recent advances in AI-generated content have transitioned from end-to-end diffusion models (Ho et al., 2020; Wang et al., 2025a; Podell et al., 2023; Li et al., 2024a, 2025c) to more sophisticated workflows on open-source platforms such as ComfyUI. Users can construct workflows by connecting a series of blocks, such as LLMs for refining input prompts, LoRAs trained to introduce specific artistic styles, improved latent decoders for finer details, super-resolution blocks, and more (Chen et al., 2025; Hu et al., 2021; Mañas et al., 2024; Ning et al., 2021). Creating a well-designed workflow and selecting appropriate nodes requires significant expertise (Xu et al., 2025b), where automated workflow generation comes into help.

ComfyUI-based workflow generation has emerged as a trending research topic in AI-driven multimodal content synthesis. For example, Xue et al. (2024) introduce a multi-agent framework to design workflows step-by-step, consisting of memory, planner, and action modules. Despite the advancements, the field remains in its early stages and faces two key limitations. Firstly, some existing approaches are limited to text-to-image generation (Sobania et al., 2024; Gal et al., 2024), restricting applicability to broader multimodal tasks. Secondly, the generated workflows often suffer from structural inconsistencies or contain hallucinated components, likely due to LLMs’ limited understanding of newly introduced nodes and error propagation in multi-agent frameworks (Huang et al., 2025; Guo et al., 2024). To overcome these issues, our work expands the scope to multimodal tasks and explores the post-training of foundational LLMs rather than relying on proprietary APIs, with the goal of improving robustness and generalization in automated workflow generation.

2.3 Large Reasoning Models

Since the release of OpenAI o1 (OpenAI, 2024), large reasoning models have rapidly become a focal point in AI research, simulating “System 2”

slow thinking processes (de Winter et al., 2024; Li et al., 2025d). Early attempts to replicate or rely on supervised fine-tuning (SFT) using chain-of-thought traces, augmented with tree search algorithms and self-reflection strategies (Shinn et al., 2023; Zhao et al., 2024; Li et al., 2025a; Zhang et al., 2024). Inspired by the success of DeepSeek-R1, which demonstrates strong performance on challenging benchmarks in mathematics and coding (DeepSeek-AI, 2025), recent studies have integrated the reasoning process directly into the optimization loop via group relative policy optimization (GRPO) (Shao et al., 2024). This line of work includes re-implementations of R1 (Face, 2025), extensions to multimodal settings (Zhao et al., 2025), applications in search (Song et al., 2025) and tool integration (Qian et al., 2025; Li et al., 2025b). Overall, the post-training landscape has progressed from SFT with curated rationales to reinforcement learning (RL) pipelines. In this work, we explore reasoning models designed specifically for automated workflow generation, propose a novel reward mechanism for RL in this task, and finally develop ComfyUI-R1.

3 ComfyUI-R1

ComfyUI-R1 is a large reasoning model designed for automated workflow generation. As shown in Figure 1, given a user query about task description, based on a set of retrieved nodes, ComfyUI-R1 first performs step-by-step reasoning to select relevant nodes and plan the workflow. It then generates the code representation of a ComfyUI workflow. In this section, we first introduce the construction of ComfyUI-related knowledge bases (KBs) in Sec. 3.1. Then ComfyUI-R1 is trained using a two-stage pipeline: (1) Cold-start supervised fine-tuning (SFT) with distilled long chain-of-thought (CoT) data to adapt the model to the ComfyUI domain (Sec. 3.2); (2) Reinforcement learning (RL) for incentivizing reasoning capability via a fine-grained hybrid reward (Sec. 3.3) and the GRPO algorithm (Sec. 3.4).

3.1 Knowledge Bases

We begin by constructing KBs of ComfyUI workflows and nodes. The original data is sourced from popular platforms for sharing generative resources, ComfyUI-related GitHub repositories, and the ComfyUI website, with NSFW (not safe for work) content filtered out. Since the raw data con-

| Instruction |
|--|
| <p>Create a workflow that generates a detailed and stylized image of a girl in Hanfu, set in a night scene under the moonlight. The final image should be visually rich and high-quality, capturing the specified mood and aesthetic.</p> <p>## Knowledge Base</p> <p>Here are the meta information to the nodes that may be used in the workflow:</p> <p>- Node: SaveImage</p> <pre><identifier> SaveImage </identifier> <description> Saves the input images to your ComfyUI output directory. </description> <input> [{'name': 'images', 'identifier': 'images', 'comfy_type': 'IMAGE', 'description': 'The images to',}] </input> <output> [] </output></pre> <p>- (More nodes omitted)</p> <p>## Format</p> <ol style="list-style-type: none"> Node Selection: First, reply with the names of the nodes you have chosen; Workflow Design Principle: Then, explain your thought process and reasoning; Workflow Construction: Finally, present the workflow as Python code. <p>Now, please provide your response according to the format specified above.</p> |
| Response |
| <pre><selected_nodes> I will choose the nodes in ['CLIPTextEncode', 'CheckpointLoaderSimple', 'KSampler',] </selected_nodes> <design_principle> The workflow begins by loading a pre-trained diffusion model checkpoint, which includes the model, CLIP, and VAE components. An empty latent image is created with the specified dimensions (1024x512 pixels). Finally, the VAEDecode node </design_principle> <workflow> model_1, clip_1, vae_1 = CheckpointLoaderSimple(ckpt_name="xxx.xxx") # CheckpointLoaderSimple conditioning_3 = CLIPTextEncode(clip=clip_1, text="A girl with red hair") # CLIPTextEnc image_12 = VAEDecode(samples=latent_11, vae=vae_1) # VAEDecode _ = PreviewImage(images=image_12) # PreviewImage </workflow></pre> |

Figure 2: An example of SFT data.

tains a large amount of noise, we conduct a comprehensive cleaning process to filter and standardize the data, as detailed in Appendix A. The final KB includes 7,238 nodes and 3,917 workflows. As illustrated in Figure 4 in the Appendix, our workflow KB encompasses a diverse set of tasks, including text-to-image generation, image editing, style transfer, video editing and generation.

