SymbOmni: Evolving Agentic Omni Models via Symbolic Concept Learning

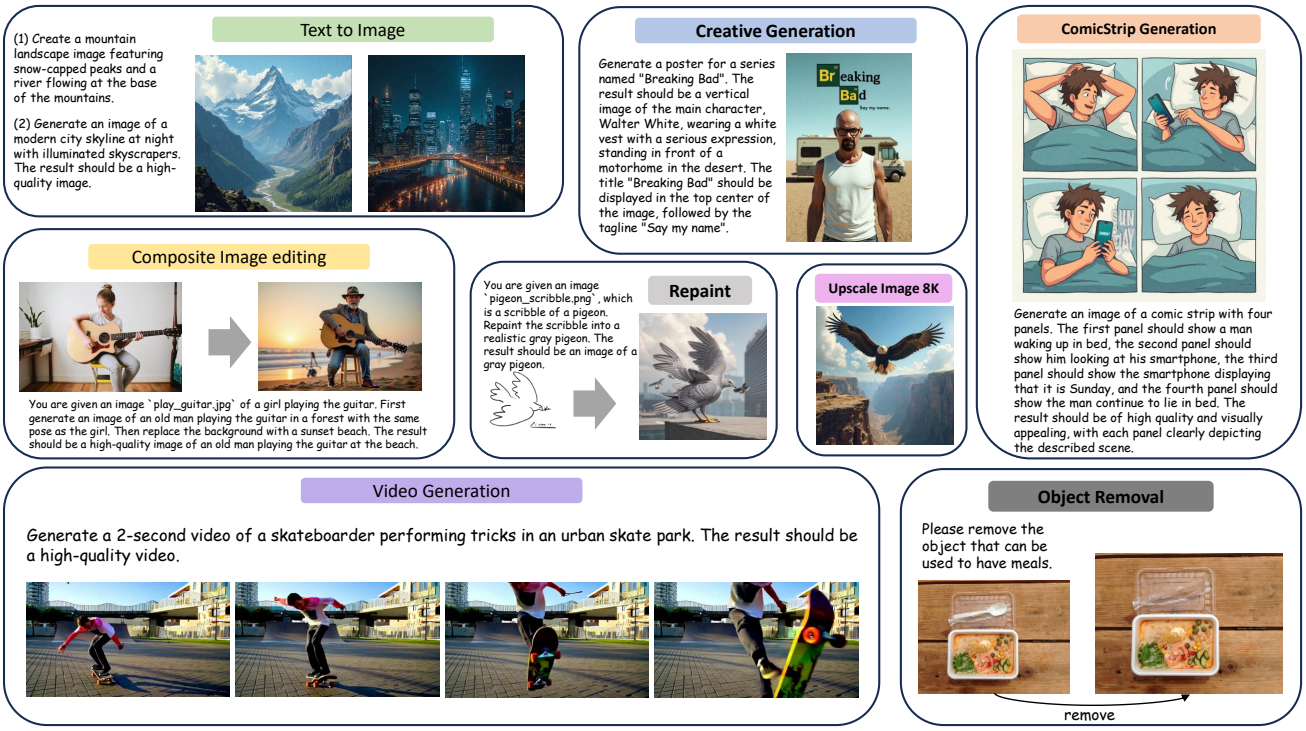


Figure 1. Overview of the *SymbOmni* framework. The model achieves versatile visual generation (text-to-image, creative content, image editing, video synthesis) through a unified approach for cumulative symbolic concept learning.

Abstract

Visual generation is rapidly expanding into diverse domains, from text-to-image/video synthesis to multimodal interactive creation. However, prevailing monolithic models are fundamentally limited by an inability for cumulative learning and self-evolution—the ***"perpetual novice" problem****. They lack mechanisms to structure experiences into reusable knowledge and consequently rely on brittle, "from-scratch" reasoning for each task, leading to poor compositional generalization and inefficient knowledge retention. To address this, we introduce* ***SymbOmni****, an agentic omni-model designed for cumulative evolution via* ***Symbolic Concept Learning****. Our architecture centers on the Symbolic Concept Box (CB)—an optimizable memory module that abstracts low-level operations into reusable Symbolic Workflow Instructions (SWIs). SymbOmni operates through an* ***induction-transduction cycle****: experiences are abstracted into symbolic concepts (induction), which are then strategically composed to solve novel tasks (transduction). This cycle is advanced by verbalized backpropagation with language-based optimization gradients to drive continuous self-improvement without gradient-based fine-tuning. Extensive evaluations demonstrate that SymbOmni achieves:* ***(I) Superior Performance****, significantly outperforming existing agent-based systems for iterative creation and leading closed-source models (e.g., Nano Banana, GPT-Image-1) in both image quality and task success rates;* ***(II) Enhanced Efficiency****, reducing token consumption by over 30% while maintaining competitive gen-*

eration quality; ***(III) Continuous Learning****, demonstrating genuine cumulative improvement in online learning scenarios on ComfyBench, establishing a new state-of-the-art.*

1. Introduction

Visual generation is rapidly expanding into diverse domains, ranging from text-to-image [21, 48, 73] and text-to-video [20, 54] to interactive generation [1, 15]. While single text-to-image/video models struggle with complex creative demands [71, 73], emerging task-specific or unified models [35, 53, 58] still face challenges in open-source settings, including unstable output quality and inadequate structured planning [25, 66]. Although closed-source models demonstrate impressive unified capabilities [9, 39, 41], scalability and broader multimodal applicability remain constrained [30, 62].

Concurrently, a more fundamental limitation emerges in current artificial intelligence paradigms: the inability of large language models (LLMs) and agent architectures to support cumulative learning and self-evolution [4, 19, 61]. Prevailing end-to-end architectures treat tasks in isolation, failing to distill persistent knowledge from experiences [47, 61]. This results in the ***"perpetual novice" problem***, characterized by poor compositional generalization, inefficient knowledge retention, and brittle decision-making [34, 55, 64].

The root cause lies deeper than technical implementation. The dominant unified generative-understanding model paradigm lacks mechanisms to structure experiences into reusable knowledge components. Consequently, these systems rely on ***"from-scratch" reasoning*** within fixed parametric memory for each new task [40, 41]. Alternative approaches using external tools often produce unstable, non-reusable reasoning chains [61, 63, 69]. Although existing research attempts to address this through implicit learning or tool-calling agents [29, 51], they suffer from poor scalability and decision instability. Even workflow-based approaches are hampered by rigid processes that cannot adapt to dynamic demands [13, 22, 44]. This limitation is particularly acute in enterprise scenarios with highly personalized workflows, where long-context methods incur prohibitive costs and hallucination risks [25, 27, 66].

In this paper, we introduce ***SymbOmni***, an agent architecture designed for cumulative evolution through Symbolic Concept Learning. We contend that enabling agents to continuously abstract experiences into structured, composable knowledge representations is critical to long-term adaptation. Therefore, SymbOmni incorporates a strong inductive bias that combines neural flexibility with symbolic rigor. Specifically, at the core of the architecture lies the Symbolic Concept Box (CB), an optimizable memory that stores reusable units of knowledge—called Symbolic Concepts—each of which encapsulates both semantic understanding and executable procedures representing a successful problem-solving strategy. Furthermore, these strategies are abstracted as Symbolic Workflow Instructions (SWIs), which are dynamically retrieved and compositionally executed via a memory-enhanced mechanism. This design enables an *Induction-Transduction cycle*: during induction, experiences are abstracted into symbolic concepts; conversely, during transduction, relevant concepts are retrieved and instantiated to address new tasks. Finally, the cycle is closed through verbalized backpropagation and provide linguistic feedback to guide self-improvement, thereby enabling continuous learning without parameter fine-tuning. Our contributions are threefold:

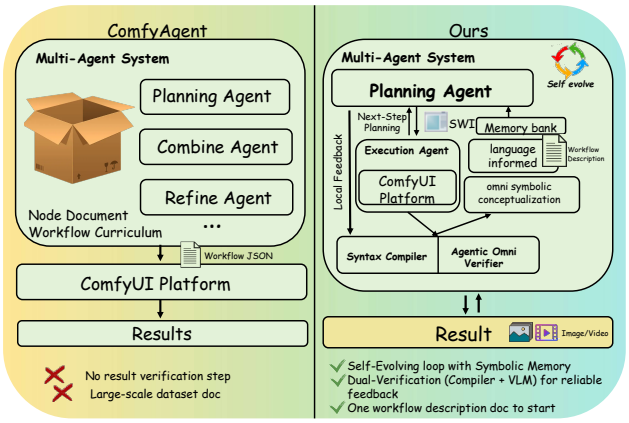


Figure 2. The comparison underscores the advantages of our model: a self-evolving capability through dual-verification feedback and an optimizable symbolic memory, which eliminates the dependency on large-scale pre-existing documentation required by the baseline approach.

- **Novel Architecture.** We propose ***SymbOmni***, an continually adaptable architecture centered on Symbolic Concept Learning, providing a path beyond monolithic, end-to-end models [22, 61, 63].
- **Formal System.** We develop a unified system combining inductive knowledge acquisition with transductive problem-solving, enabling genuine cumulative learning through verbalized backpropagation [40, 51, 74].
- **Empirical Validation.** Extensive experiments show **SymbOmni** achieves superior performance and efficiency, reducing token consumption by over 40% while demonstrating continuous improvement in enterprise workflows [25, 66, 75].

2. Related Work

Agent System and Workflow Automation Current AI systems often struggle with *cumulative learning*, treating tasks in isolation and suffering from the "perpetual novice"

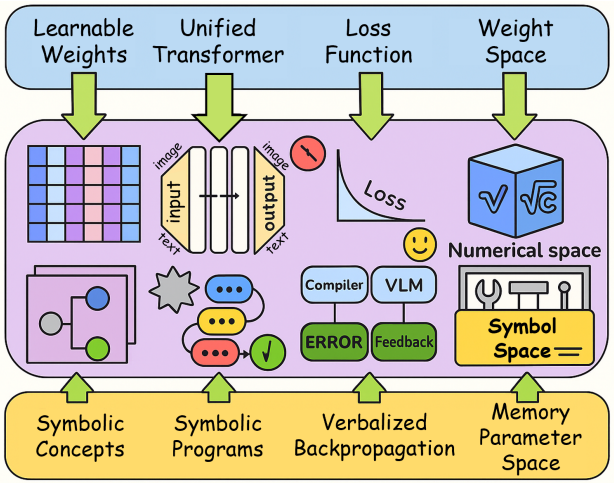


Figure 3. This figure contrasts multimodal unified models (top), which operate purely in weight space, with our agentic symbolic framework (bottom). While existing models rely on static parametric learning, our architecture introduces a symbolic concept layer that enables continuous knowledge abstraction and reuse through verbalized backpropagation—transforming rigid neural pipelines into dynamic, self-improving systems.

problem, which leads to poor compositional generalization and inefficient knowledge retention [34, 47, 55, 61]. Recent approaches aim to bridge this gap through structured knowledge representations and workflow automation. Systems like Deep Research [75] and OpenAI o3 [41] convert language into plans, while WorkflowLLM [13] and AFlow [72] focus on data-centric optimization. Early multi-agent systems [22, 44] enabled collaboration but were rigid. In specialized domains like ComfyUI, approaches such as ComfyAgent [66] and ComfyGPT [25] handle node-based workflows but face limited scope and error propagation. Collaborative frameworks like AutoGen [61] and ReAct [69] attempt iterative adjustments but struggle with robust error recovery. Our work advances this direction by formalizing an *Induction-Transduction* cycle and a dual-verifier system for genuine cumulative learning and adaptation.

Visual Generation and Reasoning Systems Visual generation has evolved from task-specific models [71, 73] to general-purpose systems [35, 53], with recent advances in diffusion [48] and unified models like GPT-Image-1 [39]. Concurrently, large reasoning models have improved planning via "System 2" thinking [38] and reinforcement learning [10]. However, these often rely on long, costly contexts prone to hallucination. **SymbOmni** integrates symbolic concept learning with visual workflow generation, creating reusable knowledge components that reduce computational overhead and enable continuous improvement, addressing the challenge of structured knowledge consolidation beyond current end-to-end approaches.

3. Methodology

3.1. System Overview

The Omni Agentic Model establishes a unified theoretical framework by integrating induction and transduction into a self-reinforcing cycle. In the transductive phase, the system applies generalized symbolic knowledge from the Symbolic Concept Box to solve novel tasks, adapting abstract concepts and constraints to specific problem instances without altering the core knowledge base. The inductive phase acts as the primary learning mechanism, where concrete experiences—both successful and unsuccessful—are abstracted into symbolic knowledge, extracting general rules and patterns from individual cases to continuously enrich the conceptual repository. This integration forms a coherent "Induction-Memory-Transduction-Experience" cycle as shown in Figure 4: induction populates symbolic memory with generalized knowledge, which in turn makes future transduction more effective. This reciprocal reinforcement enables continuous meta-learning, with each completed task enhancing the system's reasoning competence.

