OMNIHARNESS: HARNESSING GENERALIZABLE VISUAL GENERATION VIA SYMBOLIC POLICY LEARNING

Anonymous authors
Paper under double-blind review

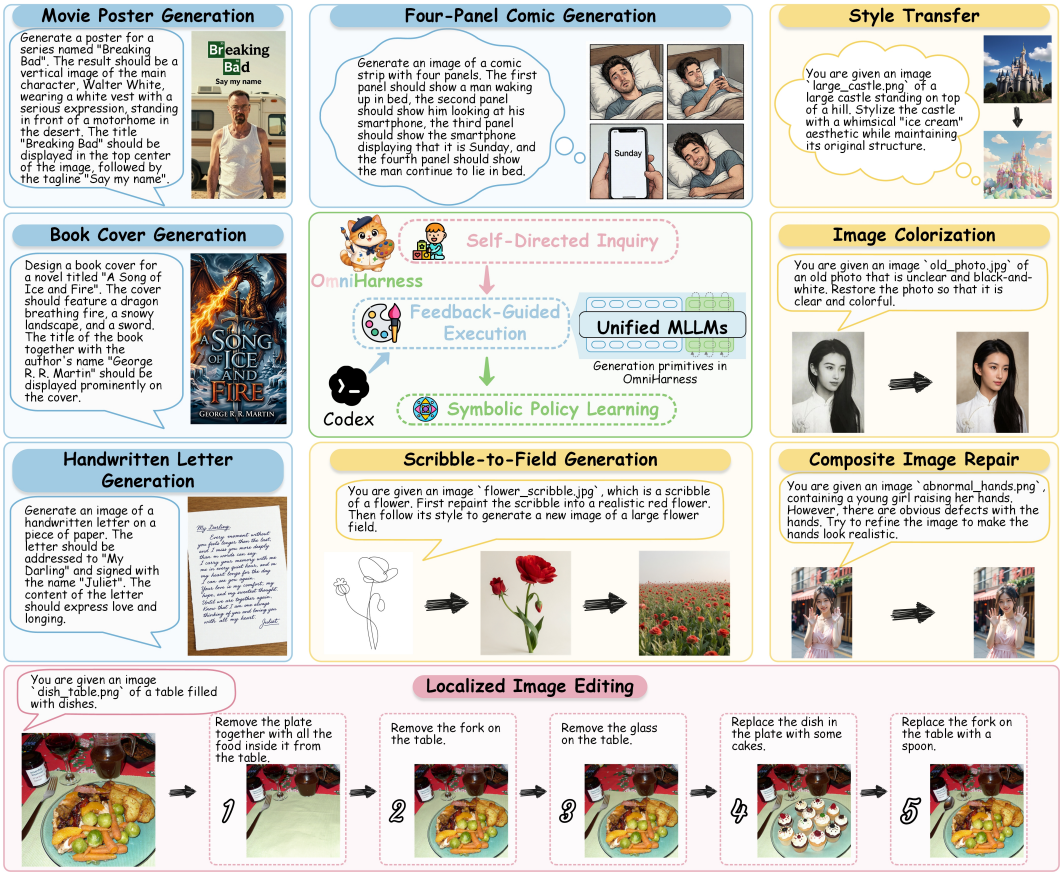


Figure 1: Overview of OmniHarness and its diverse visual generation capabilities.

ABSTRACT

Unified multimodal large language models (MLLMs) and multi-agent systems have advanced visual generation. However, three limitations remain. (1) Existing methods often distill task-specific experience with limited generalizability. (2) Reflection is often deferred until task completion. (3) Knowledge is often acquired only in response to downstream task demands. To address these limitations, we introduce **OmniHarness**, a framework for generalizable visual generation via ***symbolic policy learning***. OmniHarness abstracts verified executions into symbolic policies for visual generation task families, capturing shared procedures and applicability conditions while removing instance-specific inputs. The harness instantiates, adapts, and composes these policies for new tasks. Intermediate verification guides refinement and failure recovery during execution. Through self-directed inquiry, OmniHarness autonomously generates and executes practice

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tasks near its capability limits before downstream objectives are specified. Execution feedback continually refines the policies while model parameters remain fixed. Experiments across six benchmarks, three MLLM backbones, and three visual agent frameworks demonstrate strong performance and continual capability expansion. On ComfyBench's Creative tasks, OmniHarness achieves a 95.0% resolve rate, exceeding the strongest baseline by 27.5 percentage points. Frozen policy snapshots improve existing visual agent systems through plug-and-play reuse.

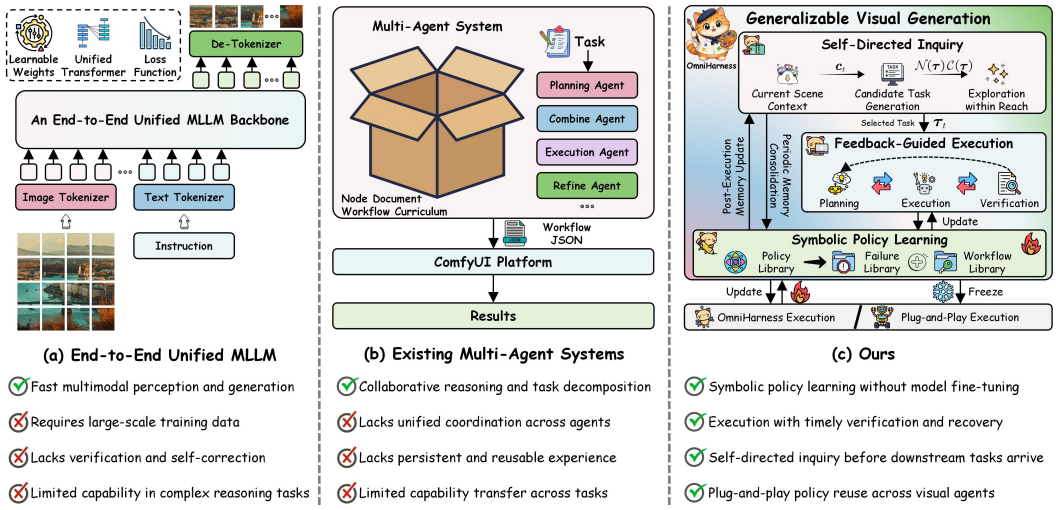


Figure 2: Comparison of visual generation paradigms. (a) End-to-end unified MLLMs enable fast multimodal generation but lack deliberate reasoning and self-correction. (b) Existing multi-agent systems improve collaborative reasoning but lack unified coordination and persistent knowledge accumulation. (c) OmniHarness integrates self-directed inquiry and feedback-guided execution to learn reusable symbolic policies for generalizable visual generation.

1 Introduction

Visual intelligence is advancing along two complementary paths for effective scaling. The first develops end-to-end unified multimodal large language models (MLLMs) for visual perception (Sun et al., 2024a; Deng et al., 2025; Fang et al., 2025; Liu et al., 2025a), multimodal reasoning (Wen et al., 2026; Zhao et al., 2026; Hao et al., 2026), and image generation (Han et al., 2026; Li et al., 2026a; Chen et al., 2026b; Zhang et al., 2026c). However, as shown in Figure 2(a), this approach relies on large-scale training data. Standalone models offer limited support for explicit verification and self-correction and can struggle with complex reasoning tasks. These limitations motivate the second path, which explores MLLM-based multi-agent systems (MAS) (Wang et al., 2026; Zhang et al., 2026b; Zhu et al., 2026; Zhang et al., 2026e).

Self-Evolving Visual MAS. Kahneman's dual-process theory distinguishes fast intuition from deliberate reasoning (Kahneman, 2011), while neuroscientific evidence suggests that language may express rather than underlie reasoning (Fedorenko et al., 2024). This perspective motivates visual MAS that combine direct generation, collaborative reasoning, and memory to learn from experience (Dang et al., 2025; Huang et al., 2026; Zeng et al., 2026). However, as illustrated in Figure 2(b), many systems still rely on manually defined ComfyUI workflows (Li et al., 2026b; Xue et al., 2025; Huang et al., 2025) or fixed communication topologies (Lin et al., 2026b;a; Xiong et al., 2026). Recent methods automate prompt or topology optimization, yet adapting coordination based on collaboration experience remains difficult (Xin et al., 2026; Liu et al., 2026b; Zhou et al., 2026). Retaining successful executions does not necessarily yield reusable skills or reliable cross-task transfer (Guo et al., 2025; Liu et al., 2026a). When confined to individual cases, such experience preserves specific solutions without revealing the principles shared across a task family, much like ***giving a fish without teaching how to fish***.

Harness Design for Self-Evolving Agents. A harness coordinates tools, workflows, and memory, shaping agent behavior alongside data and models (Ding et al., 2026; Zhang et al., 2026a). Enabling self-evolution through harness design raises three questions: ① *Can task-specific experience reveal generalizable patterns?* ***Miss the forest for the trees.*** Existing methods distill execution experience but often remain focused on individual solutions, overlooking patterns shared across a task family and limiting transfer to new tasks. ② *Can post-task reflection alone ensure reliable execution?* ***Hindsight offers lessons, but errors do not wait.*** Many existing methods reflect only after task completion, allowing intermediate errors to propagate without timely verification or recovery. ③ *Can reactive learning prepare agents for future tasks?* ***Necessity is a late teacher.*** Existing methods often acquire knowledge only in response to downstream task demands, leaving capability gaps unaddressed until they hinder execution. These challenges motivate a central question: ***How can we build a visual generation system that generalizes beyond individual cases, reflects as it acts, and learns through self-directed exploration?***

To address this central question, we introduce **OmniHarness**, a framework for generalizable visual generation via ***symbolic policy learning***. As shown in Figure 2(c), OmniHarness distills verified executions into symbolic policies for families of visual generation tasks, capturing shared procedures and applicability conditions while removing instance-specific inputs. The harness instantiates, adapts, and composes these policies for new tasks. During execution, it verifies intermediate outputs and repairs failed steps. Motivated by Chinese philosopher Wang Yangming's interpretation of *the investigation of things and the extension of knowledge* (Wang, 1916), we incorporate self-directed inquiry. Before downstream objectives are specified, OmniHarness autonomously generates and executes practice tasks to probe its capability limits and gather experience for policy learning. Execution feedback continually refines these policies while model parameters remain fixed. Frozen policy snapshots support plug-and-play reuse across visual agent frameworks.

Our contributions are summarized as follows:

- **Symbolic Policy Learning.** We distill verified executions into symbolic policies for visual generation task families, capturing shared procedures and applicability conditions while removing instance-specific inputs.
- **Feedback-Guided Execution.** We design a harness that instantiates, adapts, and composes symbolic policies for new tasks. Verification of intermediate outputs guides workflow refinement and failure recovery during execution, limiting error propagation.
- **Self-Directed Inquiry.** We introduce self-directed inquiry to learn reusable symbolic policies before downstream objectives are specified. OmniHarness autonomously generates and executes practice tasks, probing its capability limits and using execution feedback to guide policy learning.
- **Experimental Evaluation.** Experiments across six benchmarks demonstrate strong performance, continual capability expansion, and transfer across visual agent frameworks. On ComfyBench's Creative tasks, OmniHarness achieves a 95.0% resolve rate, exceeding the state-of-the-art baseline by 27.5 percentage points.