3.2 Supervised Fine-tuning

We generate long CoT reasoning sequences via a simulated node retrieval process, node selection, and workflow planning, ultimately producing a code representation of the workflow. Each entry in the workflow KB is represented as a pair $(desc, c)$, where $desc$ is the workflow description and c is its corresponding code representation. Let \mathcal{V}^g denote the set of nodes in the ground-truth workflow, and \mathcal{V}^{KB} denote the full set of nodes in the KB. To simulate the node retrieval process, we construct a candidate node set \mathcal{V}^{cand} by combining \mathcal{V}^g with a randomly sampled set of nodes \mathcal{V}^{random} , where $\mathcal{V}^{random} \subset \mathcal{V}^{KB} \setminus \mathcal{V}^g$, and:

$$\mathcal{V}^{cand} = \mathcal{V}^g \cup \mathcal{V}^{random}, \text{ with } |\mathcal{V}^{random}| = \lfloor 0.8 \cdot |\mathcal{V}^g| \rfloor.$$

Based on the workflow description $desc$ and its corresponding code representation c , we gener-

ate user instructions $query$ and rationales r about the workflow design principle using Qwen-Max, Claude 3.5 and GPT-4o. We split 3,717 workflows into training and 200 for testing. This results in 11,140 training samples and 600 testing samples. As shown in Figure 2, each sample consists of the user instruction $query$, a set of candidate nodes \mathcal{V}^{cand} , the workflow planning r , the nodes in the gold workflow \mathcal{V}^g , and the final code c .

During SFT, the model input includes the $query$ and the candidate nodes \mathcal{V}^{cand} . The output is a long CoT reasoning sequence, denoted as $s = [\mathcal{V}^g, r, c]$. The training objective can be formulated as:

$$\mathcal{L}_{SFT} = -\log \sum_{t=1}^T \Pr(s_t = i \mid desc, \mathcal{V}^{cand}, s_{<t})$$

where $s_{<t}$ is the sequence of tokens generated before time step t , $\Pr(s_t = i \mid desc, \mathcal{V}^{cand}, s_{<t})$ is the probability that the LLM predicts token s_t in step t .

3.3 Reward Design

After adapting LLMs to the ComfyUI domain via SFT, we conduct RL to further enhance the reasoning capability. For an RL algorithm to be effective, a well-designed **reward** is essential. Unlike math tasks in prior work—which typically have a single correct answer and allow for simple rule-based rewards—the workflow generation task presents several layers of complexity. For example, the generated code should avoid hallucinated nodes, and adhere to a valid format that forms a directed acyclic graph (DAG). To this end, here we propose a fine-grained rule-metric hybrid reward R_{final} , including format reward R_{format} , structure reward R_{DAG} , node fidelity $R_{fidelity}$ and precision reward $R_{correct}$. This reward formulation penalizes invalid formats, incorrect graph structures, and hallucinated nodes, while awarding accurate node selection.

Format reward (R_{format}). This component verifies whether the output response adheres to the expected structure, including a sequence of reasoning steps enclosed within the “<selected_nodes>...</selected_nodes>” and “<design_principle>...</design_principle>” tags. The final workflow code is wrapped within “<workflow>...</workflow>”. A format score of 0 is given if all required tags are present and their contents can be successfully extracted. Otherwise, a penalty is applied. The format reward function R_{format} is

defined as follows:

$$R_{format} = \begin{cases} 0 & \text{if all required fields appear} \\ -1 & \text{otherwise} \end{cases}$$

Structure reward (R_{DAG}). This reward assesses whether the parsed “<workflow>” section forms a valid DAG, which is a fundamental requirement for ComfyUI workflows. A penalty is assigned if the structure is not a valid DAG.

$$R_{DAG} = \begin{cases} 0 & \text{if the structure is a valid DAG} \\ -1 & \text{otherwise} \end{cases}$$

Node fidelity reward ($R_{fidelity}$). This component penalizes the inclusion of hallucinated or inconsistent nodes during node selection and workflow generation. A penalty of -1 is applied in either of the following cases:

- **Invalid Nodes:** Any node listed in the predicted “<selected_nodes>” block \mathcal{V}^p is not present in the provided candidate nodes \mathcal{V}^{cand} .
- **Inconsistent Nodes:** The set of nodes listed in “<selected_nodes>” block does not exactly match the set of nodes parsed from the “<workflow>” block.

$$R_{fidelity} = \begin{cases} -1, & \text{if invalid or inconsistent} \\ & \text{nodes are detected} \\ 0, & \text{otherwise} \end{cases}$$

Node Selection Accuracy ($R_{correct}$). This reward measures the overlap between generated node set \mathcal{V}^p and the ground truth node set \mathcal{V}^g , reflecting the model’s ability to select the correct nodes.

$$R_{correct} = \frac{|\mathcal{V}^p \cap \mathcal{V}^g|}{|\mathcal{V}^g|} - 1$$

Combining the above rewards The total reward R_{final} aggregates all the individual reward components using a veto-based mechanism. If any of R_{format} , R_{DAG} , or $R_{fidelity}$ is -1, indicating a fundamental error in format validity, structural integrity, or node fidelity, the total reward is immediately set to -1, preventing any positive reward for an invalid output. Otherwise, the total reward is computed based on the node selection accuracy.

$$R_{final} = \begin{cases} -1, & \text{if } R_{format} = -1 \text{ or } R_{DAG} = -1 \\ & \text{or } R_{fidelity} = -1 \\ \frac{4 + R_{correct}}{4.0}, & \text{otherwise} \end{cases}$$