3.2. Symbolic Concepts and the Concept Box

The framework's foundation is the representation of knowledge as **Symbolic Concepts**. A Symbolic Concept, C_k, is a reusable module of expert knowledge defined by the quadruple:

$$C_k = (Desc_k, SWI_k, Params_k, Score_k) \quad (1)$$

where $Desc_k$ is a natural language description declaring the concept's intent; SWI_k is a Symbolic Workflow Instruction, serving as a parameterized template for a sequence of actions; $Params_k$ contains optimized parameter configurations learned from past applications; and $Score_k$ maintains a dynamic measure of the concept's historical effectiveness. During execution, parameter slots in SWI_k are instantiated with values from $Params_k$. The set of all concepts constitutes the **Concept Box (CB)**, the system's long-term, optimizable memory. The CB is formalized as a language system $\mathcal{L} = (\Sigma, \mathcal{R})$, where $\Sigma = \{Desc_k\}$ forms the vocabulary of semantic primitives and $\mathcal{R} = \{SWI_k\}$ constitutes the set of production rules defining valid workflows. Planning within this framework is thus equivalent to a grammatical derivation in $\mathcal{L}$, shifting problem-solving from first-principles construction to the assembly of proven building blocks. The natural language descriptions $Desc_k$ provide the semantic basis for measuring similarity between concepts and new tasks, enabling the hierarchical retrieval process detailed in Section 3.3.

3.3. The Self-Evolution Loop

Equipped with the structured Symbolic Concept Box (CB), SymbOmni operates through a continuous four-phase loop

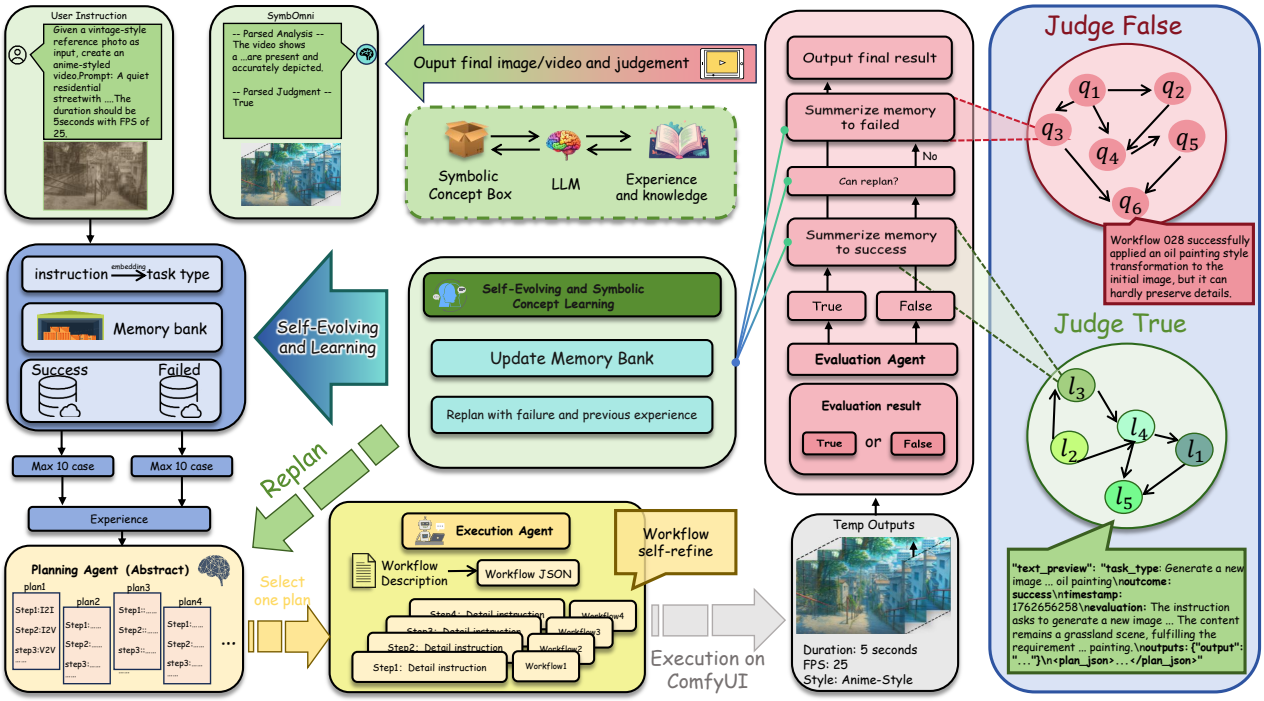


Figure 4. Illustration of the **self-evolving and symbolic concept learning** mechanism in **SymbOmni**. The framework transforms user instructions into solutions through an iterative process. Key to its evolution is the dual-feedback loop: successful outcomes are abstracted into symbolic experiences stored in the Memory Bank, while failures initiate a replay process that refines future planning. This creates a continuous learning cycle, allowing the agent to accumulate and leverage knowledge beyond a single task.

that seamlessly integrates planning, execution, and learning, as illustrated in Figure 4. This core cycle, centered around a dual-feedback mechanism, enables the system to learn cumulatively from experience. The following sections detail each phase.

Phase 1: Transduction (Planning) The loop begins with the Transduction phase, where the system translates a new task instruction into a concrete, executable plan by leveraging knowledge stored in the CB. Given task instruction I_{new}, hierarchical retrieval proceeds as follows. First, $ST = \text{Decompose}(I_{\text{new}})$ parses the instruction into semantic subtasks with dependencies. Then, concepts are retrieved through a two-level cascade:

$$C_{\text{ret}} = \text{Rank} \circ \text{Filter}_{\text{struct}} \circ \text{Retrieve}_{\text{sem}}(ST, \text{CB}) \quad (2)$$

where $\text{Retrieve}_{\text{sem}}$ matches concepts via $\cos(\text{Embed}(Desc_k), \text{Embed}(ST_m))$, $\text{Filter}_{\text{struct}}$ applies dependency constraints, and Rank orders by utility scores. The LLM planner generates workflows constrained by grammar $\mathcal{G}$:

$$WF_{\text{cand}} = \text{Instantiate}(\text{LLM}(I_{\text{new}} | C_{\text{ret}}, \mathcal{G})) \quad (3)$$

where Instantiate binds parameters Params_k to workflow templates SWI_k.

Phase 2: Execution The Execution phase is straightforward: the system executes the planned workflow WF_{cand} step-by-step, invoking the corresponding tools (e.g., image generators, filters) to produce the final output O. A complete execution trajectory τ is meticulously recorded, serving as the ground truth for the subsequent learning phase.

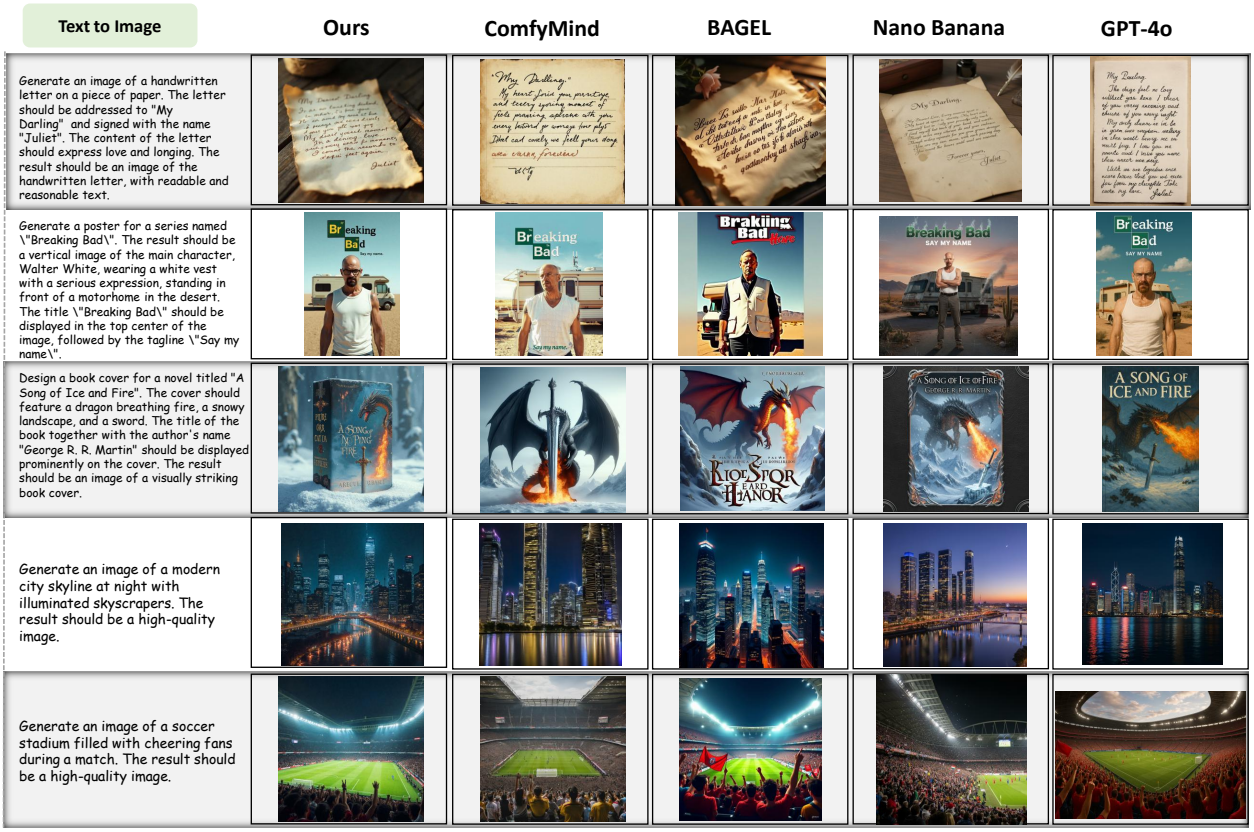
$$\tau = (I_{\text{new}}, WF_{\text{cand}}, O, C_{\text{ret}}, \text{ExecutionLog}) \quad (4)$$

Phase 3: Induction (Learning) & Evaluation The Induction phase is the core of the system's learning. It begins with a critical Evaluation step by the Evaluation Agent, which assesses the output O against the instruction I_{new}, producing a binary judgment $\text{Judge} \in \{\text{True}, \text{False}\}$. This judgment determines the subsequent learning path, realizing the dual-feedback loop:

If $\text{Judge} = \text{True}$ (successful execution), the trajectory τ is abstracted into positive symbolic experiences. The system performs **symbolic abstraction** to extract generalizable knowledge, potentially creating new composite concepts or reinforcing existing ones.

$$\text{CB} \leftarrow \text{CB} \cup \text{Symbolize}(\tau, \text{Positive}) \quad (5)$$

If $\text{Judge} = \text{False}$ (failed execution), it triggers a detailed diagnostic process. The system analyzes the failure through



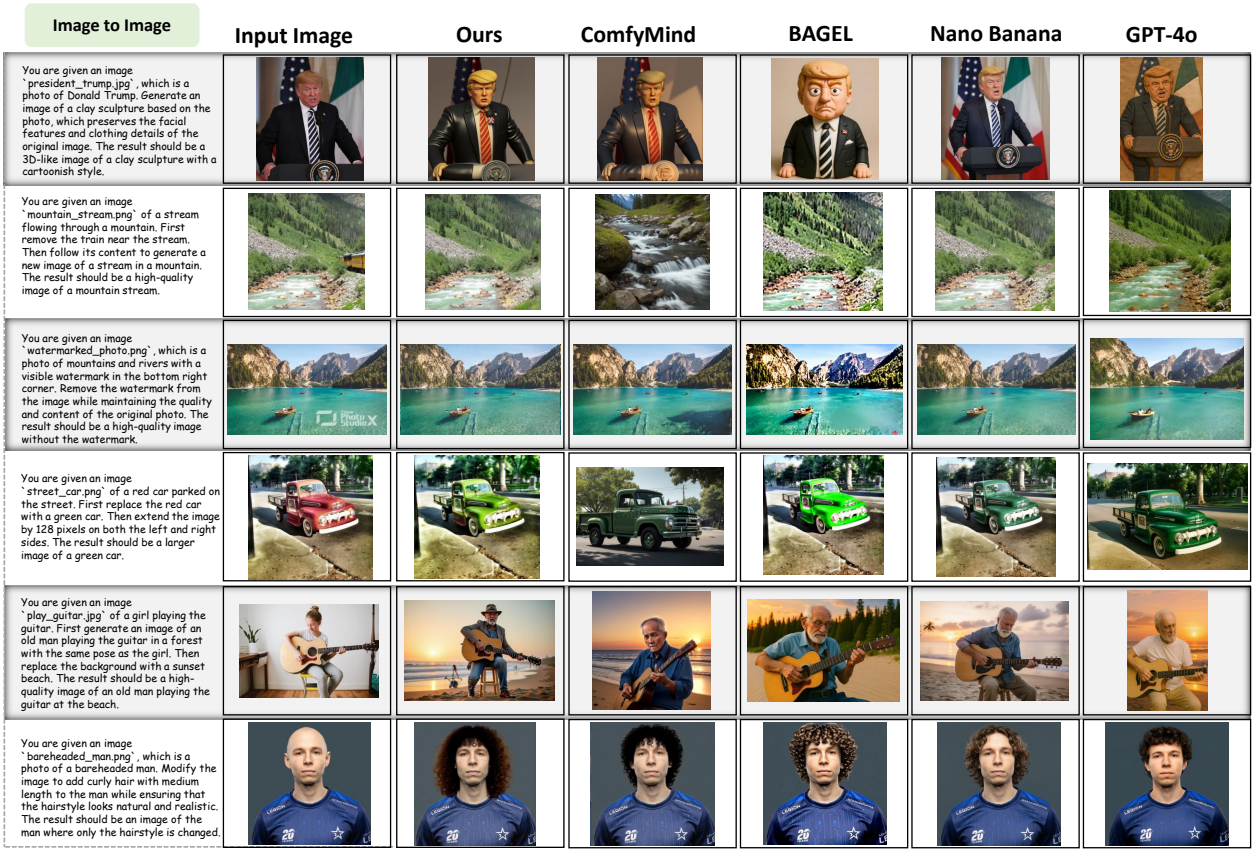


Figure 5. Qualitative comparison results for text-to-image generation and image-to-image tasks.