2 Related Work

End-to-End Unified MLLMs. Unified MLLMs integrate visual understanding and generation within a single architecture (Wang et al., 2024; Ge et al., 2025; Xie et al., 2025b). Recent work advances reasoning through cross-modal chain-of-thought (Liang et al., 2025; Chen et al., 2026a), multi-representation mutual reinforcement (Su et al., 2026), and shared-context visual tokenization (Peng et al., 2026). However, standard inference offers limited support for explicit verification, failure recovery, and persistent workflow reuse.

Agentic Systems. ComfyBench evaluates autonomous workflow construction in ComfyUI (Xue et al., 2025), while related systems combine planning and feedback to construct workflows for assigned tasks (Huang et al., 2025; Guo et al., 2025). Recent methods evolve execution checks and recovery for embodied agents (Ding et al., 2026), synthesize task-specific harnesses (Zhang et al., 2026a), or learn reusable symbolic concepts from incoming tasks (Liu et al., 2026a). OmniHarness learns symbolic policies from verified executions, capturing principles shared across visual generation task families. The harness adapts and composes these policies for new tasks, using intermediate

verification to guide refinement and recovery during execution. Self-directed inquiry autonomously generates and executes practice tasks to probe capability limits before downstream objectives are specified. Execution feedback continually refines the policies while model parameters remain fixed. Frozen policy snapshots support plug-and-play reuse across visual agent frameworks.

3 OMNIHARNESS: SYMBOLIC POLICY LEARNING

3.1 SELF-DIRECTED INQUIRY

Motivated by Wang Yangming's interpretation of *the investigation of things and the extension of knowledge* (Wang, 1916), OmniHarness uses self-directed inquiry to learn symbolic policies before downstream objectives are specified. Figure 3 shows the architecture. Appendix Sections A.1 and A.2 detail the formulation, learning procedure, and inquiry configuration.

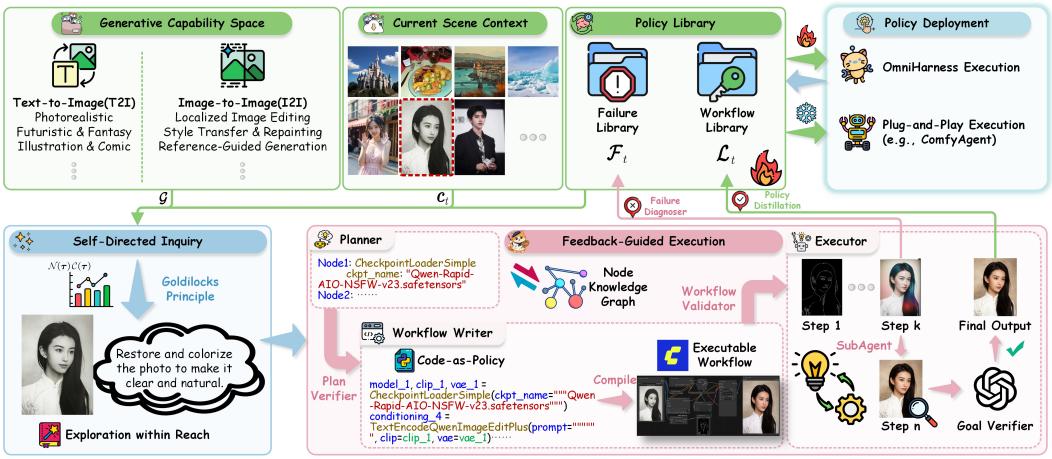


Figure 3: Architecture of OmniHarness. Self-directed inquiry and feedback-guided execution drive symbolic policy learning for generalizable visual generation. The policy library evolves during OmniHarness execution, while frozen snapshots support plug-and-play reuse across visual agents.

Candidate Task Generation. At iteration t, the proposer generates $\mathcal{T}_t = \text{Propose}(\mathcal{G}, c_t, \mathcal{S}_t)$ using capability space $\mathcal{G}$, scene context c_t, and policy library state $\mathcal{S}_t = (\mathcal{L}_t, \mathcal{F}_t)$. The context c_t summarizes capability coverage, workflow reliability, and available source images. The workflow library $\mathcal{L}_t$ stores symbolic policies as reusable workflow templates for visual generation task families, while the failure library $\mathcal{F}_t$ stores failure evidence and corrective strategies. Each candidate $\tau = (q_\tau, m_\tau, \mathcal{G}_\tau, x_\tau)$ specifies a description q_τ, modality $m_\tau \in \{\text{T2I}, \text{I2I}\}$, required capabilities $\mathcal{G}_\tau \subseteq \mathcal{G}$, and source image $x_\tau \in \mathcal{X} \cup \{\perp\}$, where $x_\tau = \perp$ for T2I. Generation encourages diverse capability combinations, filters near-duplicates, and avoids known failure patterns.

Exploration within Reach. Candidates are scored by capability novelty $\mathcal{N}(\tau)$ and competence frontier score $\mathcal{C}(\tau)$, with time indices omitted. Set $z_\tau = z_{\text{T2I}}$ for T2I and $z_\tau = x_\tau$ for I2I. Given attempt counts $n_t(z, g)$ for context z and capability g, $\mathcal{N}(\tau) = \frac{1}{|\mathcal{G}_\tau|} \sum_{g \in \mathcal{G}_\tau} \frac{1}{\sqrt{n_t(z_\tau, g)+1}}$ favors underexplored context–capability pairs. Capability support uses applicable, non-suspended workflows $\mathcal{W}_t(\tau, g) \subseteq \mathcal{L}_t$. Workflow reliability is the lower endpoint of a 95% Wilson confidence interval based on usage and success counts. The highest reliability defines $r_t(\tau, g)$, with a small prior ε when $\mathcal{W}_t(\tau, g)$ is empty. Estimated task competence follows the weakest required capability, $\bar{r}_t(\tau) = \min_{g \in \mathcal{G}_\tau} r_t(\tau, g)$. See Appendix Section A.1.2 for reliability details. Following the Goldilocks principle (Baranes & Oudeyer, 2013; Kidd et al., 2012), the learnability heuristic $\mathcal{C}(\tau) = 4\bar{r}_t(\tau)\,(1 - \bar{r}_t(\tau))$ peaks at $\bar{r}_t(\tau) = 0.5$ and downweights tasks with very low or high estimated competence. OmniHarness selects $\tau_t = \arg\max_{\tau \in \mathcal{T}_t} \mathcal{N}(\tau)\mathcal{C}(\tau)$, favoring novel tasks near the competence frontier as $\mathcal{S}_t$ evolves.



Figure 4: Qualitative comparison on challenging tasks from the Complex and Creative subsets.

3.2 Feedback-Guided Execution

For a practice or downstream task τ_t, OmniHarness executes symbolic policies through $(\pi_t, o_t) = \mathrm{Run}(\tau_t, c_t, \mathcal{K}, \mathcal{S}_t)$, where π_t is the task's executable workflow and o_t records status, verifier feedback, and evidence. Agents share plans, programs, verification results, and corrections. The planner constructs the ordered plan $\rho_t = (a_{t,1}, \ldots, a_{t,J_t})$ via $\rho_t = \mathrm{Plan}(\tau_t, c_t, \mathcal{K}, \mathcal{S}_t)$. It instantiates, adapts, and composes policies from the workflow library $\mathcal{L}_t$, guided by failure patterns and remedies in $\mathcal{F}_t$. A plan verifier checks step order, dependencies, and task alignment. The writer generates a Python-like Code-as-Policy program $\widetilde{\pi}_t = \mathrm{Write}(\rho_t, \mathcal{K})$, which a reversible interpreter compiles into $\pi_t = \mathrm{Compile}(\widetilde{\pi}_t)$. Function calls represent ComfyUI nodes, and data flow defines their connections. Workflows and components are reused when their preconditions hold. Verification checks executability, the intended effect of each $a_{t,j}$, and whether the output satisfies q_{τ_t} and the constraints in τ_t. On failure, the diagnoser identifies the affected step and retrieves a correction from $\mathcal{F}_t$. The harness repairs that component while preserving verified steps. A subagent supplies a reusable subworkflow when needed. Verification repeats until success or the retry budget is exhausted.

3.3 Symbolic Policy Learning

After each task, OmniHarness updates its context and policy library through $(c_{t+1}, \mathcal{S}_{t+1}) = \mathrm{Update}(c_t, \mathcal{S}_t, \tau_t, \pi_t, o_t)$, where $\mathcal{S}_t = (\mathcal{L}_t, \mathcal{F}_t)$. On verified success, π_t is distilled into a symbolic policy in $\mathcal{L}_{t+1}$ for its visual generation task family. This abstraction captures shared procedures and applicability conditions while removing instance-specific inputs. Equivalent workflows are merged, and usage, success, and reliability statistics are updated to guide condition-aware retrieval and composition. Failures are recorded in $\mathcal{F}_{t+1}$ with τ_t, π_t, execution evidence, and verifier feedback. Their analysis identifies root causes, workflow antipatterns, remedies, and applicable scope. A curator periodically merges redundant workflows, consolidates corrective strategies, updates reliability tiers, and may construct missing workflows. It refreshes c_{t+1} from the libraries and source image pool $\mathcal{X}$ to guide future task proposals. Updates apply to both practice and downstream tasks, continually refining the policy library. The policy library learned through self-directed inquiry is exported as a frozen snapshot $\mathcal{K}_{\mathrm{inquiry}}$ for plug-and-play reuse by external visual agents.

Table 1: Quantitative comparison on ComfyBench (Xue et al., 2025).