| Methods | Models | Format Validity | Node-level | | | Graph-level | | |
|-------------------|---------------------------|-----------------|------------|------|-------------|-------------|------|-------------|
| | | | P | R | F1 | P | R | F1 |
| Few-shot | Qwen2.5-Coder-7B-Instruct | 0.25 | 0.18 | 0.12 | 0.14 | 0.08 | 0.07 | 0.08 |
| | Qwen2.5-Max | 0.50 | 0.35 | 0.25 | 0.29 | 0.19 | 0.17 | 0.18 |
| | GPT-4o | 0.89 | 0.62 | 0.42 | 0.50 | 0.33 | 0.28 | 0.30 |
| | Claude 3.5 Sonnet | 0.93 | 0.58 | 0.48 | 0.52 | 0.37 | 0.36 | 0.37 |
| | Claude 3.7 Sonnet | 0.81 | 0.44 | 0.43 | 0.43 | 0.27 | 0.29 | 0.28 |
| CoT | Qwen2.5-Coder-7B-Instruct | 0.41 | 0.30 | 0.18 | 0.22 | 0.12 | 0.09 | 0.10 |
| | Qwen2.5-Max | 0.64 | 0.46 | 0.31 | 0.36 | 0.25 | 0.21 | 0.23 |
| | GPT-4o | 0.92 | 0.66 | 0.42 | 0.50 | 0.33 | 0.27 | 0.29 |
| | Claude 3.5 Sonnet | 0.97 | 0.70 | 0.49 | 0.57 | 0.41 | 0.36 | 0.38 |
| | Claude 3.7 Sonnet | 0.90 | 0.57 | 0.48 | 0.51 | 0.36 | 0.35 | 0.35 |
| ComfyAgent | GPT-4o | 0.47 | 0.26 | 0.20 | 0.21 | 0.11 | 0.10 | 0.10 |
| SFT + GRPO | ComfyUI-R1 | 0.97 | 0.67 | 0.58 | 0.62 | 0.52 | 0.51 | 0.51 |

Table 1: **Evaluation results of all baselines on our test set.** The node-level and graph-level *precision*, *recall* and *F1* scores, together with *Format Validity* rate, are reported. The best results are highlighted in **bold**.

3.4 RL Training with GRPO

Based on the above hybrid reward, ComfyUI-R1 is trained using the Group Relative Policy Optimization (GRPO) algorithm (Shao et al., 2024). At each training iteration, given an input text q (including the *query* and the candidate nodes $\mathcal{V}^{\text{cand}}$), a group of G candidate outputs, $\{s_1, s_2, \dots, s_G\}$, is sampled from the policy model $\pi_{\theta_{\text{old}}}$. The advantage is computed based on the group of hybrid rewards $\{r_1, r_2, \dots, r_G\}$, defined as

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \dots, r_G\})}{\text{std}(\{r_1, r_2, \dots, r_G\})},$$

The GRPO algorithm then optimizes the policy π_{θ} by maximizing the following objective function:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q \sim P(Q), \{s_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(O|q)} \left[\frac{1}{G} \sum_{i=1}^G \min \left(\frac{\pi_{\theta}(s_i | q)}{\pi_{\theta_{\text{old}}}(s_i | q)} A_i, \text{clip} \left(\frac{\pi_{\theta}(s_i | q)}{\pi_{\theta_{\text{old}}}(s_i | q)}, 1 - \varepsilon, 1 + \varepsilon \right) A_i \right) - \beta D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}}) \right]$$

$$D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(o_i | q)}{\pi_{\theta}(o_i | q)} - \log \frac{\pi_{\text{ref}}(o_i | q)}{\pi_{\theta}(o_i | q)} - 1,$$

where ε specifies clipping range, and β is a hyperparameter controlling the weight of the Kullback–Leibler (KL) divergence penalty.

4 Experiments

4.1 Data and Evaluation Metric

Our experiments are conducted on the test set described in Sec. 3.2. Each input consists of a *query*

and a set of candidate nodes $\mathcal{V}^{\text{cand}}$. The expected output is an executable workflow c that satisfies the query. Our evaluation setup is designed to assess the model’s ability to reason over pre-retrieved node information, rather than its retrieval capability. To assess end-to-end performance—including both node retrieval and workflow generation—we conduct an additional experiment in Sec. 4.4, where no candidate set $\mathcal{V}^{\text{cand}}$ is provided. Implementation details are in Appendix B.

The first evaluation metric is **Format Validity** rate. It provides an initial check of the syntactic and structural correctness of the generated workflows. Specifically, it verifies whether all node names referenced in the workflow exist and whether the resulting structure forms a valid DAG.

Following the evaluation protocol in WorFEval (Qiao et al., 2025), we quantitatively assess the matching of the predicted and gold workflows. Specifically, we identify the longest node chain via Longest Increasing Subsequence (LIS) and the largest workflow subgraph via Maximum Common Induced Subgraph (MCIS). Based on these matches, we compute and report **node-level and graph-level precision, recall and F1 scores**².

4.2 Baselines

Following previous workflow generation studies (Huang et al., 2025; Xue et al., 2024), we adopt two effective prompting methods for advanced commercial LLMs (Qwen-Max, Claude 3.5, Claude 3.7 and GPT-4o), as well as Qwen2.5-

²For detailed derivations and formal equations of the evaluation metrics, please refer to the original WorFEval paper.

| Methods | FV | Node-level | | | Graph-level | | |
|-------------------------|-------------|------------|------|-------------|-------------|------|-------------|
| | | P | R | F1 | P | R | F1 |
| ComfyUI-R1 | 0.97 | 0.67 | 0.58 | 0.62 | 0.52 | 0.51 | 0.51 |
| SFT only (Code) | 0.95 | 0.64 | 0.57 | 0.60 | 0.48 | 0.49 | 0.48 |
| SFT only (JSON) | 0.92 | 0.62 | 0.55 | 0.57 | 0.45 | 0.46 | 0.45 |
| - R_{fidelity} | 0.95 | 0.67 | 0.53 | 0.59 | 0.45 | 0.42 | 0.43 |
| - R_{correct} | 0.96 | 0.70 | 0.51 | 0.58 | 0.43 | 0.38 | 0.40 |

Table 2: **Ablation results of ComfyUI-R1** with different settings. ‘‘FV’’ means format validity.

Coder-7B-Instruct, which serves as the base model for our training.

1. **Few-shot learning** provides a set of code-formatted workflows in the prompt to utilize the in-context learning capabilities of LLMs for workflow generation.
2. **Chain-of-Thought (CoT)** is improved based on the above few-shot learning baseline, incorporating the CoT reasoning process into the few-shot demonstrations.

In addition, we include the multi-agent baseline **ComfyAgent** (Xue et al., 2024) based on its official GitHub implementation.