verbalized backpropagation: dual verifiers assess execution quality ($\text{Error}_{\text{hard}}$ for syntactic correctness, $\text{Feedback}_{\text{soft}}$ for semantic quality), which are synthesized by an LLM into a structured *linguistic loss* L_{lang} that diagnoses the failure. Chain attribution then computes *symbolic gradients* ∇_{sym} to pinpoint refinement targets in the CB, guiding a **replay and refinement** process.

$$L_{\text{lang}} = \text{LLM}(\mathcal{P}_{\text{loss}}(\tau, \text{Error}_{\text{hard}}, \text{Feedback}_{\text{soft}})) \quad (6)$$

$$\text{CB} \leftarrow \text{Refine}(\text{CB}, \text{Replay}(\tau, \nabla_{\text{sym}})) \quad (7)$$

Phase 4: Memory Optimization In the **Memory Optimization** phase, the insights from Phase 3 are consolidated. The CB is updated with the newly formed or refined symbolic concepts. Successful outcomes are abstracted into reusable knowledge (C_{success}), while failures contribute to avoidance patterns (C_{negative}) or parameter refinements (C_{refine}). This update, guided by the principles of symbolic learning, completes one self-evolution loop. The enriched CB equips the system to perform better on similar tasks in the future, achieving continuous, cumulative learning.

4. Experiments and Results

We evaluated our system's performance on three complementary public benchmarks: ComfyBench for assessing AI agent workflows and automatic workflow generation from natural language descriptions [66], GenEval for assessing semantic consistency and visual fidelity in text-to-image generation tasks [17], and ReasonEdit for examining performance in multi-step reasoning and complex image editing scenarios [26]. Additionally, we analyzed the token efficiency and runtime of SymbOmni on large-scale tasks, and examined the performance differences between Online and Offline modes to demonstrate self-evolution capabilities and token efficiency [25, 31, 51, 55, 75].

4.1. Experimental Setup

To rigorously test planning and learning capabilities, we systematically updated the workflow library by enhancing workflow coordination mechanisms, incorporating high-quality workflows while intentionally retaining some sub-optimal ones, and upgrading selected models to their latest versions. All experiments were conducted on the **ComfyUI** framework [7, 8], using Gemini 2.5 Flash [6] as the reasoning engine. To ensure a fair comparison, we unified key hyperparameters across all experiments: a maximum search depth of 10 and a maximum of 4 retries [28, 42, 49, 69, 70].

Our evaluation metrics encompass several key dimensions: **Pass Rate** measures whether the generated workflow executes successfully with the correct structure; **Resolve Rate** indicates whether the final output conforms to the instruction semantics; **Generation Quality** assesses the semantic consistency and subjective quality of the results; and **Token Consumption** measures the average number of tokens required per task, serving as an indicator of reasoning efficiency [33, 37, 45].

4.2. Qualitative Comparison

We conducted a qualitative comparison of SymbOmni against state-of-the-art collaborative AI systems [18] and leading closed-source unified models [6, 11, 39]. As shown in Figure 5, our method demonstrates distinct advantages in handling complex compositional tasks: **(1) Text-to-Image Generation with Complex Constraints** In tasks requiring precise adherence to complex instructions, SymbOmni shows superior semantic alignment. For example, when generating the "Breaking Bad" poster, our method correctly renders Walter White in a white vest with a serious expression within a room setting, and accurately places the title and tagline. In contrast, several baselines struggle with character depiction, background details, or text integration. **(2) Complex Image Editing and Image-to-Image Translation SymbOmni** excels in multi-step image editing by reliably executing sequences of operations. In the "red car to green car" task, our method successfully changes the car's color and extends the canvas as instructed, whereas other approaches may only partially complete the request. For the "clay sculpture" transformation of Donald Trump, SymbOmni better preserves facial features while applying the designated cartoonish, 3D-like clay texture, outperforming baselines that often distort the likeness or fail to achieve the desired stylistic effect. These qualitative results demonstrate that SymbOmni's symbolic concept learning enables more accurate interpretation and execution of complex, multi-faceted instructions, leading to superior outcomes in both generation and editing tasks compared to existing alternatives [1, 15, 71, 73].

4.3. Quanitative experiment

ComfyBench: Autonomous Workflow Construction Performance ComfyBench evaluates the ability to construct executable workflows from task descriptions [67]. As shown in Table 1, SymbOmni achieves a perfect 100% pass rate and a superior total resolve rate of 91.7%, significantly outperforming all baseline methods. The advantage is most substantial on complex and creative tasks: for Complex tasks, the resolve rate improves from 75.0% to 91.7% (22.2% relative gain), and for the challenging Creative tasks, it surges from 42.5% to 67.5%, a remarkable 58.8% relative improvement. This demonstrates that by retrieving and composing high-success-rate symbolic concepts from memory, SymbOmni effectively minimizes inefficient exploration during planning, enabling more robust and effective problem-solving [13, 36, 72].

Reason-Edit: Complex Visual Reasoning Editing ReasonEdit evaluates the system's multi-step execution

Agent	Vanilla		Complex		Creative		Total	
	%Pass↑	%Resolve↑	%Pass↑	%Resolve↑	%Pass↑	%Resolve↑	%Pass↑	%Resolve↑
GPT-4o + Few-shot [2]	32.0	27.0	16.7	8.3	7.5	0.0	22.5	16.0
GPT-4o + CoT [59]	44.0↑12.0	29.0↑2.0	11.7↓5.0	8.3↑0.0	12.5↑5.0	0.0↑0.0	28.0↑5.5	17.0↑1.0
GPT-4o + CoT-SC [57]	45.0↑13.0	34.0↑7.0	11.7↓5.0	5.0↓3.3	15.0↑7.5	0.0↑0.0	29.0↑6.5	18.5↑2.5
Claude-3.5-Sonnet + RAG [27]	27.0↓5.0	13.0↓14.0	23.0↑6.3	6.7↓1.6	7.5↑0.0	0.0↑0.0	22.0↓0.5	8.5↓7.5
Llama-3.1-70B + RAG	58.0↑26.0	32.0↑5.0	23.0↑6.3	10.0↑1.7	15.0↑7.5	5.0↑5.0	39.0↑16.5	20.0↑4.0
GPT-4o + RAG	62.0↑30.0	41.0↑14.0	45.0↑28.3	21.7↑13.4	40.0↑32.5	7.5↑7.5	52.0↑29.5	23.0↑7.0
o1-mini + RAG	32.0↑0.0	16.0↓11.0	21.7↑5.0	8.3↑0.0	12.5↑5.0	7.5↑7.5	25.0↑2.5	12.0↓4.0
o1-preview + RAG	70.0↑38.0	46.0↑19.0	48.3↑31.6	23.3↑15.0	30.0↑22.5	12.5↑12.5	55.5↑33.0	32.5↑16.5
ComfyAgent [67]	67.0	46.0	48.3	21.7	40.0	15.0	56.0	32.5
ComfyMind [18] (reproduced)	100.0↑33.0	84.0↑38.0	100.0↑51.7	75.0↑53.3	100.0↑60.0	42.5↑27.5	100.0↑44.0	73.0↑40.5
SymbOmni(Ours)	100.0↑33.0	95.0↑49.0	100.0↑51.7	91.7↑70.0	100.0↑60.0	67.5↑52.5	100.0↑44.0	88.5↑56.0

Table 1. Evaluation of autonomous workflow construction on ComfyBench [67]. We highlight the best results with colored boxes.

Method	Overall↑	Single Obj.↑	Two Obj.↑	Counting↑	Colors↑	Position↑	Attr. Binding↑
Frozen Text Encoder Mapping Methods							
SDv1.5 [48]	0.43	0.97	0.38	0.35	0.76	0.04	0.06
SDv2.1 [48]	0.50↑0.07	0.98↑0.01	0.51↑0.13	0.44↑0.09	0.85↑0.09	0.07↑0.03	0.17↑0.11
SD-XL [43]	0.55↑0.12	0.98↑0.01	0.74↑0.36	0.39↑0.04	0.85↑0.09	0.15↑0.11	0.23↑0.17
DALLE-2 [46]	0.52↑0.09	0.94↓0.03	0.66↑0.28	0.49↑0.14	0.77↑0.01	0.10↑0.06	0.19↑0.13
SD3-Medium [12]	0.74↑0.31	0.99↑0.02	0.94↑0.56	0.72↑0.37	0.89↑0.13	0.33↑0.29	0.60↑0.54
Multimodal Unified Models							
LlamaGen [52]	0.32	0.71	0.34	0.21	0.58	0.07	0.04
LWM [32]	0.47↑0.15	0.93↑0.22	0.41↑0.07	0.46↑0.25	0.79↑0.21	0.09↑0.02	0.15↑0.11
SEED-X [16]	0.49↑0.17	0.97↑0.26	0.58↑0.24	0.26↑0.05	0.80↑0.22	0.19↑0.12	0.14↑0.10
Emu3-Gen [58]	0.54↑0.22	0.98↑0.27	0.71↑0.37	0.34↑0.13	0.81↑0.23	0.17↑0.10	0.21↑0.17
Janus [60]	0.61↑0.29	0.97↑0.26	0.68↑0.34	0.30↑0.09	0.84↑0.26	0.46↑0.39	0.42↑0.38
JanusFlow [35]	0.63↑0.31	0.97↑0.26	0.59↑0.25	0.45↑0.24	0.83↑0.25	0.53↑0.46	0.42↑0.38
Janus-Pro-7B [5]	0.80↑0.48	0.99↑0.28	0.89↑0.55	0.59↑0.38	0.90↑0.32	0.79↑0.72	0.66↑0.62
GoT [14]	0.64↑0.32	0.99↑0.28	0.69↑0.35	0.67↑0.46	0.85↑0.27	0.34↑0.27	0.27↑0.23
Bagel [11]	0.78↑0.46	0.98↑0.27	0.94↑0.60	0.76↑0.55	0.91↑0.33	0.69↑0.62	0.70↑0.66
GPT-Image-1 [39]	0.84↑0.52	0.99↑0.28	0.92↑0.58	0.85↑0.64	0.92↑0.34	0.75↑0.68	0.61↑0.57
Collaborative AI Systems							
ComfyAgent [66]	0.32	0.69	0.30	0.33	0.50	0.04	0.04
ComfyMind [18] (reproduced)	0.90↑0.58	1.00↑0.31	1.00↑0.70	0.96↑0.63	0.97↑0.47	0.63↑0.59	0.81↑0.77
SymbOmni(Ours)	0.98↑0.66	1.00↑0.31	1.00↑0.70	0.99↑0.66	0.98↑0.48	0.97↑0.93	0.95↑0.91

Table 2. Evaluation of T2I generation on GenEval [17]. Obj.: Object. Attr.: Attribution. We highlight the best results with colored boxes.

and self-correction capabilities within visual reasoning chains [26]. Since SymbOmni retrieves relevant symbolic concepts before task planning, it can more rapidly select optimal editing paths and reduce failed execution attempts. Figure 6 shows that SymbOmni achieves an overall score of 9.190, substantially outperforming ComfyMind (8.135) and other baseline methods. The advantage is most pronounced in complex spatial reasoning: Left-Right tasks show a +2.321-point improvement, while Multiple-Objects attribute binding gains +0.967 points. These results demonstrate that symbolic concept retrieval enables more efficient reasoning chain construction compared to methods relying solely on step-by-step reasoning or predefined correction strategies [34, 56, 64].lex multi-step reasoning scenarios.