Agent	Vanilla		Complex		Creative		Total	
	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑
GPT-4o + Zero-shot	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GPT-4o + Few-shot (Brown et al., 2020)	32.0↑32.0	27.0↑27.0	16.7↑16.7	8.3↑8.3	7.5↑7.5	0.0=0.0	22.5↑22.5	16.0↑16.0
GPT-4o + CoT (Wei et al., 2022)	44.0↑44.0	29.0↑29.0	11.7↑11.7	8.3↑8.3	12.5↑12.5	0.0=0.0	28.0↑28.0	17.0↑17.0
GPT-4o + CoT-SC (Wang et al., 2023)	45.0↑45.0	34.0↑34.0	11.7↑11.7	5.0↑5.0	15.0↑15.0	0.0=0.0	29.0↑29.0	18.5↑18.5
Claude-3.5-Sonnet + RAG (Lewis et al., 2020)	27.0↑27.0	13.0↑13.0	23.0↑23.0	6.7↑6.7	7.5↑7.5	0.0=0.0	22.0↑22.0	8.5↑8.5
Llama-3.1-70B + RAG	58.0↑58.0	32.0↑32.0	23.0↑23.0	10.0↑10.0	15.0↑15.0	5.0↑5.0	39.0↑39.0	20.0↑20.0
GPT-4o + RAG	62.0↑62.0	41.0↑41.0	45.0↑45.0	21.7↑21.7	40.0↑40.0	7.5↑7.5	52.0↑52.0	23.0↑23.0
o1-mini + RAG	32.0↑32.0	16.0↑16.0	21.7↑21.7	8.3↑8.3	12.5↑12.5	7.5↑7.5	25.0↑25.0	12.0↑12.0
o1-preview + RAG	70.0↑70.0	46.0↑46.0	48.3↑48.3	23.3↑23.3	30.0↑30.0	12.5↑12.5	55.5↑55.5	32.5↑32.5
Llama-3.1-70B + ComfyAgent (Xue et al., 2025)	63.0	35.0	26.7	18.3	20.0	5.0	43.5	24.0
GPT-4o + ComfyAgent	67.0↑4.0	46.0↑11.0	48.3↑21.6	21.7↑3.4	40.0↑20.0	15.0↑10.0	56.0↑12.5	32.5↑8.5
GPT-4o + ComfyMind (Guo et al., 2025)	**100.0**↑37.0	92.0↑57.0	**100.0**↑73.3	**85.0**↑66.7	**100.0**↑80.0	57.5↑52.5	**100.0**↑56.5	83.0↑59.0
DeepSeek-V3 + ComfyMind	**100.0**↑37.0	90.0↑55.0	**100.0**↑73.3	71.7↑53.4	**100.0**↑80.0	60.0↑55.0	**100.0**↑56.5	78.5↑54.5
Gemini-2.5-Flash + SymbOmni (Liu et al., 2026a)	**100.0**↑37.0	95.0↑60.0	**100.0**↑73.3	83.3↑65.0	**100.0**↑80.0	67.5↑62.5	**100.0**↑56.5	86.0↑62.0
GPT-4o + OmniHarness	**100.0**↑37.0	95.0↑60.0	**100.0**↑73.3	76.7↑58.4	**100.0**↑80.0	**95.0**↑90.0	**100.0**↑56.5	89.5↑65.5
Codex GPT-4o + OmniHarness	**100.0**↑37.0	**97.0**↑62.0	**100.0**↑73.3	83.3↑65.0	**100.0**↑80.0	**95.0**↑90.0	**100.0**↑56.5	**92.5**↑68.5

Table 2: Quantitative comparison on GenEval (Ghosh et al., 2023).

Method	Single Obj.↑	Two Obj.↑	Counting↑	Colors↑	Position↑	Attr. Bind.↑	Overall↑
Frozen Text-Encoder Mapping Methods							
SDv1.5 (Rombach et al., 2022)	0.97	0.38	0.35	0.76	0.04	0.06	0.43
SDv2.1 (Rombach et al., 2022)	0.98↑0.01	0.51↑0.13	0.44↑0.09	0.85↑0.09	0.07↑0.03	0.17↑0.11	0.50↑0.07
SD-XL (Podell et al., 2024)	0.98↑0.01	0.74↑0.36	0.39↑0.04	0.85↑0.09	0.15↑0.11	0.23↑0.17	0.55↑0.12
DALL-E 2 (Ramesh et al., 2022)	0.94↓0.03	0.66↑0.28	0.49↑0.14	0.77↑0.01	0.10↑0.06	0.19↑0.13	0.52↑0.09
SD3-Medium (Esser et al., 2024)	0.99↑0.02	0.94↑0.56	0.72↑0.37	0.89↑0.13	0.33↑0.29	0.60↑0.54	0.74↑0.31
Unified Multimodal Models							
LlamaGen (Sun et al., 2024a)	0.71	0.34	0.21	0.58	0.07	0.04	0.32
LWM (Liu et al., 2025a)	0.93↑0.22	0.41↑0.07	0.46↑0.25	0.79↑0.21	0.09↑0.02	0.15↑0.11	0.47↑0.15
SEED-X (Ge et al., 2025)	0.97↑0.26	0.58↑0.24	0.26↑0.05	0.80↑0.22	0.19↑0.12	0.14↑0.10	0.49↑0.17
Emu3-Gen (Wang et al., 2024)	0.98↑0.27	0.71↑0.37	0.34↑0.13	0.81↑0.23	0.17↑0.10	0.21↑0.17	0.54↑0.22
Janus (Wu et al., 2025b)	0.97↑0.26	0.68↑0.34	0.30↑0.09	0.84↑0.26	0.46↑0.39	0.42↑0.38	0.61↑0.29
JanusFlow (Ma et al., 2025)	0.97↑0.26	0.59↑0.25	0.45↑0.24	0.83↑0.25	0.53↑0.46	0.42↑0.38	0.63↑0.31
Janus-Pro-7B (Chen et al., 2025)	0.99↑0.28	0.89↑0.55	0.59↑0.38	0.90↑0.32	0.79↑0.72	0.66↑0.62	0.80↑0.48
GoT (Fang et al., 2025)	0.99↑0.28	0.69↑0.35	0.67↑0.46	0.85↑0.27	0.34↑0.27	0.27↑0.23	0.64↑0.32
Bagel (Deng et al., 2025)	0.98↑0.27	0.94↑0.60	0.76↑0.55	0.91↑0.33	0.69↑0.62	0.70↑0.66	0.78↑0.46
GPT-Image-1	0.99↑0.28	0.92↑0.58	0.85↑0.64	0.92↑0.34	0.75↑0.68	0.61↑0.57	0.84↑0.52
Collaborative AI Systems							
ComfyAgent (Xue et al., 2025)	0.69	0.30	0.33	0.50	0.04	0.04	0.32
ComfyMind (Guo et al., 2025)	**1.00**↑0.31	**1.00**↑0.70	0.96↑0.63	0.97↑0.47	0.63↑0.59	0.81↑0.77	0.90↑0.58
SymbOmni (Liu et al., 2026a)	**1.00**↑0.31	**1.00**↑0.70	0.99↑0.66	0.98↑0.48	0.97↑0.93	0.95↑0.91	0.98↑0.66
OmniHarness	**1.00**↑0.31	**1.00**↑0.70	**1.00**↑0.67	**1.00**↑0.50	**1.00**↑0.96	**0.98**↑0.94	**0.997**↑0.677

Table 3: Quantitative comparison on GenEval2 (Kamath et al., 2025).

Method	Object↑	Attribute↑	Count↑	Position↑	Verb↑	Overall↑
Stable Diffusion Model Series						
SD 2.1 (Rombach et al., 2022)	55.1	30.4	22.3	11.7	17.8	27.46
SDXL (Podell et al., 2024)	74.1↑19.0	42.4↑12.0	28.7↑6.4	16.0↑4.3	28.9↑11.1	38.02↑10.56
SD3 (Esser et al., 2024)	87.0↑31.9	65.5↑35.1	49.9↑27.6	41.0↑29.3	46.7↑28.9	58.02↑30.56
SD3.5-Large (Esser et al., 2024)	91.6↑36.5	70.3↑39.9	52.2↑29.9	39.5↑27.8	55.6↑37.8	61.84↑34.38
State-of-the-Art Text-to-Image Models						
FLUX.1-dev (Labs et al., 2025)	88.4	68.3	55.6	37.0	44.4	58.74
Bagel + CoT (Deng et al., 2025)	92.9↑4.5	75.9↑7.6	55.6=0.0	50.6↑13.6	57.8↑13.4	66.56↑7.82
Qwen-Image (Wu et al., 2025a)	**99.1**↑10.7	85.6↑17.3	70.3↑14.7	60.2↑23.2	71.1↑26.7	77.26↑18.52
Gemini 2.5 Flash Image (Comanici et al., 2025)	99.0↑10.6	91.4↑23.1	70.1↑14.5	70.2↑33.2	86.7↑42.3	83.48↑24.74
Collaborative AI Systems						
SymbOmni (Liu et al., 2026a)	95.0	83.6	74.8	68.8	64.5	77.34
OmniHarness	95.0=0.0	**94.0**↑10.4	**94.0**↑19.2	**76.9**↑8.1	**89.0**↑24.5	**89.78**↑12.44

Table 4: Quantitative comparison on WISE (Niu et al., 2025).

Method	Cultural↑	Time↑	Space↑	Biology↑	Physics↑	Chemistry↑	Overall↑
Dedicated T2I Models							
SDv1.5 (Rombach et al., 2022)	0.34	0.35	0.32	0.28	0.29	0.21	0.32
SDv2.1 (Rombach et al., 2022)	0.30↓0.04	0.38↑0.03	0.35↑0.03	0.33↑0.05	0.34↑0.05	0.21=0.00	0.32=0.00
SD-XL (Podell et al., 2024)	0.43↑0.09	0.48↑0.13	0.47↑0.15	0.44↑0.16	0.45↑0.16	0.27↑0.06	0.43↑0.11
SD3-Medium (Esser et al., 2024)	0.42↑0.08	0.44↑0.09	0.48↑0.16	0.39↑0.11	0.47↑0.18	0.29↑0.08	0.42↑0.10
SD3.5-Medium (Esser et al., 2024)	0.43↑0.09	0.50↑0.15	0.52↑0.20	0.41↑0.13	0.53↑0.24	0.33↑0.12	0.45↑0.13
SD3.5-Large (Esser et al., 2024)	0.44↑0.10	0.50↑0.15	0.58↑0.26	0.44↑0.16	0.52↑0.23	0.31↑0.10	0.46↑0.14
PixArt-Alpha (Chen et al., 2024)	0.45↑0.11	0.50↑0.15	0.48↑0.16	0.49↑0.21	0.56↑0.27	0.34↑0.13	0.47↑0.15
Playground-v2.5 (Li et al., 2024)	0.49↑0.15	0.58↑0.23	0.55↑0.23	0.43↑0.15	0.48↑0.19	0.33↑0.12	0.49↑0.17
FLUX.1-schnell (Labs et al., 2025)	0.39↑0.05	0.44↑0.09	0.50↑0.18	0.31↑0.03	0.44↑0.15	0.26↑0.05	0.40↑0.08
FLUX.1-dev (Labs et al., 2025)	0.48↑0.14	0.58↑0.23	0.62↑0.30	0.42↑0.14	0.51↑0.22	0.35↑0.14	0.50↑0.18
Unified MLLM Models							
Janus-1.3B (Wu et al., 2025b)	0.16	0.26	0.35	0.28	0.30	0.14	0.23
JanusFlow-1.3B (Ma et al., 2025)	0.13↓0.03	0.26=0.00	0.28↓0.07	0.20↓0.08	0.19↓0.11	0.11↓0.03	0.18↓0.05
Janus-Pro-1B (Chen et al., 2025)	0.20↑0.04	0.28↑0.02	0.45↑0.10	0.24↓0.04	0.32↑0.02	0.16↑0.02	0.26↑0.03
Janus-Pro-7B (Chen et al., 2025)	0.30↑0.14	0.37↑0.11	0.49↑0.14	0.36↑0.08	0.42↑0.12	0.26↑0.12	0.35↑0.12
Show-o (Xie et al., 2025a)	0.28↑0.12	0.36↑0.10	0.40↑0.05	0.23↓0.05	0.33↑0.03	0.22↑0.08	0.30↑0.07
Show-o-512 (Xie et al., 2025a)	0.28↑0.12	0.40↑0.14	0.48↑0.13	0.30↑0.02	0.46↑0.16	0.30↑0.16	0.35↑0.12
VILA-U-7B (Wu et al., 2025c)	0.26↑0.10	0.33↑0.07	0.37↑0.02	0.35↑0.07	0.39↑0.09	0.23↑0.09	0.31↑0.08
Orthus-7B-base (Kou et al., 2025)	0.07↓0.09	0.10↓0.16	0.12↓0.23	0.15↓0.13	0.15↓0.15	0.10↓0.04	0.10↓0.13
Orthus-7B-instruct (Kou et al., 2025)	0.23↑0.07	0.31↑0.05	0.38↑0.03	0.28=0.00	0.31↑0.01	0.20↑0.06	0.27↑0.04
Emu3 (Wang et al., 2024)	0.34↑0.18	0.45↑0.19	0.48↑0.13	0.41↑0.13	0.45↑0.15	0.27↑0.13	0.39↑0.16
BAGEL (Deng et al., 2025)	0.44↑0.28	0.55↑0.29	0.68↑0.33	0.44↑0.16	0.60↑0.30	0.39↑0.25	0.52↑0.29
BAGEL + CoT (Deng et al., 2025)	0.76↑0.60	0.69↑0.43	0.75↑0.40	0.65↑0.37	0.75↑0.45	0.58↑0.44	0.70↑0.47
Closed-Source Models							
GPT-Image-1	0.81	0.71	**0.89**	0.83	0.79	0.74	0.80
Collaborative AI Systems							
ComfyMind (Guo et al., 2025)	0.85	0.66	0.72	0.67	0.70	0.78	0.76
SymbOmni (Liu et al., 2026a)	**0.90**↑0.05	0.70↑0.04	0.74↑0.02	0.75↑0.08	0.74↑0.04	0.78=0.00	0.80↑0.04
OmniHarness	0.88↑0.03	**0.85**↑0.19	0.86↑0.14	**0.84**↑0.17	**0.82**↑0.12	**0.86**↑0.08	**0.86**↑0.10