4.3 Results

Table 1 presents the results on our test set. In terms of format validity rate, node-level and graph-level F1 scores, our 7B ComfyUI-R1 model outperforms few-shot, CoT prompting and multi-agent methods, which all rely on top-tier closed-source LLMs such as GPT-4o and the Claude series. Compared to the original Qwen2.5-Coder model, ComfyUI-R1 achieves substantial improvements across all metrics (e.g., format validity rate increases from 41% to 97%), demonstrating the effectiveness of our two-stage training strategy. When comparing performance under few-shot and CoT settings, all models show consistent improvement with CoT reasoning in their in-context demonstrations, highlighting the importance of the reasoning process.

Table 2 reports results from two SFT variants of ComfyUI-R1: (1) an SFT-only version, and (2) a version that replaces code-based workflow representations with a JSON format proposed by Huang et al. (2025). The results show that RL training further improves the already high 95% format validity rate of the SFT-only model, validating the effectiveness of RL in the workflow generation task. On the other hand, switching from code to JSON for workflow representation leads to decreased performance, likely because the JSON format carries less

| Methods | FV | Pass | Solve |
|---------------------|-------------|-------------|-------------|
| ComfyAgent (GPT-4o) | 0.52 | 0.33 | 0.26 |
| GPT-4o | 0.93 | 0.62 | 0.36 |
| o3-mini | 0.95 | 0.69 | 0.43 |
| Gemini-2.5-pro | 1.00 | 0.71 | 0.48 |
| ComfyUI-R1 | 1.00 | 0.81 | 0.57 |

Table 3: **Format validity, pass rate, and solve rate on ComfyBench.** All methods except ComfyAgent follow the same node retrieval process.

semantic and logical structure. Code-based representations are more compact and semantically rich, highlighting their advantage for this task.

We have also conducted an ablation study in reward design in Table 2. When either the node hallucination penalty (R_{fidelity}) or the correctness reward (R_{correct}) is removed, performance drops significantly. This demonstrates that penalizing hallucinated or incorrectly selected nodes is crucial for generating well-structured and correctly executed workflows.

4.4 Additional Experiments on ComfyBench

To evaluate the end-to-end retrieval and generation performance, we conduct experiments on ComfyBench (Xue et al., 2024) without providing candidate nodes $\mathcal{V}^{\text{cand}}$. ComfyBench is a recently proposed benchmark containing challenging instructions that require complex ComfyUI workflows. We randomly select 42 examples from ComfyBench spanning text-to-image, image editing, text-to-video and image-to-video. We report the *format validity*, *pass rate* and *solve rate*. The latter two metrics are annotated by experienced ComfyUI developers. *Pass rate* measures the ratio of tasks where generated workflows can be successfully loaded and executed in the ComfyUI server. *Solve rate* measures the ratio of tasks where the generated content after execution satisfies the instructions.

Our retrieval process for obtaining candidate nodes $\mathcal{V}^{\text{cand}}$ is as follows: (1) Given an instruction, we retrieve the top 3 most semantically relevant workflows from the workflow KB based on OpenAI’s text-embedding-3-small. (2) We then aggregate the nodes from these retrieved workflows into a set, which forms $\mathcal{V}^{\text{cand}}$. Next, the *query* and candidate nodes are input into ComfyUI-R1 for workflow generation. All baselines except ComfyAgent undergo the same node retrieval process. Results in Table 3 show that the workflows generated by our model achieve the highest pass rate

User Instruction: Create an anime-style portrait of a nurse character with pink hair. The image should focus on the upper body, showing the character looking at the viewer with a smile while wearing a nurse cap. The final image should have high resolution and maintain a cartoon aesthetic style.

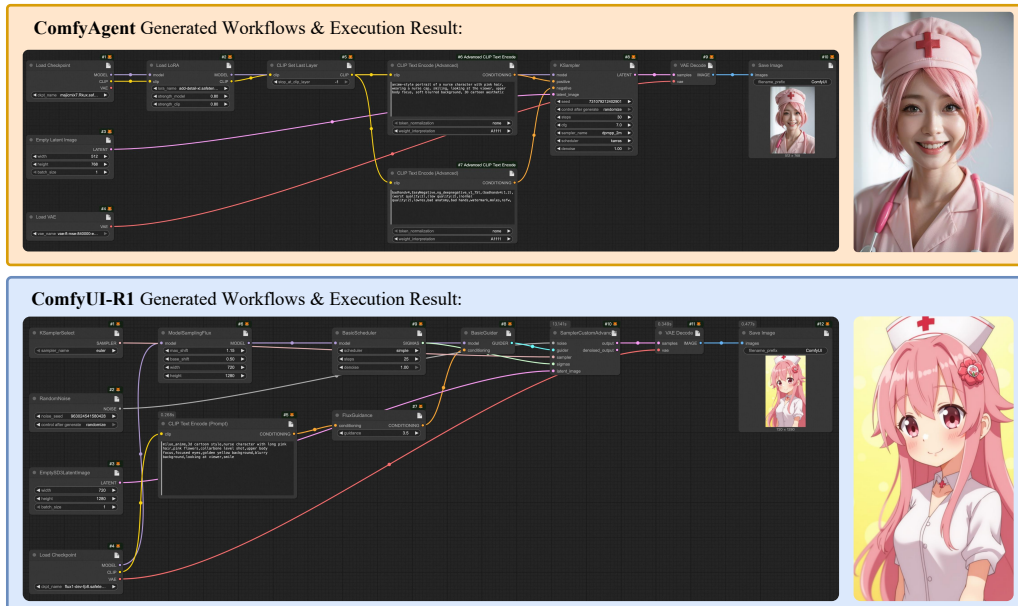


Figure 3: **Comparison between ComfyAgent and our ComfyUI-R1.** The execution result of ComfyUI-R1 accurately adheres to the “anime-style” and “cartoon” attributes in the user instruction. In contrast, ComfyAgent fails to follow these stylistic guidelines.

and task completion rate, outperforming previous state-of-the-art ComfyAgent and leading reasoning models such as o3-mini and Gemini-2.5-pro, showing the effectiveness of our training scheme tailored for the workflow generation task. Sec. 4.5 provides more qualitative comparison of our ComfyUI-R1 and other baselines.