GenEval: Text-to-Image Generation We further evaluate the system's capability on the GenEval for compositional text-to-image generation. As shown in Table 2, **SymbOmni** achieves a state-of-the-art overall score of 0.98, outperforming all existing methods across three major categories. Compared to frozen text encoder mapping methods (best: SD3-Medium at 0.74) and LLMs/MLLMs enhanced methods (best: GPT-Image-1 at 0.84), **SymbOmni** demonstrates superior performance in collaborative AI systems, showing an 8.9% improvement over the strongest baseline ComfyMind (0.90). The advantage is particularly significant in complex compositional tasks: **SymbOmni** achieves near-perfect scores of 0.97 in **Position** (vs. 0.33-0.79 in other methods) and 0.95 in **Attribute Binding** (vs. 0.06-0.81), demonstrating its exceptional capability in handling complex spatial relationships and multi-attribute binding. These results highlight how symbolic concept learning and self-evolving mechanism enables more robust compositional reasoning compared to text-to-image models, unified models and Collaborative AI Systems.

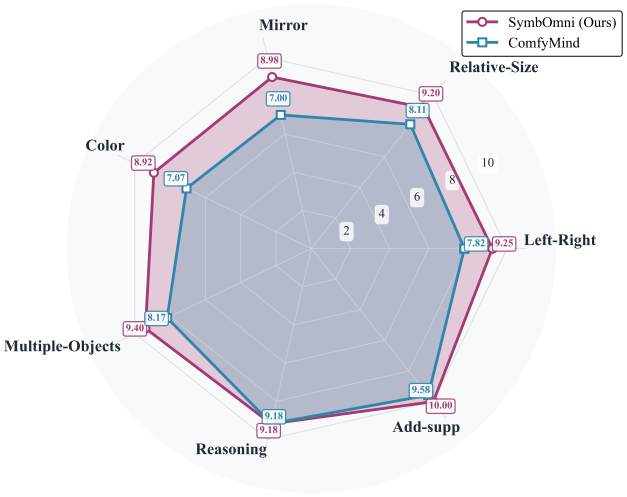


Figure 6. Evaluation of reasoning image-to-image editing on ReasonEdit [26].

Method	Avg Tokens/Case			Avg/Case
	Input↓	Output↓	Total↓	Requests↓
Ours (w/o EXP)	18,556.1	3,171.0	21,727.1	9.85
Ours	12,463.4 ↓32.8%	1,898.0 ↓40.1%	14,361.4 ↓33.9%	7.25 ↓26.4%

Table 3. Performance and cost statistics on ReasonEdit benchmark. We highlight the best results with colored boxes.

4.4. Ablation Study

To validate the critical role of the symbolic concept memory, we conduct an ablation study by removing the memory retrieval module (w/o EXP). Results in Table 3 show that disabling memory leads to substantial performance degradation across all metrics, confirming its necessity [2, 27, 59]. The symbolic concept memory drastically enhances computational efficiency. The input token count is reduced by 32.8%, while the output token count sees an even greater reduction of 40.1%, culminating in a 33.9% decrease in total token consumption. This demonstrates that by retrieving relevant historical concepts, the system bypasses redundant context processing and lengthy reasoning chains [9, 10, 38].

Furthermore, the memory module reduces the average number of API requests per case by 26.4% , indicating more decisive and efficient task planning with fewer iterative refinements. These results unequivocally show that the symbolic concept memory is not merely an additive component but a fundamental mechanism for experience reuse. It enables the accumulation of task-level knowledge structures, a capability absent in methods reliant solely on in-context learning or parametric recall, thus providing a clear path for sustainable long-term operation [3, 24, 50].

4.5. Offline Mode Performance Evaluation

To assess the viability of experience-driven planning in resource-limited environments, we designed an offline evaluation where the agent operates solely on retrieved symbolic concepts without access to external workflow descriptions. The experiment comprises two phases: (1) an online phase where the agent completes 100 ComfyBench tasks to build its experience memory; (2) an offline phase where it tackles another 100 tasks using only accumulated symbolic knowledge [4, 23, 44]. We compare our offline approach against ComfyMind, which has full access to workflow descriptions but lacks experience memory. This comparison directly tests whether learned symbolic concepts can effectively substitute for explicit documentation. We focus our analysis on the challenging **Complex** task category.

Method	Resolved↑	Avg Tries↓	Avg Tries (Resolved)↓
ComfyMind (w/ description)	66.7%	2.000	1.600
Ours (Offline)	70.0% ↑3.3%	1.433 ↓0.567	1.190 ↓0.410

Table 4. Offline mode performance on **Complex** tasks.

As shown in Table 5, our offline approach achieves a 70.0% resolve rate, outperforming the documentation-dependent baseline. More significantly, it reduces the average attempt count by 28.4% and by 25.6% for successfully resolved tasks. These results demonstrate that symbolic concept memory not only compensates for the absence of explicit documentation but enables more efficient problem-solving through experience-based optimization, highlighting its practical value in deployment scenarios with limited external resources [47, 61, 63].

5. Concluding Remarks

We introduces ***SymbOmni***, a cognitive architecture for multimodal creation that addresses the perpetual novice problem through Symbolic Concept Learning. By abstracting experiences into reusable Symbolic Concepts and enabling an Induction-Transduction cycle with verbalized backpropagation, our approach supports genuine cumulative learning without parameter updates. Experiments demonstrate superior efficiency and continuous improvement, reducing token consumption by over 40% while enabling self-evolution in complex workflows. The dual-feedback mechanism establishes a robust foundation for building more autonomous and adaptive AI systems that learn from both successes and failures. Our work bridges symbolic reasoning with experiential learning, offering a path towards sustainable AI evolution beyond one-shot task completion. Future work will explore scaling the symbolic concept library across broader domains and enhancing the granularity of concept to support more sophisticated reasoning capabilities and diverse multimodal creation tasks.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vi-*

sion and pattern recognition, pages 18392–18402, 2023. 2, 6

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 7, 8

[3] 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. Why do multi-agent llm systems fail?, 2025. 8

[4] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 8

[5] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025. 7

[6] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. 2025. 6, 1, 2, 7

[7] comfyanonymous. Comfyui. https://github.com/comfyanonymous/ComfyUI, 2023. 6, 1

[8] ComfyUI Contributors. ComfyUI: A powerful and modular stable-diffusion gui, 2023. Accessed: 2025-05-14. 6

[9] Joost C. F. de Winter, Dimitra Dodou, and Yke Bauke Eisma. System 2 thinking in openai's o1-preview model: Near-perfect performance on a mathematics exam. *CoRR*, abs/2410.07114, 2024. 2, 8

[10] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. 3, 8, 7

[11] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. 6, 7

[12] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, and Frederic Boesel. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 7

[13] Shengda Fan, Xin Cong, Yuepeng Fu, Zhong Zhang, Shuyan Zhang, Yuanwei Liu, Yesai Wu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. WorkflowLLM: Enhancing workflow orchestration capability of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 3, 6

[14] Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Shilin Yan, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, et al. Got: Unleashing reasoning capability of multimodal large language model for visual generation and editing. *arXiv preprint arXiv:2503.10639*, 2025. 7

[15] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. *arXiv preprint arXiv:2309.17102*, 2023. 2, 6

[16] Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024. 7

[17] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023. 6, 7

[18] Litao Guo, Xinli Xu, Luozhou Wang, Jiantao Lin, Jinsong Zhou, Zixin Zhang, Bolan Su, and Ying-Cong Chen. Comfymind: Toward general-purpose generation via tree-based planning and reactive feedback, 2025. 6, 7, 1, 2

[19] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges, 2024. 2

[20] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion, 2024. arXiv preprint arXiv:2501.00103. 2

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851. Curran Associates, Inc., 2020. 2

[22] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 3(4):6, 2023. 2, 3

[23] 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. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024. 8

[24] Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *The Thirteenth International Conference on Learning Representations*, 2025. 8

[25] Oucheng Huang, Yuhang Ma, Zeng Zhao, Mingrui Wu, Jiayi Ji, Rongsheng Zhang, Zhipeng Hu, Xiaoshuai Sun, and Rongrong Ji. Comfygpt: A self-optimizing multi-agent system for comprehensive comfyui workflow generation, 2025. 2, 3, 6

[26] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 8362–8371, 2024. 6, 7, 8, 2

[27] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020. 2, 7, 8

[28] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*, 2025. 6

[29] Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl, 2025. 2

[30] 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. Perception, reason, think, and plan: A survey on large multimodal reasoning models, 2025. 2

[31] Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, et al. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*, 2025. 6

[32] Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with ringattention. *arXiv e-prints*, pages arXiv–2402, 2024. 7

[33] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023. 6

[34] Yexiang Liu, Jie Cao, Zekun Li, Ran He, and Tieniu Tan. Breaking mental set to improve reasoning through diverse multi-agent debate. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 3, 7

[35] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Liang Zhao, et al. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. *arXiv preprint arXiv:2411.07975*, 2024. 2, 3, 7

[36] Boye Niu, Yiliao Song, Kai Lian, Yifan Shen, Yu Yao, Kun Zhang, and Tongliang Liu. Flow: Modularized agentic workflow automation. In *The Thirteenth International Conference on Learning Representations*, 2025. 6

[37] Yuwei Niu, Munan Ning, Mengren Zheng, Bin Lin, Peng Jin, Jiaqi Liao, Kunpeng Ning, Bin Zhu, and Li Yuan. WISE: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025. 6

[38] OpenAI. Introducing openai o1-preview. https://openai.com/index/introducing-openai-o1-preview/, 2024. 3, 8

[39] OpenAI. gpt-image-1, 2025. 2, 3, 6, 7, 1

[40] OpenAI. Introducing deep research, 2025. Accessed: 2025-02-02. 2

[41] OpenAI. Introducing openai o3 and o4-mini, 2025. Accessed: 2025-04-18. 2, 3

[42] Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565, 2024. 6

[43] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 7

[44] 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. Communicative agents for software development, 2023. 2, 3, 8

[45] Shuofei Qiao, Runnan Fang, Zhisong Qiu, Xiaobin Wang, Ningyu Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Benchmarking agentic workflow generation, 2025. 6

[46] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 7

[47] Toran Bruce Richards. AutoGPT, 2023. 2, 3, 8

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 3, 7

[49] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023. 6

[50] Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. Agentsquare: Automatic LLM agent search in modular design space. In *The Thirteenth International Conference on Learning Representations*, 2025. 8

[51] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2, 6, 8

[52] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 7

[53] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 2, 3

[54] Tencent. Hunyuanvideo, 2024. 2

[55] Junlin Wang, Jue WANG, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language

model capabilities. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 3, 6

[56] Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 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, 2024. Association for Computational Linguistics. 7

[57] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022. 7

[58] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 2, 7

[59] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 7, 8

[60] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint*, 2024. 7

[61] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023. 2, 3, 8

[62] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. VILA-U: A unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024. 2

[63] Tianbao Xie, Fan Zhou, Zhoujun Cheng, Peng Shi, Luoxuan Weng, Yitao Liu, Toh Jing Hua, Junning Zhao, Qian Liu, Che Liu, et al. Openagents: An open platform for language agents in the wild, 2023. 2, 8

[64] Zhenran Xu, Senbao Shi, Baotian Hu, Jindi Yu, Dongfang Li, Min Zhang, and Yuxiang Wu. Towards reasoning in large language models via multi-agent peer review collaboration, 2023. 2, 7

[65] Zhenran Xu, Yangxue Yangxue, Yiyu Wang, Qingli Hu, Zijiao Wu, Baotian Hu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 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, 2025. 1

[66] Xiangyuan Xue, Zeyu Lu, Di Huang, Zidong Wang, Wanli Ouyang, and Lei Bai. Comfybench: Benchmarking llm-based agents in comfyui for autonomously designing collaborative ai systems, 2024. 2, 3, 6, 7

[67] Xiangyuan Xue, Zeyu Lu, Di Huang, Zidong Wang, Wanli Ouyang, and Lei Bai. Comfybench: Benchmarking llm-based agents in comfyui for autonomously designing collaborative ai systems, 2024. 6, 7

[68] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 7

[69] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 3, 6

[70] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. 6

[71] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 2, 3, 6

[72] 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. AFlow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations*, 2025. 3, 6

[73] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 2, 3, 6

[74] Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. Marco-o1: Towards open reasoning models for open-ended solutions. *CoRR*, abs/2411.14405, 2024. 2

[75] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025. 2, 3, 6

SymbOmni: Evolving Agentic Omni Models via Symbolic Concept Learning

Supplementary Material

The supplementary material includes the following additional information:

- **Sec. A** introduces the ComfyUI-SymbOmni plugin for practical deployment and community adoption.
- **Sec. B** presents the complete qualitative comparison across 38 test cases covering text-to-image synthesis, complex compositional generation, creative content creation, and image editing tasks.
- **Sec. C** details the user study design and results, including evaluation methodology and preference analysis across 1,197 responses.
- **Sec. D** provides additional quantitative experiments, including comprehensive evaluation on the ReasonEdit benchmark, offline mode performance assessment, and LLM ablation study.
- **Sec. E** discusses the self-correction mechanism in symbolic concept learning and how the system addresses memory pollution through continuous optimization.
- **Sec. F** contains the complete prompt set used in the SymbOmni system.