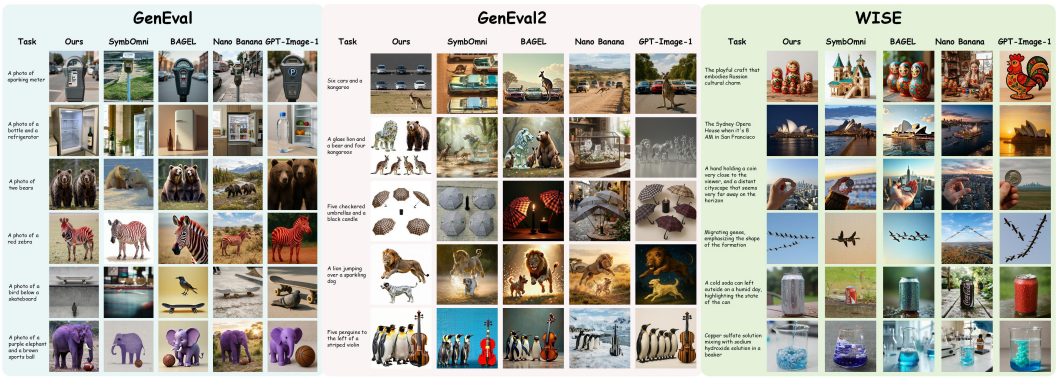


Figure 5: Qualitative comparison on representative text-to-image tasks.

4 Experiments

4.1 Autonomous Workflow Construction

We evaluate autonomous workflow construction on ComfyBench (Xue et al., 2025), where each agent must construct an executable ComfyUI workflow that satisfies the task requirements. Table 1 shows that both OmniHarness variants achieve a 100.0% Pass rate across all subsets. GPT-4o + OmniHarness and Codex GPT-4o + OmniHarness achieve Total Resolve rates of 89.5% and 92.5%, respectively. The latter exceeds SymbOmni by 6.5 percentage points overall, with the largest gain on Creative tasks, where it achieves 95.0% Resolve compared with SymbOmni's 67.5%. On Complex tasks, OmniHarness matches SymbOmni at 83.3% Resolve, while ComfyMind achieves 85.0%. Figure 4 provides qualitative examples of multi-step editing, reference-style transfer, restoration, and content preservation.

Table 5: Quantitative comparison on Reason-Edit (Huang et al., 2024).

Method	Understanding Scenarios				Reasoning Scenarios			
	PSNR↑	SSIM↑	LPIPS↓	CLIP↑	PSNR↑	SSIM↑	LPIPS↓	CLIP↑
InstructPix2Pix (Brooks et al., 2023)	21.58	0.72	0.09	22.76	24.23	0.71	0.08	19.41
MagicBrush (Zhang et al., 2023)	18.12↓3.46	0.68↓0.04	0.14↑0.05	22.62↓0.14	22.10↓2.13	0.69↓0.01	0.11↑0.03	19.76↑0.34
InstructDiffusion (Geng et al., 2024)	23.26↑1.68	0.74↑0.02	0.07↓0.02	23.08↑0.32	21.45↓2.78	0.67↓0.04	0.12↑0.03	19.52↑0.11
SmartEdit-7B (Huang et al., 2024)	22.05↑0.47	0.73↑0.01	0.09≈0.00	23.61↑0.85	25.26↑1.02	0.74↑0.04	0.06↓0.03	20.95↑1.54
SmartEdit-13B (Huang et al., 2024)	23.60↑2.02	0.75↑0.03	0.07↓0.02	23.54↑0.77	**25.76**↑1.52	0.75↑0.04	**0.05**↓0.03	20.78↑1.36
InsightEdit (Xu et al., 2025)	23.59↑2.01	0.75↑0.03	0.07↓0.02	23.73↑0.97	25.71↑1.48	0.75↑0.04	**0.05**↓0.03	20.87↑1.45
OmniHarness	**23.89**↑2.32	**0.86**↑0.14	**0.05**↓0.04	**24.55**↑1.79	23.87↓0.36	**0.80**↑0.09	**0.05**↓0.03	**21.32**↑1.91

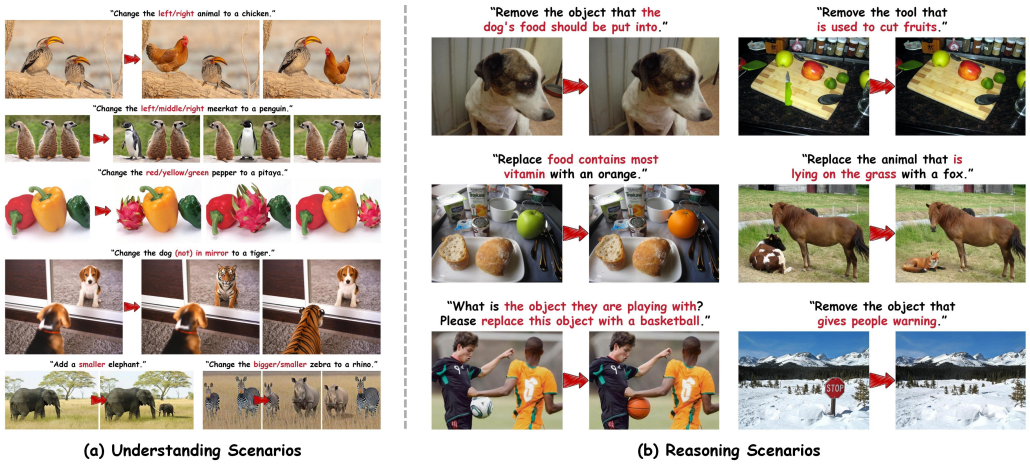


Figure 6: Qualitative results on (a) Understanding Scenarios, (b) Reasoning Scenarios.

4.2 Text-to-Image Generation

We evaluate text-to-image generation on GenEval (Ghosh et al., 2023), GenEval2 (Kamath et al., 2025), and WISE (Niu et al., 2025). GenEval measures six object-centric compositional skills, GenEval2 tests fine-grained attributes, counting, and spatial and transitive verb relations, while WISE assesses knowledge-informed synthesis across cultural, spatiotemporal, and scientific domains. For these evaluations, self-directed inquiry uses a general T2I generative capability space without access to evaluation tasks from these benchmarks. As shown in Tables 2–4, OmniHarness achieves the highest GenEval overall score of 0.997, reaching 1.00 in five categories and 0.98 in attribute binding. On GenEval2, it leads in Attribute, Count, Position, and Verb with scores of 94.0, 94.0, 76.9, and 89.0, exceeding the best competing scores by 2.6, 19.2, 6.7, and 2.3 points, respectively. Its Object score of 95.0 matches SymbOmni but remains below Qwen-Image and Gemini 2.5 Flash Image. On WISE, it achieves the highest overall WiScore of 0.86, exceeding both SymbOmni and GPT-Image-1 by 0.06. It leads in Time, Biology, Physics, and Chemistry and remains within 0.03 of the best Cultural and Space scores. Figure 5 further illustrates adherence to object counts, attribute combinations, spatial and action relations, and world-knowledge constraints.

4.3 Image Editing

We evaluate instruction-based image editing on Reason-Edit (Huang et al., 2024), which includes explicit target cues in Understanding Scenarios and indirect target descriptions in Reasoning Scenarios. During self-directed inquiry, OmniHarness may access raw source images used by ComfyBench, but downstream task instructions, target outputs, reference workflows, benchmark annotations, and evaluation labels are withheld to prevent task-level leakage. As shown in Table 5, OmniHarness leads on all four metrics in Understanding Scenarios, with 23.89 dB PSNR, 0.86 SSIM, 0.05 LPIPS, and 24.55 CLIP Score. In Reasoning Scenarios, it achieves the highest SSIM of 0.80 and CLIP Score of 21.32, while its LPIPS of 0.05 matches the best baselines at two-decimal precision. Figure 6 provides complementary qualitative evidence.

Table 6: Quantitative ablation results on ComfyBench (Xue et al., 2025).

Agent	Vanilla		Complex		Creative		Total	
	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑
w/o Self-Directed Inquiry	**100.0**=0.0	96.0↓1.0	**100.0**=0.0	81.7↓1.6	**100.0**=0.0	72.5↓22.5	**100.0**=0.0	87.0↓5.5
w/o Online Policy Updates	**100.0**=0.0	96.0↓1.0	**100.0**=0.0	76.7↓6.6	**100.0**=0.0	87.5↓7.5	**100.0**=0.0	88.5↓4.0
OmniHarness	**100.0**	**97.0**	**100.0**	**83.3**	**100.0**	**95.0**	**100.0**	**92.5**

Table 7: Ablation of task selection in self-directed inquiry on the Creative subset.