4.5 Case Study

We present a comparison between ComfyAgent and our ComfyUI-R1. In Figure 3, the image style generated by ComfyUI-R1’s workflow aligns more closely with the user instruction compared to that produced by ComfyAgent. From the multi-image combination example in Figures 5 and 6, ComfyUI-R1 successfully loads and seamlessly blends the two input images. In contrast, ComfyAgent’s workflow loads the second image but fails to utilize it further, resulting in an incomplete output. This highlights the importance of effective workflow planning. From these examples, we observe that ComfyUI-R1’s workflows typically include more nodes than those of ComfyAgent, demonstrating ComfyUI-R1’s capability to synthesize complex, executable, and instruction-aligned workflows with diverse nodes. In Appendix C, we present more side-by-side visual comparisons of ComfyUI-R1

and other baselines in Figure 7. During human evaluation, we find that ComfyUI-R1 performs better at node selection and linking, while other models tend to generate hallucinated or isolated nodes and subgraphs. In Figure 8, ComfyUI-R1 shows the capability to go beyond simple memorization and generalize to newly-introduced ComfyUI nodes (e.g., API nodes released on May 6th, 2025).

5 Conclusion

We introduce **ComfyUI-R1**, the first large reasoning model for automated workflow generation on the ComfyUI platform. We employ supervised fine-tuning for cold start and reinforcement learning for incentivizing reasoning capability, guiding the model to generate structurally sound and executable workflows. Experimental results demonstrate that our 7B ComfyUI-R1 model significantly outperforms existing SOTA methods based on GPT-4o and Claude series, achieving a 97% format validity rate and superior node-level and graph-level F1 scores. Qualitative analysis highlights the model’s ability to synthesize complex workflows, showcasing the power of structured reasoning in AI-driven content creation. This work underscores the potential of reasoning-based approaches in lowering the barrier for advanced AI tools.

Limitations

Although ComfyUI-R1 has significantly improved workflow generation, there are certain limitations to our approach. First, the workflows generated by our model rely on default parameter settings for nodes, which may require manual adjustment to achieve optimal visual results. One potential solution is to employ large multimodal models as evaluators to provide feedback, enabling a closed-loop system where initial workflows could be refined iteratively based on evaluations, ultimately leading to better results (Guo et al., 2025). Second, more fine-grained reward signals, such as parameter name and content matching, could be incorporated. Future work may explore reward designs to better guide the intricate reasoning required for workflow generation.

Ethics Statement

We do not use any personally identifiable data or copyrighted material in our training or evaluation. All knowledge bases are either curated from public documentation, synthetically generated, or derived from open-source platforms under permissive licenses. We encourage responsible deployment of our work, especially in preventing misuse such as deepfakes or deceptive content.

Acknowledgments

We thank the reviewers for their efforts in improving our paper. This work was supported by National Natural Science Foundation of China (Grant No. 62422603 and No. 625B2061), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024B0101050003), and Shenzhen Science and Technology Program (Grant No. ZDSYS20230626091203008).

References

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025. [Why do multi-agent llm systems fail?](#) *Preprint*, arXiv:2503.13657.

Yulin Chen, Haoran Li, Yuan Sui, Yufei He, Yue Liu, Yangqiu Song, and Bryan Hooi. 2025. [Can indirect prompt injection attacks be detected and removed?](#) In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18189–18206, Vienna, Austria. Association for Computational Linguistics.

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

Joost C. F. de Winter, Dimitra Dodou, and Yke Bauke Eisma. 2024. [System 2 thinking in openai’s o1-preview model: Near-perfect performance on a mathematics exam.](#) *CoRR*, abs/2410.07114.

DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.](#) *Preprint*, arXiv:2501.12948.

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.

Hugging Face. 2025. open-r1. <https://github.com/huggingface/open-r1>. GitHub Repository.

Shengda Fan, Xin Cong, Yuepeng Fu, Zhong Zhang, Shuyan Zhang, Yuanwei Liu, Yesai Wu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2025. [WorkflowLLM: Enhancing workflow orchestration capability of large language models.](#) In *The Thirteenth International Conference on Learning Representations*.

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.

Litao Guo, Xinli Xu, Luozhou Wang, Jiantao Lin, Jinsong Zhou, Zixin Zhang, Bolan Su, and Ying-Cong Chen. 2025. [Comfymind: Toward general-purpose generation via tree-based planning and reactive feedback.](#) *Preprint*, arXiv:2505.17908.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xi-angliang Zhang. 2024. [Large language model based multi-agents: A survey of progress and challenges.](#) *Preprint*, arXiv:2402.01680.

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.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. [MetaGPT: Meta programming for a multi-agent collaborative framework.](#) In *The Twelfth International Conference on Learning Representations*.

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*.