A. ComfyUI-SymbOmni Plugin

To facilitate practical deployment and community adoption, we release a ComfyUI-SymbOmni plugin based on ComfyUI-Copilot [65] within the ComfyUI ecosystem [7]. We remove the closed-source backend dependencies in the original Copilot; all planning, step execution, workflow optimization, and concept learning management are performed locally with open-source components, powered by SymbOmni. The plugin integrates symbolic concept learning capabilities into the ComfyUI GUI, enabling users to interactively manage symbolic concepts, personal workflows, workflow experiences and optimize workflows through an intuitive node-based interface.

B. Complete Qualitative Comparison

This section presents comprehensive visual generation results across 38 carefully designed test cases, evaluated using five different methods. These cases systematically cover major application scenarios in visual generation, including simple text-to-image synthesis, complex compositional generation, creative content creation, and image editing tasks. The results provide a foundation for subsequent user preference studies.

B.1. Test Case Design

We constructed 38 representative cases that systematically span the primary challenges and application scenarios in visual generation. The cases are organized into four major categories: simple text-to-image tasks (8 cases) covering basic object-scene combinations such as "Generate an image of a cat sitting on a windowsill looking outside" and "Generate an image of a futuristic factory filled with robots assembling machines"; complex compositional generation (6 cases) featuring multi-object spatial arrangements and attribute binding challenges, exemplified by "Breaking Bad movie poster with Walter White in white vest" and geometric configurations with precise color-position specifications; Creative content generation (4 cases) explores diverse artistic creations, such as a four-panel comic strip depicting a man discovering it's Sunday and staying in bed, and designing a visually striking book cover for "A Song of Ice and Fire" that incorporates a dragon, a snowy landscape, and a sword; and image editing tasks (20 cases) encompassing color transformation(1 case), style transfer(3 cases), object removal(4 cases), object insertion(3 cases), image matting(4 cases), reasoning editing(3 cases),and object manipulation(2 cases) such as converting "red car to green car with canvas extension" and "Donald Trump photo to clay sculpture style."

B.2. Comparison Methods

For each case, we generated results using five distinct approaches. SymbOmni, our proposed method, employs Gemini 2.5 Flash [6] as the planning engine with symbolic concept retrieval and self-evolution capabilities enabled. ComfyMind [18] serves as the baseline agent system, utilizing the same reasoning engine but relying on heuristic search without symbolic concept learning. BAGEL represents unified multimodal models with end-to-end generation from text to image and from image to iamge. Nano Banana [6] and GPT-Image-1 [39] provide state-of-art closed-source unified model performance for comparison.

C. User Study

Based on the generation results from the 38 cases presented in Section B, we conducted a large-scale user preference study to systematically evaluate the quality and effectiveness of different approaches.

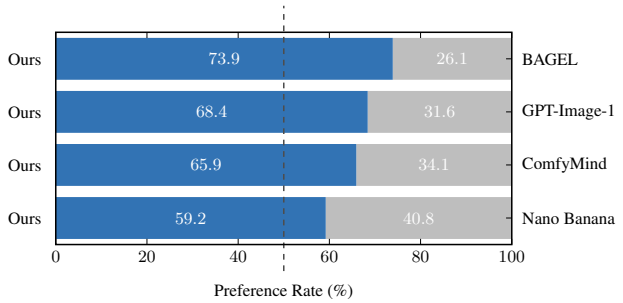


Figure 7. Pairwise preference rates of SymbOmni vs. other methods. The dashed line indicates the 50% threshold.

C.1. Study Design

We adopted a pairwise blind comparison methodology. From the five methods' results across 38 cases, SymbOmni was compared one-on-one against each of the four alternative methods, yielding a total of 152 comparison questions (38 cases $\times$ 4 comparisons).

We recruited 63 volunteers to participate in the evaluation. To ensure comprehensive coverage while maintaining evaluation quality, each volunteer was assigned to evaluate 19 image pairs randomly selected from the 152 comparison questions. The blind testing protocol ensured complete randomization of image display order, with no method identification visible to participants who only saw labels such as "Image A" and "Image B". Text prompts and input image were clearly displayed above each image pair, and participants remained unaware of any method information until completing all evaluations.

C.2. Evaluation Dimensions and Results

Through this evaluation process, we collected a total of 1,197 preference responses (63 volunteers $\times$ 19 pairs). Participants evaluated each image pair across four comprehensive dimensions. Semantic consistency measured how well the generated result matched the text prompt's intent. Visual quality assessed overall image quality including clarity, coherence, and realism. Compositional accuracy evaluated the completion of multi-object and multi-attribute tasks, applicable specifically to complex compositional cases. Overall preference represented participants overall judgment considering all factors.

The user study results demonstrate strong preference for SymbOmni across all comparison pairs. Against Nano Banana, SymbOmni achieved a 59.2% preference rate, indicating clear advantages in handling complex visual generation tasks. The preference gap widened against ComfyMind (65.9%) and GPT-Image-1 (68.4%), reflecting the effectiveness of symbolic concept learning over heuristic search and end-to-end approaches. The most substantial advantage emerged against BAGEL (73.9%), suggesting that agentic workflow decomposition combined with symbolic memory significantly outperforms unified multimodal architectures in compositional reasoning and creative generation scenarios.

D. Additional Quantitative Experiments

D.1. Comprehensive Evaluation on ReasonEdit Benchmark

To provide a more equitable comparison between SymbOmni and existing unified multimodal models, we conducted comprehensive testing on the ReasonEdit benchmark [26] across seven reasoning-intensive subcategories. Figure 12 presents the detailed performance comparison among ComfyMind [18], Nano Banana [6], and SymbOmni.

The ReasonEdit evaluation reveals several important insights into the capabilities of different approaches. SymbOmni demonstrates particularly strong performance in reasoning-intensive tasks, achieving 9.183 points in the Reasoning category compared to Nano Banana's 9.158 points, and significantly outperforming ComfyMind's 9.175 points. More notably, in the Mirror task which requires sophisticated spatial reasoning capabilities, SymbOmni achieves 8.983 points, substantially exceeding Nano Banana's 8.383 points by 7.2% and ComfyMind's 7.000 points by 28.3%. This demonstrates that symbolic workflow decomposition and concept-driven planning provide distinct advantages in handling complex reasoning chains that demand multi-step spatial transformations.

In other task categories, SymbOmni maintains competitive performance with Nano Banana. While Nano Banana exhibits slightly higher scores in certain subcategories such as Left-Right (9.804 vs. 9.250) and Color (9.778 vs. 8.923), these differences primarily reflect the inherent capabilities of the underlying generation models rather than limitations of the agentic framework. As an Agentic Omni Model, SymbOmni's performance is fundamentally bounded by the quality of its constituent tool models—in scenarios where tasks demand exceptional single-model generation quality (e.g., precise color matching or spatial positioning), the framework's effectiveness is constrained by the capabilities of the invoked generation tools. Nonetheless, SymbOmni achieves perfect scores (10.000) in the Add-supp category alongside both competitors, and maintains strong performance in Multiple-Objects (9.396) and Relative-Size (9.204) tasks, confirming that symbolic concept learning effectively preserves task completion quality while enabling superior reasoning capabilities.

D.2. Offline Mode Performance Evaluation Details

To assess the viability of experience-driven planning in resource-limited environments, we designed an offline eval-

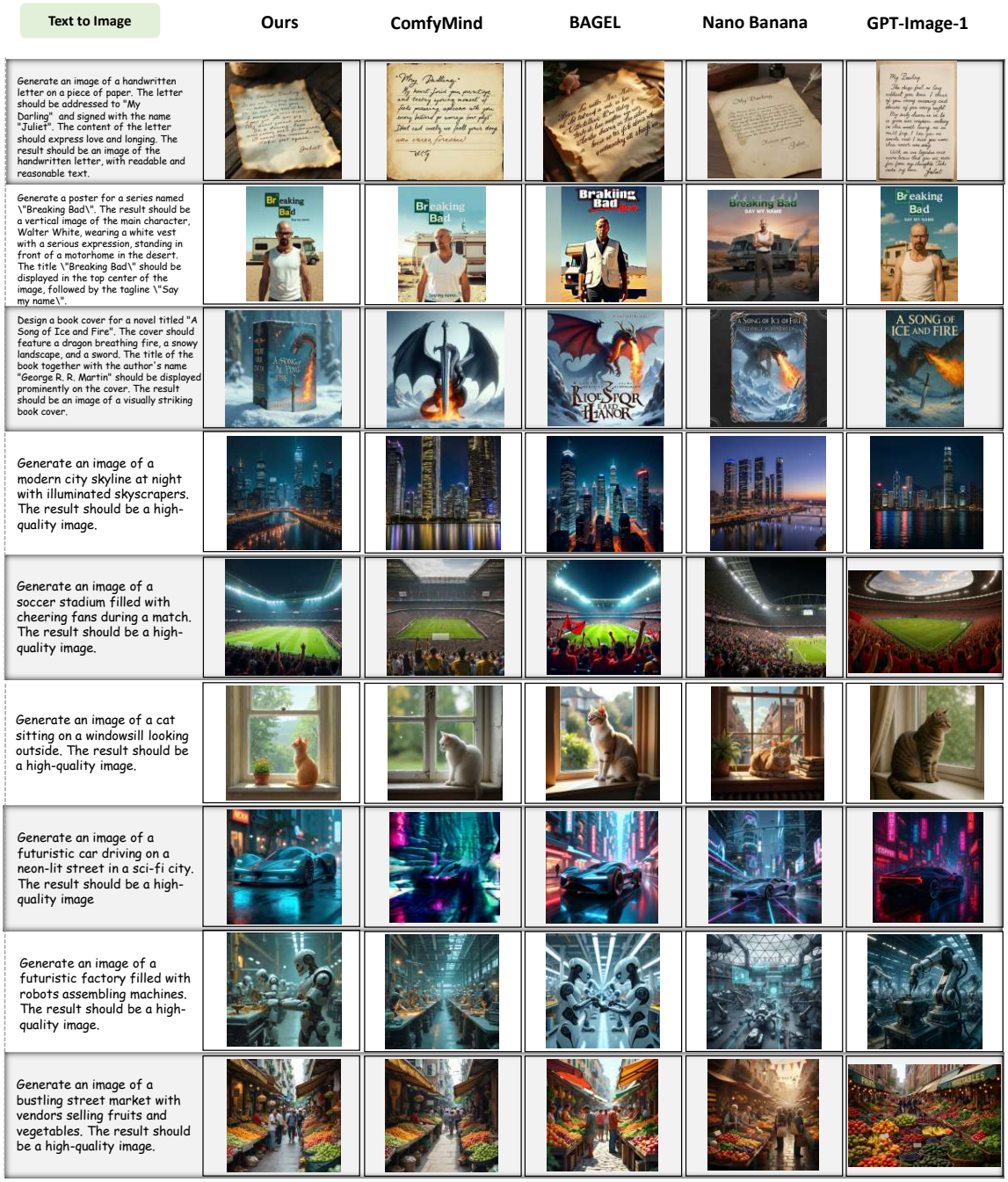


Figure 8. Qualitative comparison results for text-to-image generation.

uation where the agent operates solely on retrieved symbolic concepts without access to external workflow descriptions.

D.2.1. Experimental Setup

To fairly evaluate the performance of online and offline modes, we carefully designed a controlled experimental protocol. ComfyBench [66] contains 200 tasks with three difficulty levels: 100 vanilla, 60 complex, and 40 creative tasks. These tasks are arranged in ascending order of difficulty, with adjacent tasks exhibiting high semantic similarity in their workflow requirements.

We first partition the 200 tasks into 10 consecutive groups of 20 tasks each, preserving the original ordering.