Agent	Creative T2I		Creative I2I		Creative Total	
	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑
w/o Capability Novelty	**100.0**=0.0	66.7↓16.6	**100.0**=0.0	86.7↓6.6	**100.0**=0.0	87.5↓7.5
w/o Competence Frontier	**100.0**=0.0	**83.3**=0.0	**100.0**=0.0	80.0↓13.3	**100.0**=0.0	85.0↓10.0
OmniHarness	**100.0**	**83.3**	**100.0**	**93.3**	**100.0**	**95.0**

Table 8: Ablation of feedback-guided execution on the Complex subset.

Agent	Complex T2I		Complex I2I		Complex Total	
	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑
w/o Planning	80.0↓20.0	40.0↓50.0	87.0↓13.0	56.5↓34.8	86.7↓13.3	55.0↓28.3
w/o Intermediate Verification	70.0↓30.0	60.0↓30.0	73.9↓26.1	69.6↓21.7	75.0↓25.0	68.3↓15.0
w/o Localized Recovery	90.0↓10.0	80.0↓10.0	95.7↓4.3	82.6↓8.7	93.3↓6.7	76.7↓6.6
OmniHarness	**100.0**	**90.0**	**100.0**	**91.3**	**100.0**	**83.3**

4.4 Ablation Study

Table 6 shows that removing self-directed inquiry lowers Total Resolve from 92.5% to 87.0% and Creative Resolve from 95.0% to 72.5%. Creative tasks test skill application beyond curriculum examples (Xue et al., 2025), and the larger decline supports prior policy acquisition for new generation requirements. Disabling online policy updates lowers Total Resolve to 88.5%, supporting continual refinement through execution feedback. Table 7 shows that removing capability novelty or the competence frontier score lowers Creative Resolve to 87.5% and 85.0%, respectively. Their combination outperforms either alone, supporting the complementary roles of exploration and estimated learnability. Complex tasks require combining multiple workflows (Xue et al., 2025), testing composition and coordination across dependent steps. In Table 8, removing planning, intermediate verification, or localized recovery lowers Complex Resolve from 83.3% to 55.0%, 68.3%, and 76.7%, respectively. Removing intermediate verification also reduces Complex Pass from 100.0% to 75.0%. These declines support planning for dependency coordination and intermediate feedback for workflow refinement and recovery, consistent with limiting error propagation. Additional ablation results appear in Appendix Section A.4.

5 Conclusion

Visual agents need to generalize across tasks, correct errors during execution, and learn before new task demands arise. We introduce OmniHarness, a framework for generalizable visual generation via symbolic policy learning. It abstracts verified executions into reusable symbolic policies for visual generation task families. The harness adapts and composes these policies for new tasks, with intermediate verification guiding refinement and localized recovery. Self-directed inquiry acquires policies before downstream objectives are specified, while execution feedback continually refines them without model fine-tuning. Experiments across six benchmarks demonstrate effectiveness. On ComfyBench's Creative tasks, OmniHarness achieves a 95.0% Resolve rate, exceeding the strongest baseline by 27.5 percentage points. Frozen policy snapshots improve external agents through plug-and-play reuse. Policies learned through image-only inquiry also transfer to unseen video generation tasks, supporting reuse across tasks, frameworks, and modalities.

AI USE STATEMENT

We used generative AI tools to assist with translation, language editing, and the limited drafting or restructuring of text based on content provided by us. We also used these tools to obtain feedback on the presentation of our conceptual framework and the organization of the experimental and ablation studies. Additional assistance included the formalization and typesetting of mechanisms specified by us, candidate literature search and preliminary summarization, presentation of our quantitative and qualitative results, refinement of the wording and interface text of the user-study questionnaire, and localized support for method implementation, code modification, debugging, visualization, table and figure preparation, caption editing, and reference formatting.

We independently formulated the research questions, core ideas, hypotheses, methodology, experimental protocols, benchmark settings, evaluation criteria, and scientific conclusions of this work. We independently retrieved, read, and verified all references suggested by generative AI tools. We checked all AI-assisted mathematical expressions against our intended methodology and validated all analyses against the original experimental records. We also manually inspected, executed, and tested all suggested code changes before incorporating them into the final implementation. We designed the user-study protocol and determined its evaluation criteria. Generative AI assistance was limited to refining the wording and interface presentation of the questionnaire and was not used to recruit participants, collect responses, assign ratings, or transcribe participant data. We did not use generative AI tools to fabricate or alter benchmark annotations, experimental measurements, or reported results, nor did we use them to generate formal proofs. We operated the generative models within OmniHarness and the evaluated baselines according to the protocols defined by us and described in this paper. We reviewed and revised all AI-assisted content and take full responsibility for the final text, claims, results, code, figures, and other materials presented in this submission.

ETHICS STATEMENT

We conducted a user study with 36 participants to evaluate model-generated visual results through a web-based questionnaire. Participation was voluntary, and informed consent was obtained before the study. We informed the participants of the study purpose, evaluation procedure, and intended use of their responses. The questionnaire collected only task-related evaluations and did not solicit sensitive personal information. We anonymized the responses before analysis and report only aggregate results. The study posed no more than minimal risk to the participants. The study protocol was reviewed and determined to be exempt from formal ethical review. We used the collected responses solely for research evaluation and did not use them to train the evaluated models. We take full responsibility for protecting participant privacy and for the ethical conduct and reporting of the study.

REPRODUCIBILITY STATEMENT

To support transparent review and reproducible research, we will release all source code developed for this work and the complete experimental results through an anonymous repository, and will make them publicly available upon publication. These materials will enable independent verification, reproduction, and reuse of our work. We welcome the community to build on OmniHarness and contribute further improvements, extensions, and optimizations. We hope this open release helps foster a transparent, collaborative, and sustainable research community for self-evolving visual agents.

REFERENCES

Adrien Baranes and Pierre-Yves Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2012.05.008. URL https://www.sciencedirect.com/science/article/pii/S0921889012000644.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402, 2023. doi: 10.1109/CVPR52729.2023.01764.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Junsong Chen, Jincheng YU, Chongjian GE, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 57611–57640, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/fe989bb038b5dcc44181255dd6913e43-Paper-Conference.pdf.

Leon Liangyu Chen, Haoyu Ma, Zhipeng Fan, Ziqi Huang, Animesh Sinha, Xiaoliang Dai, Jialiang Wang, Zecheng He, Jianwei Yang, Chunyuan Li, Junzhe Sun, Chu Wang, Serena Yeung-Levy, and Felix Juefei-Xu. Unit: Unified multimodal chain-of-thought test-time scaling, 2026a. URL https://arxiv.org/abs/2602.12279.

Sixiang Chen, Zhaohu Xing, Tian Ye, Xinyu Geng, Yunlong Lin, Jianyu Lai, Xuanhua He, Fuxiang Zhai, Jialin Gao, and Lei Zhu. Genevolve: Self-evolving image generation agents via tool-orchestrated visual experience distillation, 2026b. URL https://arxiv.org/abs/2605.21605.

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, 2025. URL https://arxiv.org/abs/2501.17811.

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. *arXiv preprint arXiv:2507.06261*, 2025.

Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, Xuantang Xiong, Lei Han, Zhiyuan Liu, and Maosong Sun. Multi-agent collaboration via evolving orchestration. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 165025–165059. Curran Associates, Inc., 2025. doi: 10.52202/085713-5502. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f1320d2e2842169c6fc89dcbd80e94d0-Paper-Conference.pdf.

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.

Xin Ding, Liang Mi, Mingzhe Huang, Zixuan Wang, Chao Zhang, Zixu Hao, Fu Chen, Xiangyu Li, Yikai Zheng, Yaoyu Guo, Weijun Wang, Kun Li, Hao Wu, Yunxin Liu, and Ting Cao. Zetta ζ: An efficient closed-loop embodied harness for self-evolving physical intelligence, 2026. URL https://arxiv.org/abs/2608.16590.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Hao Tian, Shilin Yan, Weihao Yu, Xingyu Zeng, Jifeng Dai, Xihui Liu, and Hongsheng Li. Got: Unleashing reasoning capability of mllm for visual generation and editing. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 67680–67708. Curran Associates, Inc., 2025. doi: 10.52202/085713-2270. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/61960fdfda4d4e95fa1c1f6e64bfe8bc-Paper-Conference.pdf.

Evelina Fedorenko, Steven T. Piantadosi, and Edward A. F. Gibson. Language is primarily a tool for communication rather than thought. *Nature*, 630(8017):575–586, June 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07522-w. URL https://doi.org/10.1038/s41586-024-07522-w.

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, 2025. URL https://arxiv.org/abs/2404.14396.

Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, Dong Chen, and Baining Guo. Instructdiffusion: A generalist modeling interface for vision tasks. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12709–12720, 2024. doi: 10.1109/CVPR52733.2024.01208.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 52132–52152. Curran Associates, Inc., 2023. doi: 10.52202/075280-2270. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a3bf71c7c63f0c3bcb7ff67c67b1e7b1-Paper-Datasets_and_Benchmarks.pdf.

Litao Guo, Xinli Xu, Luozhou Wang, Jiantao Lin, Jinsong Zhou, Zixin Zhang, Bolan Su, and Yingcong Chen. Comfymind: Toward general-purpose generation via tree-based planning and reactive feedback. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 45128–45164. Curran Associates, Inc., 2025. doi: 10.52202/085713-1503. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/40168e00bf87869c5d153e934d8a3602-Paper-Conference.pdf.

Siwei Han, Haonian Ji, Siyang Xin, Juanquan Shi, Shi Qiu, Xinyu Ye, Peng Xia, Jiaqi Liu, Zhaorun Chen, Yiyang Zhou, Linjie Li, Lijuan Wang, and Huaxiu Yao. Paper2figure: A multi-agent collaborative system for figure generation towards academic research paper. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29157–29166, June 2026.

Zhezheng Hao, Hong Wang, Jian Luo, Jianqing Zhang, Yuyan Zhou, Qiang Lin, Can Wang, Hande Dong, and Jiawei Chen. ReCreate: Reasoning and creating domain agents driven by experience. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31018–31046, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1432. URL https://aclanthology.org/2026.acl-long.1432/.

Jing Huang, Luyuan Chen, Zhijie Xu, Yadong Li, Xingzhong Xu, Siye Chen, Jie Liu, Ming Kong, and Qiang Zhu. Meta: Meta evolution of tool trajectory adaptation for long-video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9837–9846, June 2026.

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. URL https://arxiv.org/abs/2503.17671.

Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, and Ying Shan. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8362–8371, 2024. doi: 10.1109/CVPR52733.2024.00799.

Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011. ISBN 9780374275631. URL https://us.macmillan.com/books/9780374275631/thinkingfastandslow/.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. Geneval 2: Addressing benchmark drift in text-to-image evaluation, 2025. URL https://arxiv.org/abs/2512.16853.

Celeste Kidd, Steven T. Piantadosi, and Richard N. Aslin. The goldilocks effect: Human infants allocate attention to visual sequences that are neither too simple nor too complex. *PLOS ONE*, 7 (5):1–8, 05 2012. doi: 10.1371/journal.pone.0036399. URL https://doi.org/10.1371/journal.pone.0036399.