- Shengran Hu, Cong Lu, and Jeff Clune. 2025. [Automated design of agentic systems](#). In *The Thirteenth International Conference on Learning Representations*.
- Oucheng Huang, Yuhang Ma, Zeng Zhao, Mingrui Wu, Jiayi Ji, Rongsheng Zhang, Zhipeng Hu, Xiaoshuai Sun, and Rongrong Ji. 2025. [Comfygpt: A self-optimizing multi-agent system for comprehensive comfyui workflow generation](#). *Preprint*, arXiv:2503.17671.
- Binyuan Hui, Jian Yang, Zeyu Cui, Jiayi 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.
- Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025a. [Search-o1: Agentic search-enhanced large reasoning models](#). *arXiv preprint arXiv:2501.05366*.
- Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025b. [Torl: Scaling tool-integrated rl](#). *Preprint*, arXiv:2503.23383.
- Yunxin Li, Xinyu Chen, Shenyuan Jiang, Haoyuan Shi, Zhenyu Liu, Xuanyu Zhang, Nanhao Deng, Zhenran Xu, Yicheng Ma, Meishan Zhang, Baotian Hu, and Min Zhang. 2025c. [Uni-moe-2.0-omni: Scaling language-centric omnimodal large model with advanced moe, training and data](#). *Preprint*, arXiv:2511.12609.
- Yunxin Li, Shenyuan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. 2024a. [Uni-moe: Scaling unified multimodal llms with mixture of experts](#). *Preprint*, arXiv:2405.11273.
- Yunxin Li, Zhenyu Liu, Zitao Li, Xuanyu Zhang, Zhenran Xu, Xinyu Chen, Haoyuan Shi, Shenyuan Jiang, Xintong Wang, Jifang Wang, Shouzheng Huang, Xinpeng Zhao, Borui Jiang, Lanqing Hong, Longyue Wang, Zhuotao Tian, Baoxing Huai, Wenhan Luo, Weihua Luo, Zheng Zhang, Baotian Hu, and Min Zhang. 2025d. [Perception, reason, think, and plan: A survey on large multimodal reasoning models](#). *Preprint*, arXiv:2505.04921.
- Yunxin Li, Haoyuan Shi, Baotian Hu, Longyue Wang, Jiashun Zhu, Jinyi Xu, Zhen Zhao, and Min Zhang. 2024b. [Anim-director: A large multimodal model powered agent for controllable animation video generation](#). In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA. Association for Computing Machinery.
- Yexiang Liu, Jie Cao, Zekun Li, Ran He, and Tieniu Tan. 2025. [Breaking mental set to improve reasoning through diverse multi-agent debate](#). In *The Thirteenth International Conference on Learning Representations*.
- Oscar Mañas, Pietro Astolfi, Melissa Hall, Candace Ross, Jack Urbanek, Adina Williams, Aishwarya Agrawal, Adriana Romero-Soriano, and Michal Drozdal. 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.
- Boye Niu, Yiliao Song, Kai Lian, Yifan Shen, Yu Yao, Kun Zhang, and Tongliang Liu. 2025. [Flow: Modularized agentic workflow automation](#). In *The Thirteenth International Conference on Learning Representations*.
- OpenAI. 2024. [Introducing openai o1-preview](#). <https://openai.com/index/introducing-openai-o1-preview/>.
- OpenAI. 2025a. [Introducing deep research](#). Accessed: 2025-02-02.
- OpenAI. 2025b. [Introducing openai o3 and o4-mini](#). Accessed: 2025-04-18.
- 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.
- Chen Qian, Xin Cong, Wei Liu, Cheng Yang, Weize Chen, Yusheng Su, Yufan Dang, Jiahao Li, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. [Communicative agents for software development](#). *Preprint*, arXiv:2307.07924.
- Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiushi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. [Toolrl: Reward is all tool learning needs](#). *Preprint*, arXiv:2504.13958.
- Shuofei Qiao, Runnan Fang, Zhisong Qiu, Xiaobin Wang, Ningyu Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2025. [Benchmarking agentic workflow generation](#). *Preprint*, arXiv:2410.07869.
- Simo Ryu. 2023. [Low-rank adaptation for fast text-to-image diffusion fine-tuning](#). <https://github.com/cloneofsimo/lora>.
- Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. 2025. [Agentsquare: Automatic LLM agent search in modular design space](#). In *The Thirteenth International Conference on Learning Representations*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.

- Haoyuan Shi, Yunxin Li, Xinyu Chen, Longyue Wang, Baotian Hu, and Min Zhang. 2025. [Animaker: Multi-agent animated storytelling with mcts-driven clip generation](#). In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, SA Conference Papers '25, New York, NY, USA. Association for Computing Machinery.
- 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.
- Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Jirong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.
- Jifang Wang, Xue Yang, Longyue Wang, Zhenran Xu, Yiyu Wang, Yaowei Wang, Weihua Luo, Kaifu Zhang, Baotian Hu, and Min Zhang. 2025a. [A unified agentic framework for evaluating conditional image generation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12626–12646, Vienna, Austria. Association for Computational Linguistics.
- Junlin Wang, Jue WANG, Ben Athiwaratkun, Ce Zhang, and James Zou. 2025b. [Mixture-of-agents enhances large language model capabilities](#). In *The Thirteenth International Conference on Learning Representations*.
- Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024. [Rethinking the bounds of LLM reasoning: Are multi-agent discussions the key?](#) In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6106–6131, Bangkok, Thailand. Association for Computational Linguistics.
- Chen Henry Wu and Fernando De la Torre. 2023. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *ICCV*.
- Minghao Wu, Yulin Yuan, Gholamreza Haffari, and Longyue Wang. 2024. [\(perhaps\) beyond human translation: Harnessing multi-agent collaboration for translating ultra-long literary texts](#). *Preprint*, arXiv:2405.11804.
- Jia Xu, Weilin Du, Xiao Liu, and Xuejun Li. 2024. [Llm4workflow: An llm-based automated workflow model generation tool](#). In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering, ASE '24*, page 2394–2398, New York, NY, USA. Association for Computing Machinery.
- 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, Longyue Wang, Jifang Wang, Zhouyi Li, Senbao Shi, Xue Yang, Yiyu Wang, Baotian Hu, Jun Yu, and Min Zhang. 2025a. [Filmagent: A multi-agent framework for end-to-end film automation in virtual 3d spaces](#). *Preprint*, arXiv:2501.12909.
- Zhenran Xu, Yangxue Yangxue, Yiyu Wang, Qingli Hu, Zijiao Wu, Baotian Hu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025b. [ComfyUI-copilot: An intelligent assistant for automated workflow development](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 632–643, Vienna, Austria. 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.
- Hu Ye, Jun Zhang, Sibao 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*.
- Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, et al. 2024. [Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning](#). *arXiv preprint arXiv:2410.02884*.
- Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. 2025. [AFlow: Automating agentic workflow generation](#). In *The Thirteenth International Conference on Learning Representations*.
- Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. [Adding conditional control to text-to-image diffusion models](#). In *CVPR*.
- Jiaxing Zhao, Xihan Wei, and Liefeng Bo. 2025. [R1-omni: Explainable omni-multimodal emotion recognition with reinforcement learning](#). *Preprint*, arXiv:2503.05379.
- Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024. [Marco-o1: Towards open reasoning models for open-ended solutions](#). *CoRR*, abs/2411.14405.
- Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. [Deepresearcher: Scaling deep research via reinforcement learning in real-world environments](#). *arXiv preprint arXiv:2504.03160*.

A Details of Knowledge Bases

Node KB Starting with a collection of 40K nodes, we first apply exact-match deduplication, and then filter out nodes that lack input or output parameters. For nodes without structured documentation, following Xu et al. (2025b), we use Claude 3.5 to generate detailed documentation by analyzing their GitHub repositories. The final documentation is standardized to include the node type, usage, and the meanings of input and output parameters.