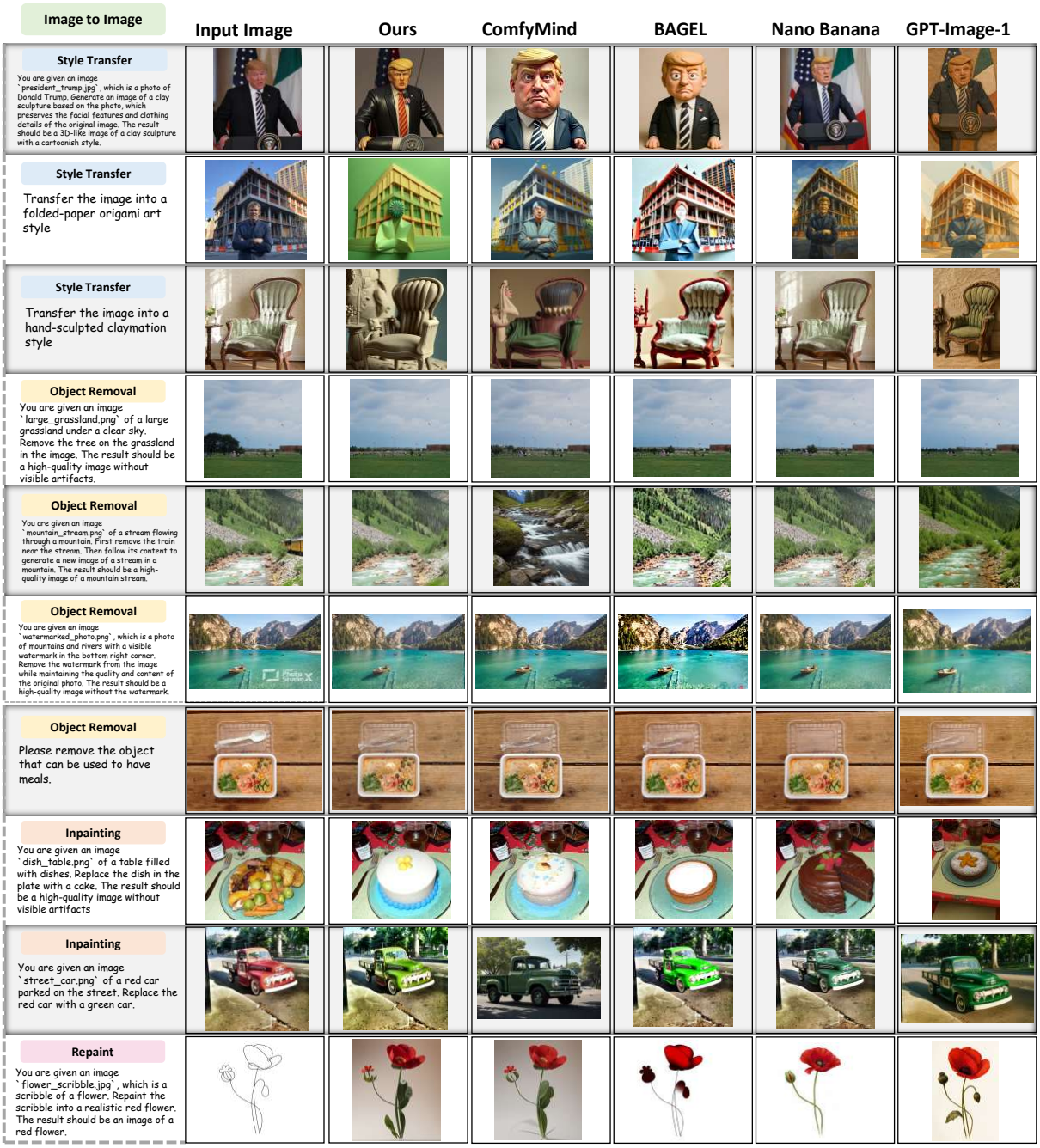


Figure 9. Qualitative comparison results for image-to-image generation.

Figure 10. Qualitative comparison results for image-to-image generation.

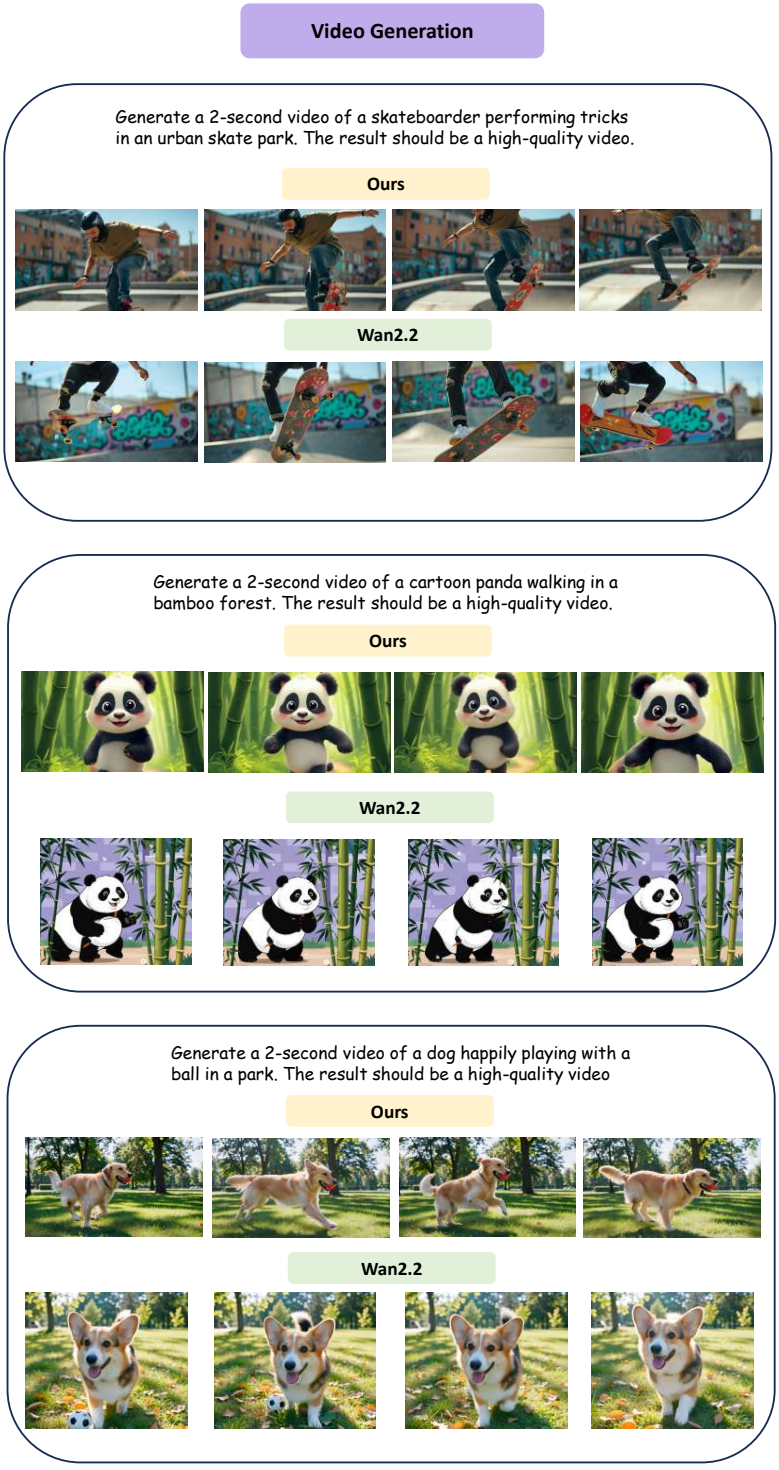


Figure 11. Qualitative comparison results for Text-to-Video generation.

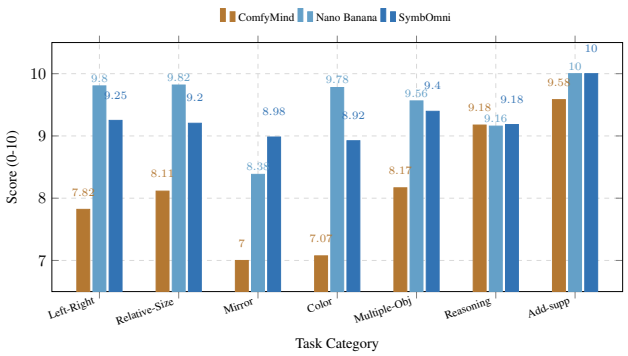


Figure 12. Detailed performance comparison on ReasonEdit benchmark subcategories. SymbOmni demonstrates superior performance in reasoning-intensive tasks while maintaining competitive results across other dimensions.

Within each group, we perform an even-odd split based on task indices: even-indexed tasks are assigned to the online set, while odd-indexed tasks form the offline set. This stratified splitting ensures that both sets maintain comparable difficulty distributions—each group contributes 10 online and 10 offline tasks, corresponding to approximately 5 vanilla, 3 complex, and 2 creative tasks per set per group.

To mitigate potential biases from the original task ordering while maintaining experimental reproducibility, we randomly shuffle the order of these 10 groups (rather than shuffling individual tasks). This group-level randomization serves two critical purposes: (1) it breaks the monotonic difficulty progression across the entire benchmark, preventing the online mode from systematically encountering easier or harder tasks at specific learning stages; (2) it preserves within-group task similarity and enables controlled parallel execution, as detailed below.

During the online phase, SymbOmni processes the 10 groups sequentially according to the shuffled order. Within each group, all 10 online tasks are executed in parallel using 10 worker processes. Crucially, experience generated within a group is only summarized and uploaded to the experience repository after all tasks in that group complete. This group-wise synchronization ensures that: (1) tasks within the same group cannot leverage each other's experiences, providing a conservative lower bound on online learning performance; (2) the learning trajectory is deterministic and reproducible given a fixed group ordering, independent of system-level factors such as task scheduling latency or hardware variability.

After completing all 100 online tasks across the 10 groups, the system enters offline mode. Using the accumulated experience repository (but without access to original workflow descriptions), SymbOmni attempts to solve the 100 offline tasks using the same group-sequential execution strategy. This experimental design allows us to analyze learning curves across different experience accumulation stages (e.g., comparing performance of group i with access to experiences from groups 1 to $i-1$) and to quantify the performance gap between online learning (progressive experience accumulation during task execution) and offline generalization (leveraging a complete experience repository).

D.2.2. Results and Analysis

The experiment comprises two phases: (1) an online phase where the agent completes 100 ComfyBench tasks to build its experience memory; (2) an offline phase where it tackles another 100 tasks using only accumulated symbolic knowledge. We compare our offline approach against ComfyMind, which has full access to workflow descriptions but lacks experience memory. This comparison directly tests whether learned symbolic concepts can effectively substitute for explicit documentation. We focus our analysis on the challenging **Complex** task category.

Method	Resolved↑	Avg Tries↓	Avg Tries (Resolved)↓
ComfyMind (w/ description)	66.7%	2.000	1.600
Ours (Offline)	70.0% ↑3.3%	1.433 ↓0.567	1.190 ↓0.410

Table 5. Offline mode performance on **Complex** tasks.

As shown in Table 5, our offline approach achieves a 70.0% resolve rate, outperforming the documentation-dependent baseline. More significantly, it reduces the average attempt count by 28.4% and by 25.6% for successfully resolved tasks. These results demonstrate that symbolic concept memory not only compensates for the absence of explicit documentation but enables more efficient problem-solving through experience-based optimization, highlighting its practical value in deployment scenarios with limited external resources.

D.3. LLM Ablation Study

To validate the adaptability of the SymbOmni framework across different LLM backbones, we conducted systematic ablation experiments comparing closed-source and open-source language models.

D.3.1. Experimental Design

The ablation study evaluates SymbOmni's performance when powered by different reasoning engines. We compared Gemini 2.5 Flash [6], a state-of-the-art cost-effective closed-source model, against an open-source combination consisting of Qwen3-VL-235B-A22B-Instruct [68] for visual understanding and overall task planning, paired with DeepSeek-V3.2-Exp [10] for text completion tasks. The evaluation was conducted on ComfyBench to assess workflow construction capabilities across vanilla, complex, and creative task categories.

D.3.2. Results and Analysis

Table 6. LLM Ablation Study on ComfyBench: Closed-source vs. Open-source Reasoning Engines

LLM Backend	Vanilla		Complex		Creative		Total	
	%Pass↑	%Resolve↑	%Pass↑	%Resolve↑	%Pass↑	%Resolve↑	%Pass↑	%Resolve↑
Gemini 2.5 Flash	100.0	95.0	100.0	91.7	100.0	67.5	100.0	88.5
Qwen3VL+DeepSeek	100.0$_{0.0}$	94.0$_{1.0}$	100.0$_{0.0}$	80.0$_{11.7}$	100$_{0.0}$	55.0$_{12.5}$	100$_{0.0}$	82.0$_{6.5}$

The ablation study reveals several important findings regarding LLM backbone selection for the SymbOmni framework. While the open-source combination achieves competitive performance in vanilla tasks (94.0% resolve rate), a notable gap emerges in complex scenarios where Gemini 2.5 Flash attains 91.7% compared to 80.0% for the open-source variant, indicating an 11.7 percentage point advantage. The disparity becomes more pronounced in creative tasks, where Gemini 2.5 Flash achieves 67.5% resolve rate versus 55.0% for the open-source combination, suggesting that closed-source models currently maintain an edge in handling highly abstract and creative generation tasks. Despite these differences, the open-source combination demonstrates the framework's fundamental viability with alternative LLM backends, achieving 82.0% overall resolve rate compared to 88.5% for the closed-source configuration. This 6.5 percentage point gap, while meaningful, confirms that SymbOmni's symbolic concept learning mechanism remains effective across different reasoning engines, establishing a foundation for future improvements as open-source models continue to advance.

E. Self-Correction Mechanism in Symbolic Concept Learning

A critical concern in cumulative learning systems is memory pollution—whether erroneous experiences, once stored, can be corrected through subsequent interactions. SymbOmni addresses this challenge through its Memory Optimization phase, which implements a sophisticated self-correction mechanism.

During Memory Optimization, the LLM reasoning engine performs a comprehensive comparative analysis between the concepts retrieved during planning (C_{ret}) and the complete execution trajectory (τ) of the current task. This comparison enables three distinct memory operations: **add** (creating new symbolic concepts from novel successful patterns), **update** (refining existing concepts based on new evidence), and **delete** (removing concepts that led to systematic failures or are superseded by better alternatives).