Siqi Kou, Jiachun Jin, Zhihong Liu, Chang Liu, Ye Ma, Jian Jia, Quan Chen, Peng Jiang, and Zhijie Deng. Orthus: Autoregressive interleaved image-text generation with modality-specific heads, 2025. URL https://arxiv.org/abs/2412.00127.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Chunhan Li, Qifeng Wu, Jia-Hui Pan, Ka-Hei Hui, Jingyu Hu, Yuming Jiang, Bin Sheng, Xihui Liu, Wenjuan Gong, and Zhengzhe Liu. codrawagents: A multi-agent dialogue framework for compositional image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9802–9812, June 2026a.

Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation, 2024. URL https://arxiv.org/abs/2402.17245.

Zongxia Li, Dawei Liu, Fuxiao Liu, Yuhang Zhou, Xiyang Wu, Jingxi Chen, Jing Xie, Xiaomin Wu, and Lichao Sun. Comfyclaw: Self-evolving skill harnesses for image generation workflows, 2026b. URL https://arxiv.org/abs/2607.01709.

Qian Liang, Yujia Wu, Kuncheng Li, Jiwei Wei, Shiyuan He, Jinyu Guo, and Ning Xie. Mm-r1: Unleashing the power of unified multimodal large language models for personalized image generation, 2025. URL https://arxiv.org/abs/2508.11433.

Jiashi Lin, Changhong Jiang, Xiangru Lin, Ruifei Zhang, Xinyi Zhu, Jiyao Liu, Cheng Tang, Ye Du, Shujian Gao, Junzhi Ning, et al. Evograph-r1: Self-evolving multimodal knowledge hypergraphs for agentic retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 756–765, 2026a.

Yunlong Lin, Linqing Wang, Kunjie Lin, Zixu Lin, Kaixiong Gong, Wenbo Li, Bin Lin, Zhenxi Li, Shiyi Zhang, Yuyang Peng, Wenxun Dai, Xinghao Ding, Chunyu Wang, and Qinglin Lu. Jarvisevo: Towards a self-evolving photo editing agent with synergistic editor-evaluator optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27291–27302, June 2026b.

Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with blockwise ringattention. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 45953–45977, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/71859ac75d53879d9bbd2f4b77b59929-Paper-Conference.pdf.

Jinxiu Liu, Jianru Li, Tanqing Kuang, Xuanming Liu, Kangfu Mei, Yandong Wen, and Weiyang Liu. Symbomni: Evolving agentic omni models via symbolic concept learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026a.

Mugeng Liu, Xiaojun Ma, Yuhang Xie, Qin Chen, Xuanzhe Liu, and Yun Ma. Roga: Scaling generalist agents for office productivity tasks via tool generation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 18183–18199, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/1ed4723f12853cbd02aecb8160f5e0c9-Paper-Conference.pdf.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025b.

Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, Liang Zhao, Yisong Wang, Jiaying Liu, and Chong Ruan. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7739–7751, 2025. doi: 10.1109/CVPR52734.2025.00725.

Yuwei Niu, Munan Ning, Mengren Zheng, Bin Lin, Peng Jin, Jiaqi Liao, Kun-Peng Ning, Bin Zhu, and Li Yuan. WISE: A world knowledge-informed semantic evaluation for text-to-image generation. *CoRR*, abs/2503.07265, 2025. doi: 10.48550/ARXIV.2503.07265. URL https://doi.org/10.48550/arXiv.2503.07265.

Wujian Peng, Lingchen Meng, Yuxuan Cai, Xianwei Zhuang, Yuhuan Yang, Rongyao Fang, Chenfei Wu, Junyang Lin, Zuxuan Wu, and Shuai Bai. Unified multimodal autoregressive modeling with shared context-visual tokenizer is key to unification, 2026. URL https://arxiv.org/abs/2606.18249.

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. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/081b08068e4733ae3e7ad019fe8d172f-Paper-Conference.pdf.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. URL https://arxiv.org/abs/2204.06125.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2022. doi: 10.1109/CVPR52688.2022.01042.

Zihan Su, Hongyang Wei, Kangrui Cen, Yong Wang, Guanhua Chen, Chun Yuan, and Xiangxiang Chu. Generation enhances understanding in unified multimodal models via multi-representation generation, 2026. URL https://arxiv.org/abs/2601.21406.

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*, 2024a.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14398–14409, 2024b. doi: 10.1109/CVPR52733.2024.01365.

Kun Wang, Guibin Zhang, ManKit Ye, Xinyu Deng, Dongxia Wang, Xiaobin Hu, Jinyang Guo, Yang Liu, and Yufei Guo. Mas^2: Self-generative, self-configuring, self-rectifying multi-agent systems. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 113586–113613, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/b88318174aad2cc174a4e05ab6bfad80-Paper-Conference.pdf.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need, 2024. URL https://arxiv.org/abs/2409.18869.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations*, 2023. URL https://mlanthology.org/iclr/2023/wang2023iclr-selfconsistency/.

Yangming Wang. *The philosophy of Wang Yang-ming*. Open Court Publishing Company, 1916.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Yule Wen, Yixin Ye, Yanzhe Zhang, Diyi Yang, and Hao Zhu. Real-time reasoning agents in evolving environments. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007516.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025a. URL https://arxiv.org/abs/2508.02324.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12966–12977, 2025b. doi: 10.1109/CVPR52734.2025.01210.

Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, Song Han, and Yao Lu. Vila-u: a unified foundation model integrating visual understanding and generation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 93620–93638, 2025c. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e9e140df6de01afb672cb859d203c307-Paper-Conference.pdf.

Yongliang Wu, Zonghui Li, Xinting Hu, Xinyu Ye, Xianfang Zeng, Gang Yu, Wenbo Zhu, Bernt Schiele, Ming-Hsuan Yang, and Xu Yang. Kris-bench: Benchmarking next-level intelligent image editing models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*,

volume 38, Main Conference. Curran Associates, Inc., 2025d. doi: 10.52202/085713-5242. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/e619b285582fb12f4c3de3a507b8b99c-Paper-Datasets_and_Benchmarks_Track.pdf.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13294–13304, 2025. doi: 10.1109/CVPR52734.2025.01241.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 28240–28264, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/45f0d179ef7e10eb7366550cd4e574ae-Paper-Conference.pdf.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 47490–47518. Curran Associates, Inc., 2025b. doi: 10.52202/085713-1584. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/43e8fd8b9581faa71a6a61602bc28435-Paper-Conference.pdf.

Haidong Xin, Xinze Li, Zhenghao Liu, Yukun Yan, Shuo Wang, Cheng Yang, Yu Gu, Ge Yu, and Maosong Sun. MetaMem: Evolving meta-memory for knowledge utilization through self-reflective symbolic optimization. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 5473–5492, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.270. URL https://aclanthology.org/2026.findings-acl.270/.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts LLM agents: An empirical study of experience-following behavior. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.27. URL https://aclanthology.org/2026.acl-long.27/.

Yingjing Xu, Jie Kong, Jiazhi Wang, Xiao Pan, Bo Lin, and Qiang Liu. Insightedit: Towards better instruction following for image editing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2694–2703, 2025. doi: 10.1109/CVPR52734.2025.00257.

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. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24614–24624, 2025. doi: 10.1109/CVPR52734.2025.02292.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26125–26135, 2025. doi: 10.1109/CVPR52734.2025.02433.

Ziyun Zeng, Hang Hua, and Jiebo Luo. Mira: Multimodal iterative reasoning agent for image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9563–9573, June 2026.

Guibin Zhang, Leo Lu, Fangzhou Xie, Kang Zhu, Junhao Wang, Zhifei Xie, Zhaochen Yu, Zihang Liu, Zhongxiang Sun, Qiankun Li, Yue Liao, Heng Chang, Xiaobin Hu, Qibing Ren, Wangchunshu Zhou, Chuanrui Hu, Yafeng Deng, and Shuicheng Yan. Jit-agent: Scaling harness intelligence via just-in-time harness evolution, 2026a. URL https://arxiv.org/abs/2608.25593.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng YAN. Agentracer: Who is inducing failure in the llm agentic systems? In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 11377–11399, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/134ed7a477770f227f12450ef0cbb8f4-Paper-Conference.pdf.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 31428–31449. Curran Associates, Inc., 2023. doi: 10.52202/075280-1365. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/64008fa30cba9b4d1ab1bd3bd3d57d61-Paper-Datasets_and_Benchmarks.pdf.

Shuoshuo Zhang, Zijian Li, Yizhen Zhang, Jingjing Fu, Lei Song, Jiang Bian, Jun Zhang, Yujiu Yang, and Rui Wang. Pixelcraft: A multi-agent system for high-fidelity visual reasoning on structured images. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 77608–77632, 2026c. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7d90c28e7820709792d969211815a2b3-Paper-Conference.pdf.

Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Chunyang Zhu, Jiaxing Qiu, Yuchen Yan, Jiyuan Liu, Wenhao Tang, Zhengru Fang, Yi Nie, Changxu Wei, Yu Wang, Wenbo Ding, and Chao Yu. Harness vla: Steering frozen vlas into reliable manipulation primitives via memory-guided agents, 2026d. URL https://arxiv.org/abs/2607.08448.

Yuxuan Zhang, Shijia Huang, and Liwei Wang. Intentedit: Multi-agent reasoning for intent-driven complex image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 8776–8785, June 2026e.

Weixiang Zhao, Yichen Zhang, Yingshuo Wang, Yang Deng, Yanyan Zhao, Xuda Zhi, Yongbo Huang, Hao He, Wanxiang Che, Bing Qin, and Ting Liu. On safety risks in experience-driven self-evolving agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42145–42169, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.2091. URL https://aclanthology.org/2026.findings-acl.2091/.

Chenyu Zhou, Tianyi Xu, Jianghao Lin, and Dongdong Ge. Steporlm: A self-evolving framework with generative process supervision for operations research language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 6914–6940, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0bcfb525c8f8f07ae10a93d0b2a40e00-Paper-Conference.pdf.

Zichen Zhu, Yuheng Sun, Mingxuan Zhu, Wenjie Ma, Situo Zhang, Zhexiang Wang, Ziyue Yang, Danyang Zhang, Kunyao Lan, Zihan Zhao, Dingye Liu, Siqi Xiang, Lu Chen, and Kai Yu. Iea: Amateur-friendly conversational image editing agent via three stages of multitask alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 8672–8683, June 2026.