Workflow KB Workflows in ComfyUI are natively represented on the canvas in JSON format. We implement a parser to extract a Directed Acyclic Graph (DAG) from the JSON file, which is then converted into a sequence of Python-like function calls in topological order. Another parser is implemented to reverse the conversion, ensuring that workflows can be expressed through code and remain compatible with the ComfyUI environment.

Starting with the collected 27K workflows, we perform a series of filtering and refinement steps. First, we discard workflows that do not meet ComfyUI’s execution standards. Second, we remove exact duplicates. Third, we verify whether each workflow can be successfully transformed between JSON and code representations, retaining only those that support both conversions. Next, we further refine workflows by removing irrelevant information, such as “Anything Anywhere” node. Finally, we discard workflows that contain nodes not present in the node KB. After this cleaning process, the final workflow KB contains 3,917 workflows, with an average of 21 nodes per workflow.

Since community-sourced content tends to focus more on installation instructions, there is often a lack of detailed functional descriptions for workflows. To address this, we leverage the multimodal understanding capabilities of GPT-4o, by prompt-

ing it with community-sourced texts and accompanying images that typically illustrate workflow outcomes. This approach helps fill in the gaps in informative functional descriptions. Therefore, each entry in our workflow KB includes a JSON-formatted and code-based representation of the workflow, along with an explanation of its functionality. As shown in Figure 4, our workflow KB encompasses a diverse set of tasks, including text-to-image generation, image editing, style transfer, video editing and generation.

B Implementation Details

We use the Qwen2.5-Coder-7B-Instruct as the backbone of ComfyUI-R1. The SFT training stage is conducted on $8 \times 80\text{G}$ NVIDIA A100 GPUs over 1 epoch, with a learning rate of $1e-5$ and a batch size of 1 on each GPU. During RL training, we utilize $8 \times 80\text{G}$ NVIDIA A100 GPUs to train for a total of 300 steps, with a learning rate of $1e-6$ and a total batch size of 64. In the GRPO equation in Sec. 3.4, the number of group computations G is set to 4, the clipping coefficient ε to 0.2, the KL penalty weight β to 0.001. The maximum tokens are set to 32,768 for both training and inference.

C More Case Study

In Appendix C, we present a multi-image combination example in Figures 5 and 6. We present more side-by-side visual comparisons of ComfyUI-R1 and other baselines in Figure 7. In Figure 8, ComfyUI-R1 shows the capability to generalize to the newly-introduced API node “FluxKontextImageNode”.

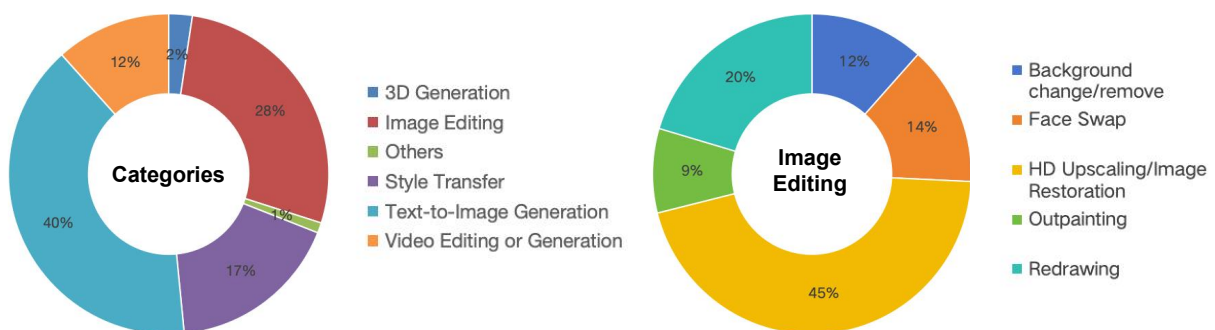


Figure 4: An illustration of the task categories in our constructed workflow KB, including the subcategories under image editing.

User Instruction: Create a workflow that can seamlessly combine two images into one wider image by intelligently generating a transition area between them. The workflow should analyze the style and content of the first image and use it to create a natural-looking bridge to the second image. The result should look like one cohesive image rather than two separate images placed side by side.



ComfyUI-R1 Generated Workflows & Execution Result:

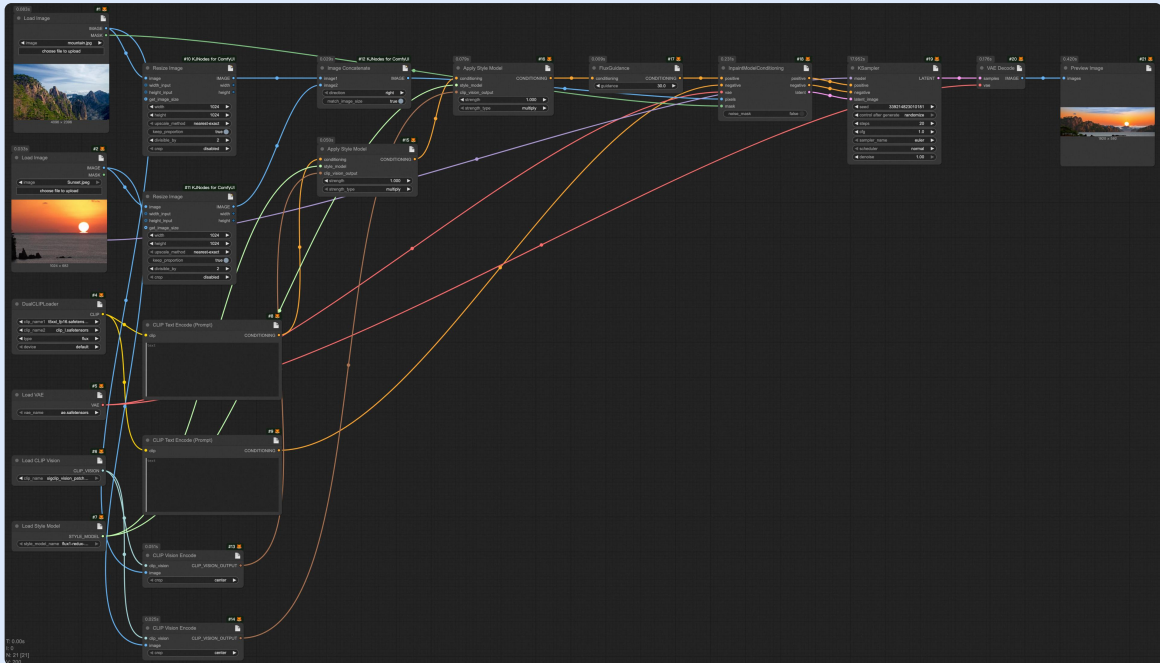


Figure 5: **Case study: multi-image combination.** Here ComfyUI-R1 seamlessly combines the input images by generating a correct and executable workflow.