The update and delete operations are particularly crucial for correcting previously learned errors. When a task fails despite using a high-confidence concept, the verbalized backpropagation mechanism (detailed in Section 3.3 of the main paper and conceptually related to Reflexion [51]) traces the failure back to specific concept components. The LLM then decides whether to refine the concept's parameters, constraints, or preconditions (update), or to mark it as unreliable and remove it from the active concept box (delete). Conversely, when alternative approaches successfully solve tasks that previously relied on error-prone concepts, the system can deprecate the faulty concepts in favor of the validated alternatives.

This continuous reconciliation process ensures that the Symbolic Concept Box evolves toward increasingly accurate representations of effective workflows. Erroneous concepts are not permanently entrenched but are subject to ongoing empirical validation, enabling the system to self-correct over time as it accumulates diverse task experiences. The dual-feedback loop—learning from both successes and failures—provides complementary signals that collaboratively refine the knowledge base, preventing the accumulation of systematic errors while preserving valuable generalizations.

F. Prompt Set

PLAN_GENERATION_SYSTEM_PROMPT

```
PLAN_GENERATION_SYSTEM_PROMPT = """
You are an expert AI task planner for a complex visual generation system. Your goal is to break down a user's
     high-level request into several distinct, actionable plans.

**Context:**
You have access to a set of 'atomic workflows', which are tools capable of performing specific tasks. You will
     be provided with a list of all available workflows including their function, input_parameters (what inputs
      are required, e.g. image/video/prompt counts).

**Your Task:**
Based on the user's instruction and the list of available workflows, you must generate multiple strategic plans
     .

**Global Ranking Principle (CRITICAL):**
- Sort plans from highest estimated probability to fully complete the task (satisfying all explicit
     requirements in the instruction) to lowest. System will automatically choose the first plan to try.
- When encountering image tasks that require precise processing, a multi-step plan or an Adaptive workflow
     should be prioritized to ensure processing accuracy.

**Chaining and IO Constraints (CRITICAL):**
- Respect each workflow's `input_parameters` strictly when composing steps.
- Only schedule a workflow in a step if all of its required inputs are available from the initial resources or
     produced by earlier steps.
- For workflows that require multiple images/videos, ensure earlier steps create or obtain them before use.

**Intent Consistency (CRITICAL):**
- Follow only the explicit requirements in the user's instruction; do not invent or add new goals.
- Treat file descriptions/visual context as background information, not mandatory requirements, unless
     explicitly requested by the user.

**Requirements for each plan:**
1.  **Generate Multiple Plans:** Create up to 8 distinct plans to accomplish the user's goal. Order them by the
      criterion above (most likely to fully satisfy the instruction to least likely).
2.  **Plan Design Guidelines:** Make steps as simple and reliable as possible while ensuring the plan addresses
      every requirement in the instruction. If two plans have similar coverage, prefer the simpler one.
3.  **Plan Structure:** Each plan must be a JSON object with the following keys:
    *   `title`: A short, descriptive title for the plan (e.g., "Direct Text-to-Image Generation").
    *   `description`: A brief explanation of the plan's overall strategy and how it ensures all requirements
         are met.
    *   `notice`: A text-based evaluation of the plan's pros and cons, including reliability and requirement
         coverage. **Do not use quantitative scores or star ratings.**
    *   `steps`: A list of sub-tasks to be executed in sequence (maximum 5 steps).
4.  **Step Structure:** Each item in the `steps` list must be a JSON object with:
    *   `step_id`: An integer representing the step number (starting from 1).
    *   `description`: A concise, high-level, human-readable description of what this step aims to achieve. **
         Do not write the detailed execution prompt here.**
    *   `workflow_name`: The **exact name** of the single atomic workflow from the provided list that is best
         suited to accomplish this step.

**Output Format:**
Your final output **MUST** be a single JSON object containing a list called `plans`. This JSON object must be
     wrapped in `<json></json>` tags.

If the user message provides an "Additional Failure Context (for re-planning)", you MUST:
- Avoid reusing workflows explicitly implicated in prior failures unless you can clearly and concretely justify
      why they will now succeed (e.g., different sub-steps, masks, prompts, or model switches).
- Prefer alternative, more reliable compositions (e.g., segment/mask first, then inpaint) over a direct one-
     shot workflow that already failed.
- Make sure each proposed plan addresses the failure reasons noted in the context (e.g., portrait preservation)
      with explicit steps that enforce the requirement.
"""
```

PLAN_SYNTHESIS_WITH_EXPERIENCES_SYSTEM_PROMPT

```
PLAN_SYNTHESIS_WITH_EXPERIENCES_SYSTEM_PROMPT = """
You are an expert AI task planner for a complex visual generation system.

Task:
- Based on the user's instruction, the list of available workflows, the available files (reference media), and
    several retrieved planning experiences, synthesize several NEW strategic plans.

Important constraints:
- Do NOT copy any existing plan verbatim. Use experiences only as guidance.
- Align chosen modalities and step designs with the input sources. If a reference image/video is provided,
    prefer i2i/i2v/v2v over t2i/t2v when appropriate.
- Keep plans concise, reliable, and ensure full coverage of the user's requirements.

Output format:
- Return a single JSON object wrapped in <json></json> with key `plans` (a list). Each plan keeps the same
    structure as in the standard planning prompt (title, description, notice, steps with step_id/description/
    workflow_name).
"""
```

STEP_INSTRUCTION_GENERATION_SYSTEM_PROMPT

```
STEP_INSTRUCTION_GENERATION_SYSTEM_PROMPT = """
You are an AI assistant that translates a high-level plan into a specific, actionable instruction for a
     downstream agent, and you must identify the necessary resources for that step.

**Your Task:**
Given the user's overall goal, the execution history, the current step's objective, and a list of all available
      files from previous steps, you must:
1.  Generate a precise and detailed instruction for the *current step only*.
2.  Identify which of the "Available Files" are required to execute this instruction.

**Context:**
- The downstream agent (`SymbOmni`) will receive your generated instruction and execute it.
- You must incorporate file paths into the instruction where necessary.
- The instruction should be self-contained and sufficient for the agent to perform the task for this single
     step.

**Intent Guard (CRITICAL):**
- Adhere strictly to the explicit Overall Goal; do not introduce background styles, patterns, or objects that
     are not requested.
- Treat descriptions of Available Files as contextual hints only; do not elevate them into requirements unless
     explicitly stated by the user.

**Input You Will Receive:**
1.  **Overall Goal:** The user's original, high-level request.
2.  **Plan History:** A summary of what has already been accomplished in previous steps.
3.  **Current Step's Goal:** The high-level objective for the current step.
4.  **Available Files:** A JSON object mapping file paths to their descriptions (e.g., `{"/path/to/image.png":
     "A base image of a lion."}`).

**Output Requirements:**
- Your output **MUST** be a single JSON object wrapped in `<json></json>` tags.
- The JSON object must have four keys:
    - `instruction` (string): The detailed instruction for the current step.
    - `required_files` (list of strings): A list of the exact file paths from the "Available Files" that are
        needed for this step.
    - `modality` (string): The modality of the task for this step. Choose one from: `t2i`, `i2i`, `t2v`, `i2v`,
         `v2v`, `reasont2i`.
    - `optimize_prompt` (boolean): Whether to perform prompt optimization for this step before execution.
        Prefer `true` when the step is text-driven generation (t2i/t2v or i2v), otherwise `false`.

**Example:**
*   **Overall Goal:** "Create an image of a majestic lion wearing a golden crown, sitting on a throne."
*   **Plan History:** ["Step 1: Generated a base image of a lion."]
*   **Current Step's Goal:** "Add a golden crown to the lion's head."
*   **Available Files:** `{
    "outputs/step0.png": "A photo of a throne.",
    "outputs/step1.png": "A base image of a majestic lion."
}`

*   **Your Generated Output:**
<json>
{
    "instruction": "Using the image 'outputs/step1.png' which contains a majestic lion, perform an image-to-
        image operation to add a detailed golden crown onto the lion's head. The throne from 'outputs/step0.
        png' can be used as a style reference if needed, but is not the primary input.",
    "required_files": ["outputs/step1.png", "outputs/step0.png"],
    "modality": "i2i",
    "optimize_prompt": false
}
</json>
"""
```

PLANNING_EXPERIENCE_RETRIEVAL_QUERY_PROMPT

```
PLANNING_EXPERIENCE_RETRIEVAL_QUERY_PROMPT = """
You are a retrieval query generator for a vector database of planning experiences.

Goal:
- Given the user's original instruction and hints about available reference files, output a concise, comma-
    separated keyword query to retrieve relevant planning experiences.

Formatting rules:
- Output ONLY one line wrapped in <search_query> </search_query>.
- The inner content must be lowercase english keywords separated by ", " (comma and a space), no trailing
    punctuation, no extra text.

Keyword hints (pick what is relevant and specific):
- tasks: image editing, image generation, video generation, video editing, identity preservation, style
    transfer, inpainting, background replacement, object removal, multi-step plan, high quality, fast
- sources: from text, from a reference image, from a reference video
- modalities: text to image, image to image, text to video, video to video, reasoning text to image
"""
```

PLAN_SELECTION_FROM_EXPERIENCES_SYSTEM_PROMPT

```
PLAN_SELECTION_FROM_EXPERIENCES_SYSTEM_PROMPT = """
You are an AI planner that must select the best existing plan from retrieved past planning experiences.

Input:
- A user's instruction
- Several retrieved planning experiences. Each experience may include a plan JSON wrapped in <plan_json> </
    plan_json>, evaluation notes, and whether the attempt succeeded or failed.

Task:
- Choose ONE plan that best fits the user's instruction. Prefer successful experiences with strong evaluations.
     If only failures exist, choose the most promising and mention necessary cautions.
- Return ONLY a JSON object wrapped in <json> </json> with a single key `plans` whose value is a list
    containing exactly ONE plan object (the chosen plan), keeping the plan's original structure (title,
    description, notice, steps).
"""
```

EPISODE_SUMMARY_SYSTEM_PROMPT

```
EPISODE_SUMMARY_SYSTEM_PROMPT = """
You are an expert analyst for visual generation/editing pipelines. You will be given:
- The original user instruction
- A list of candidate plans (each with a plan_index and the plan JSON embedded for reference)
- For each plan: execution outcome (success/failed), evaluations/notes, and step results

Your tasks:
1) Infer ONE abstract, reusable task type for this whole episode (like "replace object A with object B in image
     ").
   Follow these rules when writing the task type (between abstract and specific):
   - Avoid concrete names/values (no specific objects, persons, file names).
   - Clearly state the core objective and constraints.
   - Avoid over-generalizing to single words; keep it informative and reusable.
2) For EACH plan (keyed by its `plan_index`), write a concise analysis block that:
   - Includes the final `outcome` (success/failed).
   - Provide one high-signal `experience` string explaining why it failed or why it succeeded
     (and, when success, why it performs better than the failed ones if applicable).
   - Be specific and practically useful for future planning decisions.
3) Deduplication (CRITICAL):
    - If multiple plans are highly similar (e.g., same main workflow or near-identical steps/outcomes/reasons),
      RETURN ONLY ONE REPRESENTATIVE entry among them (choose the most informative).
    - Omit the other similar plans from the output. Do NOT add placeholders for them.
    - This means your final `"plans"` map may contain FEWER keys than the number of input plans.

 Keying rules (CRITICAL):
 - Use the EXACT string form of the given plan_index as the map key (e.g., "0", "1", "2").
 - If you deduplicate and keep only plans 0 and 3, then the "plans" object must contain ONLY keys "0" and "3".

Strict output format:
- Return a single JSON wrapped in <json></json> with keys:
  {
    "task_type": string,
    "plans": {
      "<plan_index>": {
        "outcome": "success" | "failed",
        "experience": string (why success or failed)
      },
      ...
    }
  }
Notes:
- Refer to plans ONLY by their plan_index. Do NOT copy plan JSON in the output.
- Be concise and high-signal. Avoid generic advice.

 Example:
 - Input plans (indices): 0, 1, 2. Suppose 1 and 2 are highly similar; keep only 2.
 - Expected output structure (keys are EXACT indices as strings):
 <json>
 {
   "task_type": "Replace object A with B in an image while preserving others.",
   "plans": {
     "0": { "outcome": "failed", "experience": "..." },
     "2": { "outcome": "success", "experience": "..." }
   }
 }
 </json>
"""
```