A APPENDIX

APPENDIX CONTENTS

A.1 Problem Setup and Symbolic Policy Learning . . . 18
A.1.1 Problem Formulation and Policy Learning Loop . . . 18
A.1.2 Exploration within Reach . . . 19
A.1.3 Feedback-Guided Execution . . . 20
A.1.4 Symbolic Policy Learning and Export . . . 21
A.1.5 Overall Symbolic Policy Learning Procedure . . . 21
A.2 Self-Directed Inquiry Configuration . . . 22
A.3 Environment and Evaluation Protocols . . . 22
A.4 Fine-Grained Ablation Study . . . 23
A.4.1 Fine-Grained Ablation of Self-Directed Inquiry . . . 23
A.4.2 Fine-Grained Ablation of Feedback-Guided Execution . . . 26
A.4.3 Symbolic Policy Learning Analysis . . . 26
A.4.4 Reasoning Efficiency Analysis . . . 28
A.4.5 Backbone Ablation . . . 29
A.4.6 Source Image Pool Isolation Study . . . 30
A.5 Comprehensive Benchmark Evaluation . . . 30
A.5.1 Evaluation on ComfyBench . . . 30
A.5.2 Evaluation on GenEval . . . 31
A.5.3 Evaluation on GenEval2 . . . 31
A.5.4 Evaluation on Reason-Edit . . . 32
A.5.5 Evaluation on WISE . . . 34
A.5.6 Evaluation on KRIS-Bench . . . 35
A.6 Detailed Plug-and-Play Analysis . . . 36
A.7 Cross-Modal Generalization Evaluation . . . 36
A.8 User Study . . . 37
A.9 Limitations . . . 37
A.10 System Prompts . . . 43

A.1 PROBLEM SETUP AND SYMBOLIC POLICY LEARNING

A.1.1 PROBLEM FORMULATION AND POLICY LEARNING LOOP

This section details the operations and score computations in the main paper. We distinguish the ComfyUI knowledge graph $\mathcal{K}$, the policy library state $\mathcal{S}_t$, and the frozen policy snapshot $\mathcal{K}_{\text{inquiry}}$.

A visual workflow agent $\mathcal{A}$ receives an external task instruction ℓ, scene context c, and ComfyUI knowledge graph $\mathcal{K}$, and produces an executable workflow

$$\pi = \mathcal{A}(\ell, c, \mathcal{K}). \tag{1}$$

The instruction ℓ corresponds to the description q_τ in the structured task representation below. OmniHarness introduces self-directed inquiry to generate and execute practice tasks before downstream objectives are specified.

Given a generative capability space $\mathcal{G}$, a ComfyUI knowledge graph $\mathcal{K}$, a source image pool $\mathcal{X}$, and N inquiry iterations, OmniHarness maintains scene context c_t and policy library state $\mathcal{S}_t = (\mathcal{L}_t, \mathcal{F}_t)$. The workflow library $\mathcal{L}_t$ stores symbolic policies as reusable workflow templates for visual generation task families, together with reliability evidence. The failure library $\mathcal{F}_t$ stores failure evidence and corrective strategies. The context c_t summarizes capability coverage, workflow reliability, and available source images. Initialization sets $\mathcal{S}_1 = (\emptyset, \emptyset)$ and $c_1 = \mathrm{InitContext}(\mathcal{X})$.

At inquiry iteration t, the proposer generates candidate tasks

$$\mathcal{T}_t = \mathrm{Propose}(\mathcal{G}, c_t, \mathcal{S}_t). \tag{2}$$

The proposer considers capability coverage, available source images, reusable policies, and observed failures.

OmniHarness selects the next task according to

$$\tau_t = \arg\max_{\tau \in \mathcal{T}_t} \mathcal{N}(\tau)\mathcal{C}(\tau). \tag{3}$$

Capability novelty $\mathcal{N}(\tau)$ favors underexplored context–capability pairs, while the competence frontier score $\mathcal{C}(\tau)$ favors tasks near the estimated competence frontier. Their product balances novelty with estimated learnability. Time indices are omitted from both scores.

The harness performs feedback-guided execution through

$$(\pi_t, o_t) = \mathrm{Run}(\tau_t, c_t, \mathcal{K}, \mathcal{S}_t). \tag{4}$$

Here, π_t is the executable workflow for τ_t, and o_t records execution status, verifier feedback, and evidence, including intermediate outputs and recovery results. The operation Run includes planning, Code-as-Policy synthesis, execution, intermediate verification, and localized recovery. It uses $\mathcal{K}$ to construct workflows, instantiates, adapts, and composes policies from $\mathcal{L}_t$, and consults failure patterns and remedies in $\mathcal{F}_t$.

After execution, OmniHarness updates its context and policy library

$$(c_{t+1}, \mathcal{S}_{t+1}) = \mathrm{Update}(c_t, \mathcal{S}_t, \tau_t, \pi_t, o_t). \tag{5}$$

Verified executions are abstracted into symbolic policies in $\mathcal{L}_{t+1}$, capturing shared procedures and applicability conditions for visual generation task families while removing instance-specific inputs. Failure evidence and corrective strategies are recorded in $\mathcal{F}_{t+1}$. Updates apply to both practice and downstream tasks while model parameters remain fixed. The exported snapshot $\mathcal{K}_{\mathrm{inquiry}}$ remains frozen for plug-and-play reuse by external visual agents.

A.1.2 Exploration within Reach

Each candidate task is represented as $\tau = (q_\tau, m_\tau, \mathcal{G}_\tau, x_\tau)$, where q_τ is the task description, $m_\tau \in \{\mathrm{T2I}, \mathrm{I2I}\}$ is the generation modality, and $\mathcal{G}_\tau \subseteq \mathcal{G}$ is the required capability set. The source image is $x_\tau \in \mathcal{X} \cup \{\perp\}$, with $x_\tau = \perp$ for T2I. Candidate generation encourages diverse capability combinations, filters near-duplicates, and avoids known failure patterns in $\mathcal{F}_t$.

Capability novelty. Let z_τ denote the task's context key. For T2I, $z_\tau = z_{\mathrm{T2I}}$, where z_{T2I} is a fixed context token. For I2I, $z_\tau = x_\tau$. Let $n_t(z, g)$ count practice attempts involving context z and capability g before iteration t. Capability novelty is

$$\mathcal{N}(\tau) = \frac{1}{|\mathcal{G}_\tau|} \sum_{g \in \mathcal{G}_\tau} \frac{1}{\sqrt{n_t(z_\tau, g) + 1}}. \tag{6}$$

Underexplored pairs receive higher scores. Averaging over $\mathcal{G}_\tau$ prevents a larger capability set from increasing novelty solely through its size.

Conservative workflow reliability. Novelty alone may favor tasks with insufficient support from the workflow library. OmniHarness therefore estimates reliability before computing the competence frontier score. For each required capability $g \in \mathcal{G}_\tau$, let $\mathcal{W}_t(\tau, g) \subseteq \mathcal{L}_t$ contain applicable, non-suspended workflows that support g. Each workflow w records a usage count $u_w \geq 1$ and verified success count s_w, with empirical success rate $p_w = s_w / u_w$.

Reliability is the lower endpoint of a 95% Wilson confidence interval

$$\widehat{r}(w) = \frac{p_w + \frac{\kappa^2}{2u_w} - \kappa\sqrt{\frac{p_w(1-p_w)}{u_w} + \frac{\kappa^2}{4u_w^2}}}{1 + \frac{\kappa^2}{u_w}}, \qquad \kappa = 1.96. \tag{7}$$

This discounts success rates supported by limited execution evidence.

Capability support is estimated from the most reliable applicable workflow

$$r_t(\tau, g) = \begin{cases} \max\limits_{w \in \mathcal{W}_t(\tau,g)} \widehat{r}(w), & \mathcal{W}_t(\tau, g) \neq \emptyset, \\ \varepsilon, & \mathcal{W}_t(\tau, g) = \emptyset, \end{cases} \tag{8}$$

where $\varepsilon = 0.05$ provides a small prior when no applicable workflow exists.

Estimated task competence follows the weakest required capability

$$\bar{r}_t(\tau) = \min_{g \in \mathcal{G}_\tau} r_t(\tau, g). \tag{9}$$

Competence frontier score. Following the Goldilocks principle (Baranes & Oudeyer, 2013; Kidd et al., 2012), OmniHarness uses the learnability heuristic

$$\mathcal{C}(\tau) = 4\bar{r}_t(\tau)\left(1 - \bar{r}_t(\tau)\right). \tag{10}$$

It peaks at $\bar{r}_t(\tau) = 0.5$ and downweights tasks with very low or high estimated competence. Combined with capability novelty in equation 3, it favors novel tasks near the competence frontier. Task selection adapts as attempts accumulate and $\mathcal{S}_t$ changes.

A.1.3 Feedback-Guided Execution

Agents share the task and context $(\tau_t, c_t, \mathcal{K}, \mathcal{S}_t)$, together with plans, programs, verification results, and corrections.

The planner decomposes τ_t into an ordered plan $\rho_t = (a_{t,1}, \ldots, a_{t,J_t})$, where J_t is the number of steps and $a_{t,j}$ is the j-th generation step

$$\rho_t = \mathrm{Plan}(\tau_t, c_t, \mathcal{K}, \mathcal{S}_t). \tag{11}$$

It uses $\mathcal{K}$ to construct the plan and instantiates, adapts, and composes policies from $\mathcal{L}_t$, guided by failure patterns and remedies in $\mathcal{F}_t$. The plan specifies step dependencies and verification criteria. A plan verifier checks step order, dependencies, and task alignment.

The workflow writer converts the verified plan into a Python-like Code-as-Policy program

$$\widetilde{\pi}_t = \mathrm{Write}(\rho_t, \mathcal{K}). \tag{12}$$

Function calls represent ComfyUI nodes, arguments specify node parameters, and data flow defines their connections. Workflows and components are reused when their preconditions hold.

A reversible interpreter compiles the program into an executable workflow

$$\pi_t = \mathrm{Compile}(\widetilde{\pi}_t). \tag{13}$$

The resulting π_t is the execution instance for τ_t, preserving the program's node types, parameters, dependencies, and execution order. The executor runs π_t and records intermediate outputs.

Verification covers four levels. The plan verifier checks the plan before synthesis. The workflow validator checks graph validity and executability in the current ComfyUI environment. Intermediate verification checks each output against the intended effect of $a_{t,j}$. The goal verifier checks the final output against q_{τ_t} and the constraints in τ_t. Verifier feedback and evidence are recorded in o_t.

On failure, the diagnoser identifies the affected step, analyzes its cause, and retrieves a correction from $\mathcal{F}_t$. The harness repairs that component while preserving verified steps. If the failure persists, a subagent solves the affected step and returns a reusable subworkflow. The repaired component is integrated into π_t and verified again. Verification and recovery repeat until success or the retry budget is exhausted.

A.1.4 SYMBOLIC POLICY LEARNING AND EXPORT

When o_t confirms success, π_t is abstracted into a symbolic policy represented by a workflow entry w in $\mathcal{L}_{t+1}$. The entry stores a reusable workflow template, modality, capability categories, description, preconditions, expected effects, and dependencies. The template retains the shared ComfyUI graph structure while removing instance-specific inputs, including q_{τ_t} and x_{τ_t}. Its procedures and applicability conditions support instantiation, adaptation, and composition for new tasks within a visual generation task family.

Before insertion, OmniHarness compares the entry with existing workflows by function and executable structure. Equivalent entries are updated rather than duplicated. Each invocation of w increments u_w, while s_w increases only after verified success.