ComfyAgent Generated Workflows & Execution Result:

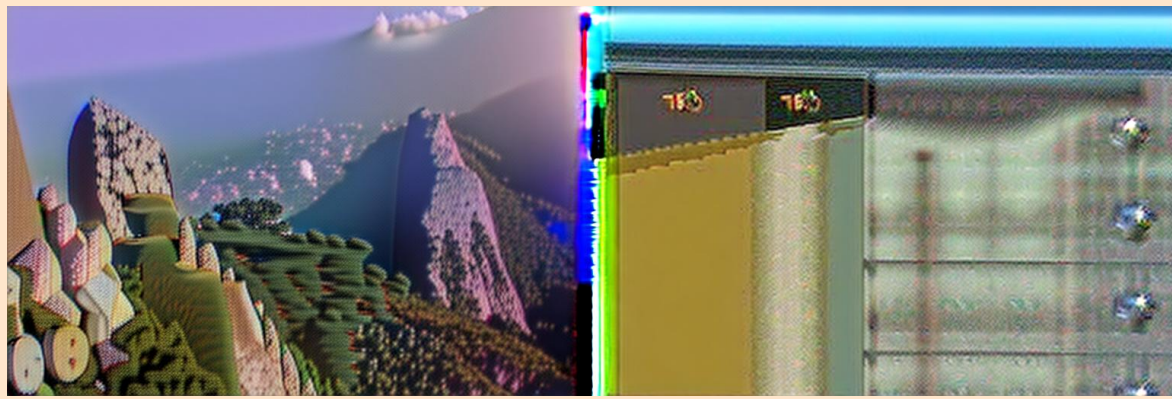
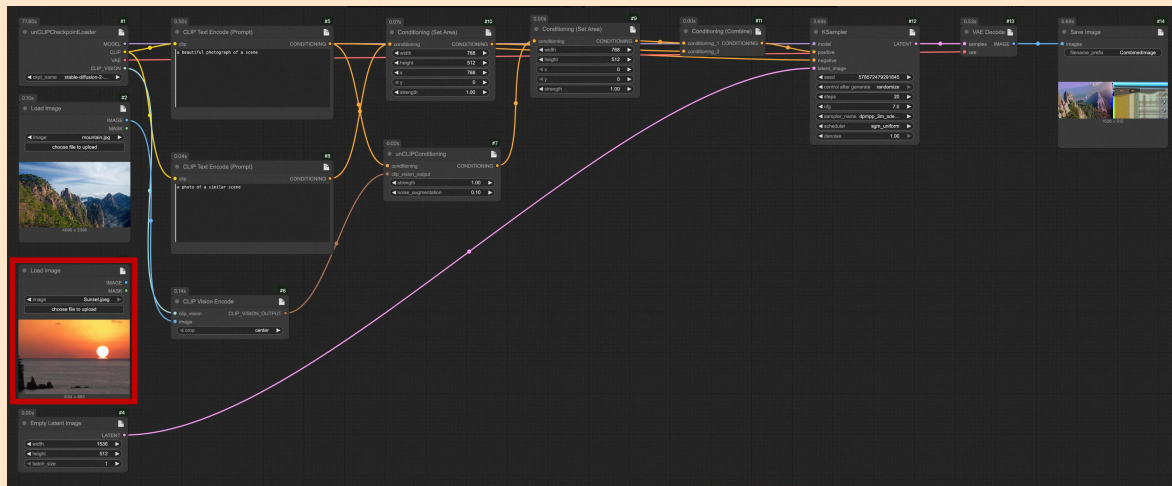


Figure 6: **Case study: multi-image combination (continued)**. This figure shows the workflow and the corresponding execution result generated by ComfyAgent (Xue et al., 2024).












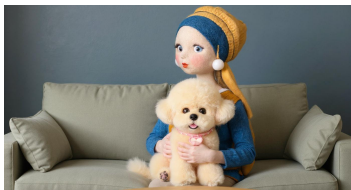
| Instruction | ComfyUI-R1 (ours) | GPT-4o | Gemini-2.5-pro |
|--|--|---|--|
| Generate an image of a woman riding a motorcycle on a highway. |  |  |  |
| Generate an image of a rainy city street at night with reflections of neon signs on the wet pavement. |  |  | Failed to Execute |
| First generate an image of a pagoda in a quiet forest. Then replace the background with a scene of a mountain landscape. |  |  |  |

Figure 7: Visual comparison of workflows generated by ComfyUI-R1, GPT-4o, and Gemini-2.5-pro in Table 3.

A woman is sitting on a couch holding a dog.


→


```

IMAGE_1, MASK_1 = LoadImage(image="person.jpg") # LoadImage
IMAGE_2, MASK_2 = LoadImage(image="furniture.jpg") # LoadImage
IMAGE_3, MASK_3 = LoadImage(image="pet.jpg") # LoadImage
IMAGE_4 = ImageStitch(direction="up", match_image_size=True, spacing_width=8,
spacing_color="white", image1=IMAGE_2, image2=IMAGE_3) # ImageStitch
IMAGE_5 = ImageStitch(direction="right", match_image_size=True, spacing_width=8,
spacing_color="white", image1=IMAGE_1, image2=IMAGE_4) # ImageStitch
IMAGE_6 = FluxKontextProImageNode(prompt="A woman is sitting on a couch holding a dog",
aspect_ratio="16:9", guidance=3, steps=50, seed=42, prompt_upsampling=False,
input_image=IMAGE_5) # FluxKontextProImageNode
_ = SaveImage(filename_prefix="ComfyUI", images=IMAGE_6) # SaveImage

```

Figure 8: An example of ComfyUI-R1 using newly-introduced API node “FluxKontextImageNode”.