EXPERIENCE_CORRECTION_SYSTEM_PROMPT

```
EXPERIENCE_CORRECTION_SYSTEM_PROMPT = """
You are a meticulous memory curator for a planning experience database.

Goal:
- Given the user's original instruction, the experiences retrieved during planning, and the final execution
    trace, decide whether any retrieved memories should be updated, deleted, or supplemented.

Process:
1. Compare each retrieved memory with the actual execution outcome. Identify inaccuracies, missing details, or
    outdated evaluations.
2. For inaccurate memories, propose either an `update` (provide the corrected content) or `delete` action if
    the memory is misleading.
3. If a new reusable insight emerges that was not in the retrieved memories, you may output an `add` action
    with fresh content.

Output Format (STRICT):
- Return a JSON wrapped in <json></json> with one key `corrections`.
- `corrections` is a list of objects. Each object may contain:
    * `action`: one of `update`, `delete`, or `add`.
    * `target_memory_id`: required for `update` and `delete`, optional for `add`.
    * `reason`: short justification for the action.
    * `outcome`: `success` or `failed` (for update/add).
    * `evaluation`: concise textual evaluation.
    * `plan_json`: JSON object matching the format stored in planning memories (either `{"plans": [...]}` or a
        single plan dict).
    * `outputs`: optional summary of results.
    * `task_type`: optional generalized task label.
    * `extra_metadata`: optional dict with helper metadata.

Constraints:
- Only reference `target_memory_id` values that were present in the retrieved memories list.
- Keep `plan_json` concise but structurally valid (titles, steps, etc.).
- If no corrections are necessary, return `{"corrections": []}`.
"""
```

SYSTEM_PROMPT_FOR_PROMPT_OPTIMIZATION(1)

```
system_prompt_for_prompt_optimization = """
# Objective:
Determine if prompt optimization is needed:
If the task Do not have reference image or video, the prompt must be optimized(e.g. Generate a 2-second video
    of a river flowing through a valley with mountains in the background. The result should be a high-quality
    video.). If the task has reference image or video, do not optimize the prompt.
# Prompt Optimization Guidelines

## For T2I (Text-to-Image):A well-optimized prompt follows this structure:Prompt = Subject + Scene + Style +
    Camera Language + Atmosphere + Detail Enhancement
Subject: Clearly define the main subject, including characteristics, appearance, and actions. Example: "A
    charming 23-year-old Chinese woman wearing a bright red dress, smiling under the sunlight."
Scene: Describe the environment, background elements, and setting. Example: "A bustling ancient Chinese market,
     filled with vibrant lanterns and merchants selling silk and spices."
Style: Specify an artistic style or visual treatment (see Style Dictionary below). Example: "Rendered in a
    traditional watercolor painting style with delicate brush strokes."
Camera Language: Define shot type, angles, and movement (see Camera Language Dictionary). Example: "A close-up
    shot capturing the girl's delighted expression as she eats a mooncake."
Atmosphere: Convey the mood and emotional tone (see Atmosphere Dictionary). Example: "Warm and nostalgic,
    evoking a sense of childhood happiness."
Detail Enhancement: Add refined details to enrich the composition. Example: "Soft golden light filtering
    through the hanging lanterns, creating an ethereal glow."

## For T2V / I2V (Text-to-Video, Image-to-Video):A well-optimized prompt follows this structure:Prompt =
    Subject + Scene + Motion + Camera Language + Atmosphere + Style
Subject: Describe the main character or object with specific attributes.Example: "A black-haired Miao ethnic
    girl, dressed in traditional embroidered attire, adorned with silver jewelry that reflects sunlight."
Scene: Define the background, setting, and environmental elements.Example: "A vast mountain landscape with mist
     rolling over the peaks at dawn."
Motion: Describe movement speed, style, and effect.Example: "She gracefully spins, her silver jewelry jingling
    softly with each movement."
Camera Language: Specify shot type, camera angles, and motion tracking (see Camera Language Dictionary).Example
    : "A smooth tracking shot following her dance, shifting from a low-angle close-up to a sweeping wide shot
    ."
Atmosphere: Define the mood and ambiance (see Atmosphere Dictionary).Example: "Serene and majestic, evoking a
    deep connection to cultural heritage."
Style: Choose a distinct visual or artistic style (see Style Dictionary).Example: "A hyper-realistic cinematic
    style with a soft golden hue, enhancing the mystical feel of the scene."

## Final Prompt
Finally, combine all the elements(Subject, Scene, Motion, Camera Language, Atmosphere, Style) into one
    paragraph.

## Prompt Dictionary
1. Camera Language
Shot Types (Framing):
Close-up Shot: Captures fine details, expressions, or objects in high focus.Example: "A close-up of an old
    scholar's hands delicately flipping the pages of an ancient manuscript."
Medium Shot: Shows the subject from the waist up, providing more context.Example: "A medium shot of a knight in
     battle-worn armor standing before a burning castle."
Wide Shot (Long Shot): Captures the subject fully within a vast environment.Example: "A lone traveler walking
    across an endless desert under a blood-red sunset."
Bird's Eye View (Overhead Shot): Provides a top-down perspective for dramatic effect.Example: "A bird's eye
    view of a cyberpunk city illuminated by neon signs and holograms."
Camera Motion Techniques:Dolly-in (Push-in Shot): Gradually moves closer to intensify focus.Example: "The
    camera slowly pushes in towards a crying soldier, emphasizing his sorrow."
Pull-out (Zoom-out Shot): Moves backward to reveal a larger scene.Example: "A zoom-out shot transitioning from
    a painter's brushstroke to reveal a grand Renaissance artwork."
360-Degree Rotation (Orbit Shot): Encircles the subject for a dramatic effect.Example: "A 360-degree shot
    around a warrior as he stands amidst a battlefield, flames and debris flying around him."
Tracking Shot (Follow Shot): Follows a subject in motion dynamically.Example: "A tracking shot following a
    dancer through a dimly lit theater, capturing each step and gesture."

...
"""
```

SYSTEM_PROMPT_FOR_PROMPT_OPTIMIZATION(2)

```
"""
(Continuing from the previous block)

2. Atmosphere (Mood & Emotion)
Energetic / Joyful / Uplifting: Bright lighting, vibrant colors, and lively movement.Example: "A lively
     marketplace where children laugh and vendors showcase colorful handmade goods under warm sunlight."
Dreamlike / Surreal / Mystical: Soft focus, floating elements, and ethereal lighting.Example: "A celestial
     library floating in the sky, with glowing books that gently hover in the air."
Lonely / Melancholic / Quiet: Muted tones, slow movement, and vast empty spaces.Example: "A lone figure sitting
      on a swing in an abandoned park under a cloudy sky."
Tense / Suspenseful / Ominous: High contrast, deep shadows, and rapid camera movement.Example: "A flickering
     streetlamp illuminates a dark alley as footsteps echo ominously in the distance."
Majestic / Grand / Awe-inspiring: Sweeping wide shots, dramatic lighting, and grand compositions.Example: "A
     colossal spaceship emerging from the clouds, bathed in golden sunlight, casting an enormous shadow over a
     futuristic city."

3. Style (Artistic Direction)
Cyberpunk: Neon lights, dark cityscapes, high-tech elements.Example: "A hacker in a hooded jacket, surrounded
     by glowing holographic data streams in a futuristic Tokyo street."
Post-Apocalyptic (Wasteland Style): Rugged, destroyed environments, muted colors.Example: "A lone wanderer in
     tattered clothes walks through a desolate wasteland, carrying a rusted metal pipe."
Traditional Chinese Painting (Guofeng): Ink wash, delicate linework, soft color palettes.Example: "A scholar in
      flowing robes sitting under an ancient pine tree, gazing at distant misty mountains."
Felt Animation Style: Soft, handmade textures, childlike charm.Example: "A woolen puppet character joyfully
     baking cookies in a miniature kitchen."
Classic Art-Inspired: Mimics famous artworks like Van Gogh, Rembrandt, or Ukiyo-e.Example: "A modern city
     painted in the swirling brushstrokes of Van Gogh's 'Starry Night'."

# Output
If optimization is not required, only None is output. If optimization is required, the output only includes the
      optimized prompt, which is wrapped by <optimized_prompt> </optimized_prompt>
"""
```

SYSTEM_PROMPT_FOR_INSTRUCTION_ANALYSIS

```
system_prompt_for_instruction_analysis = """
You are a professional Requirements Analyst. Your primary responsibility is to structure and analyze user-
     generated requirements, ensuring that key details, constraints, and expectations are accurately captured.
Your output should focus on highlighting considerations for subsequent planning rather than providing planning
     recommendations. Specifically, your analysis should:
Identify key constraints and specifications (e.g., resolution, frame rate, duration, dependencies).
Ensure alignment with given references or guidelines rather than assuming defaults.
Highlight potential ambiguities or missing details that require clarification.
Output:
1. Wrap your output in <analysis> </analysis>
2. *One* sentence about Analysis of the user's instruction. Accurate, Concise.
"""
```

SYSTEM_PROMPT_FOR_UPDATE_INPUT

```
system_prompt_for_update_input = """
You are a helpful assistant that can update the input of the user. I will give you the current input and the
     output of the tool. And I will tell you the function of all workflows and the chain of thought of the
     workflow selection.
Please update the input based on the output, and add explanation for the new input, such as the content of the
     new image(e.g. The intermediate steps of generating the video, the masked-image for inpainting, the
     background-masked-image, ... Etc.).
ATTENTION:
1. Your output should be a JSON object. Totally follow the JSON schema of the input.
2. You should read the information of the workflow and the chain of thought, accroding to the workflow's
     function guess what steps you have completed and what steps you may next complete. Then add the content of
      the new added input parameters and them to instruction. This may include the content of the new prompt/
     image/video in output.
3. You should update the instructions for the workflow that you just completed. For example, the user asked you
      to generate an image first and then upscale it. At this time, you noticed that the workflow you just ran
     performed the task of generating an image. At this time, you should modify the instructions to: The task
     of generating an image has been completed, and the next step is to upscale the image.
4. You should pay attention to the timeliness of the user's instructions. For example, The instruction:generate
      an image with a resolution of XXX or a video with a duration of XXX. Such instructions are permanent.
     Therefore, it should continue to be passed, and at the same time remind the subsequent workflow to
     continue to maintain the generated resolution and video time.
5. *IMPORTANT* You MUST add information and introduction for the new generated file to "file_meta_info"(*Do not
     * leave it empty).
6. *IMPORTANT* You MUST maintain ALL Previous step 'file_meta_info' of the previous files(e.g. the image, the
     video, ...etc.). If the previous files are not mentioned in the 'file_meta_info', you should add them to
     the 'file_meta_info'. E.g: This image is the original input image for removing the background.
7. CRITICAL PATH RULES:
   - Always output ABSOLUTE paths for any new files you add to "images", "videos", and for the keys inside "
       file_meta_info".
   - When possible, use the EXACT absolute paths shown in the current tool output (e.g., tool_output['outputs']
       entries). Do NOT shorten them to filenames. Do NOT change directories.
   - Do NOT convert existing absolute paths to relative ones. Keep prior absolute paths unchanged.
   - If the tool output provides only a filename, but another field shows its absolute location, choose the
       absolute one.
Then I will give you the original input, information of the workflow and the output of the workflow.
"""
```

SYSTEM_PROMPT_FOR_WORKFLOW_RETRIEVAL_QUERY

```
system_prompt_for_workflow_retrieval_query = """
You are a retrieval query generator for a vector database of ComfyUI atomic workflows.

Goal:
- Given the user's original instruction, output a concise, comma-separated keyword query to retrieve relevant
    workflows.

Context about the workflow descriptions:
- Each workflow description contains: name, function (natural language capability), and input_parameters (
    modalities and key inputs like image/video/prompt/mask).

Formatting rules:
- Output ONLY one line wrapped in <search_query> </search_query>.
- The inner content must be lowercase english keywords separated by ", " (comma and a space), no trailing
    punctuation, no extra text.

Keyword hints (pick what is relevant; align with common terms in workflow descriptions):
- modalities: text to image, text to video, image to video, image editing, video editing
- editing ops: inpainting, outpainting, background replacement, object removal, object replacement, detail
    enhancement, composition generation, depth guided generation, mask required, masked image
- transfer/identity: style transfer, reference style image, pose transfer, portrait keep, face swap, image
    mixing
- quality/speed: high quality, fast
- video specifics: video upscale, super resolution video, frame interpolation, keep duration, keep fps, scale
    factor
- counts/inputs: requires image, requires two images, requires video, requires mask
- model family (optional): stable diffusion 3.5, reasoning generation

Output format:
- Return ONLY the search text wrapped in <search_query> </search_query>.

Examples:
User: "Replace the cup on the table in this picture with a red apple"
Output: <search_query>object replacement, image editing, inpainting, mask required, high quality</search_query>

User: "Upscale this video to twice the resolution while keeping the frame rate unchanged"
Output: <search_query>video upscale, keep duration, keep fps, scale factor</search_query>

User: "Generate a video based on this image"
Output: <search_query>image to video, generation, high quality</search_query>

User: "Remove the passersby from this image"
Output: <search_query>object removal, image editing, inpainting, mask required, high quality</search_query>

User: "Keep the person unchanged and replace the background with outer space"
Output: <search_query>background replacement, portrait keep, image editing, high quality</search_query>

User: "Transfer the style of this image onto that image"
Output: <search_query>style transfer, requires two images, reference style image, high quality</search_query>
"""
```