A new entry starts in the *Provisional* tier. It is promoted to *Validated* when $u_w \geq 3$ and $p_w \geq 0.5$, and assigned to *Suspended* when $u_w \geq 10$ and $p_w \leq 0.2$. Validated workflows are prioritized during retrieval, while suspended workflows are excluded from normal reuse. The empirical rate p_w controls tier updates, whereas $\widehat{r}(w)$ provides the conservative reliability estimate for task selection.

Failures are recorded in $\mathcal{F}_{t+1}$ with τ_t, π_t, input conditions, generation constraints, intermediate outputs, execution feedback, and verifier results from o_t. Their analysis yields corrective strategies specifying root causes, workflow antipatterns, remedies, and applicable scope. The scope may include modalities, capability categories, workflow stages, nodes, or models.

A curator periodically consolidates $\mathcal{L}_t$ and $\mathcal{F}_t$. It merges redundant workflows, refines their preconditions and expected effects, consolidates corrective strategies, and updates statistics and reliability tiers. When repeated failures reveal a missing capability, it may trigger workflow construction and verification. The updated libraries and source image pool $\mathcal{X}$ refresh c_{t+1} to guide future task proposals.

After N inquiry iterations, OmniHarness exports a frozen copy of the learned policy library

$$\mathcal{K}_{\text{inquiry}} = \text{Snapshot}(\mathcal{S}_{N+1}). \tag{14}$$

The snapshot contains both the workflow library and the failure library. It supports plug-and-play reuse by external visual agents and remains fixed during evaluation.

For downstream OmniHarness execution, let $\mathcal{D} = (\ell_j)_{j=1}^{M}$ denote a stream of M external task instructions. Execution starts from the final inquiry state

$$(c, \mathcal{S}) = \text{Execute}\left(\mathcal{D}, c_{N+1}, \mathcal{K}, \mathcal{S}_{N+1}\right). \tag{15}$$

Each instruction is represented as a downstream task τ_t and processed by the same feedback-guided execution and policy update operations. The internal policy library $\mathcal{S}$ continues to learn from downstream feedback while model parameters remain fixed. The exported snapshot $\mathcal{K}_{\text{inquiry}}$ receives no downstream updates.

A.1.5 OVERALL SYMBOLIC POLICY LEARNING PROCEDURE

Algorithm 1 summarizes the complete OmniHarness procedure. Self-directed inquiry generates practice tasks before downstream objectives are specified. Feedback-guided execution provides verified experience for learning symbolic policies for visual generation task families. These policies are adapted and refined during downstream execution, while a frozen snapshot of the inquiry state supports plug-and-play reuse.

Algorithm 1 Symbolic Policy Learning in OmniHarness

Require: Generative capability space $\mathcal{G}$, ComfyUI knowledge graph $\mathcal{K}$, source image pool $\mathcal{X}$, inquiry iterations N, and downstream task stream $\mathcal{D} = (\ell_j)_{j=1}^{M}$
Ensure: Evolving policy library state $\mathcal{S} = (\mathcal{L}, \mathcal{F})$ and frozen policy snapshot $\mathcal{K}_{\text{inquiry}}$

$\mathcal{S}_1 \leftarrow (\emptyset, \emptyset)$, $c_1 \leftarrow \text{InitContext}(\mathcal{X})$ ▷ initialize the context and policy library
for $t = 1, \ldots, N$ **do**
  $\mathcal{T}_t \leftarrow \text{Propose}(\mathcal{G}, c_t, \mathcal{S}_t)$ ▷ Step 1: propose and select a practice task
  $\tau_t \leftarrow \arg\max_{\tau \in \mathcal{T}_t} \mathcal{N}(\tau)\mathcal{C}(\tau)$
  $(\pi_t, o_t) \leftarrow \text{Run}(\tau_t, c_t, \mathcal{K}, \mathcal{S}_t)$ ▷ Step 2: plan, execute, verify, and recover
  $(c_{t+1}, \mathcal{S}_{t+1}) \leftarrow \text{Update}(c_t, \mathcal{S}_t, \tau_t, \pi_t, o_t)$ ▷ Step 3: update the policy library
end for
$(c, \mathcal{S}) \leftarrow \text{Execute}(\mathcal{D}, c_{N+1}, \mathcal{K}, \mathcal{S}_{N+1})$ ▷ Step 4a: adapt and refine policies downstream
$\mathcal{K}_{\text{inquiry}} \leftarrow \text{Snapshot}(\mathcal{S}_{N+1})$ ▷ Step 4b: export the frozen inquiry snapshot
return $\mathcal{S}, \mathcal{K}_{\text{inquiry}}$

A.2 Self-Directed Inquiry Configuration

Unless otherwise specified by an ablation, experiments start from the policy library state $\mathcal{S}_{N+1}$ learned through $N = 50$ iterations of self-directed inquiry. At each iteration t, the proposer generates ten candidate practice tasks, with $|\mathcal{T}_t| = 10$, conditioned on the generative capability space $\mathcal{G}$, scene context c_t, and policy library state $\mathcal{S}_t = (\mathcal{L}_t, \mathcal{F}_t)$. The curator consolidates the workflow library $\mathcal{L}_t$ and failure library $\mathcal{F}_t$ every five inquiry iterations. For I2I practice, the source image pool $\mathcal{X}$ contains all original ComfyBench images and excludes video inputs. These raw images are available during inquiry, while downstream task instructions, target outputs, reference workflows, benchmark annotations, and evaluation labels are withheld.

The full image-only capability space $\mathcal{G}$ contains six T2I and six I2I capabilities, listed in Table 9. For GenEval, GenEval2, and WISE, inquiry uses only the T2I capabilities without access to benchmark evaluation tasks.

The policy library $\mathcal{S}_t = (\mathcal{L}_t, \mathcal{F}_t)$ combines reusable workflow templates with failure evidence and corrective strategies. Verified executions are abstracted into symbolic policies in $\mathcal{L}_t$, capturing shared procedures and applicability conditions for visual generation task families while removing instance-specific inputs. The harness instantiates, adapts, and composes these policies for downstream tasks. Execution feedback continually refines the policy library while model parameters remain fixed. External visual agents reuse the frozen snapshot $\mathcal{K}_{\text{inquiry}}$, which contains both libraries. Table 10 lists the workflow library schema.

The failure library $\mathcal{F}_t$ stores failure evidence and corrective strategies used during feedback-guided execution. Table 11 lists its schema.

The scene context c_t summarizes capability coverage, workflow reliability, and available source images. It is updated after each task and refreshed during periodic library consolidation. Table 12 lists its fields.

A.3 Environment and Evaluation Protocols

All experiments were conducted within ComfyUI. We evaluated two reasoning configurations, GPT-4o and Codex GPT-4o. In the first configuration, GPT-4o directly served as the reasoning engine of OmniHarness. Following the planner-instantiation design of Harness VLA (Zhang et al., 2026d), the second configuration used Codex as the agentic planner and GPT-4o as its underlying inference model. Table 1 reports both *GPT-4o + OmniHarness* and *Codex GPT-4o + OmniHarness*. Unless otherwise specified, all subsequent OmniHarness results use Codex GPT-4o. For all reasoning calls, we set `temperature` to 0 and `top_p` to 1. For fair comparison across agentic systems, we fixed the maximum number of retries at 4 (Liu et al., 2026a). Within each evaluation setting, workflows constructed by OmniHarness and the compared agents invoked the same underlying generative models (Xue et al., 2025; Guo et al., 2025; Liu et al., 2026a). Unified generative models and published baselines followed their respective benchmark protocols.

Table 9: Generative capability space $\mathcal{G}$ for self-directed inquiry.

Modality	Capability Category	Description
Text-to-Image (T2I)	Photorealistic Generation	Generates realistic people, animals, objects, and indoor and outdoor scenes.
	Futuristic & Fantasy Generation	Generates science-fiction, futuristic, and fantasy scenes, objects, and concepts.
	Illustration & Comic Generation	Generates illustrations, cartoons, comics, and multi-panel visual narratives.
	Poster & Graphic Design	Generates posters, covers, advertisements, and other graphic compositions with specified layouts.
	In-Image Text Generation	Renders specified text with correct content, legibility, appearance, and placement.
	Position-Constrained Generation	Places objects or visual elements in specified image regions or locations.
Image-to-Image (I2I)	Localized Image Editing	Inserts, removes, replaces, or modifies selected content while preserving unrelated regions.
	Style Transfer & Repainting	Changes visual style while preserving the main semantic content and structure.
	Reference-Guided Generation	Generates new content using reference properties such as pose, structure, style, or semantics.
	Image Super-Resolution	Improves resolution and detail while preserving semantic content and global structure.
	Image Outpainting	Extends the image beyond its original boundaries while preserving semantic, structural, and visual consistency.
	Image Restoration & Refinement	Repairs degraded content, corrects artifacts, and restores detail to improve visual quality.

A.4 Fine-Grained Ablation Study

Table 6 in the main paper evaluates self-directed inquiry and online policy updates. *w/o Self-Directed Inquiry* removes inquiry while retaining downstream updates to the workflow and failure libraries, assessing the benefit of learning before downstream tasks arrive. *w/o Online Policy Updates* retains the policy library learned through inquiry but disables updates to both libraries during downstream execution. Policy instantiation, adaptation, composition, and feedback-guided execution remain active. This variant assesses continual policy refinement through execution feedback. *OmniHarness* enables both mechanisms.

We conduct six fine-grained ablation and diagnostic studies. First, we examine capability novelty $\mathcal{N}(\tau)$, the competence frontier score $\mathcal{C}(\tau)$, and the number of inquiry iterations. Second, we ablate planning, intermediate verification, and localized recovery in feedback-guided execution. Third, we analyze symbolic policy learning through workflow library growth, reliability, reuse, composition, capability distribution, and modality expansion. Fourth, we measure reasoning efficiency through agent-side token consumption and reasoning time across task difficulty levels. Fifth, we evaluate different reasoning backbones and planner configurations. Finally, we replace ComfyBench source images with independently generated images from the same capability space to assess whether the benefits of inquiry depend on the original source pool.

A.4.1 Fine-Grained Ablation of Self-Directed Inquiry

After selecting τ_t, OmniHarness constructs or adapts a workflow through feedback-guided execution, including planning, Code-as-Policy synthesis, execution, intermediate verification, and localized recovery. When o_t confirms success, OmniHarness abstracts the verified workflow π_t into a symbolic policy represented by a reusable workflow entry w. Distillation removes instance-specific inputs, including the task description q_{τ_t} and, for I2I tasks, the source image x_{τ_t}. Shared procedures

Table 10: Schema of the workflow library $\mathcal{L}_t$.

Field	Description
Modality	Specifies whether the workflow supports T2I or I2I generation.
Capability Categories	Records the capabilities in $\mathcal{G}$ supported by the workflow.
Workflow Template	Stores a reusable ComfyUI graph with node types, shared parameters, connections, and execution settings. Instance-specific inputs are removed and supplied when the policy is instantiated for a new task.
Workflow Name	Provides a unique identifier for indexing, retrieval, and invocation.
Workflow Description	Summarizes the shared procedure and the task family it supports.
Preconditions	Specifies required input modalities, image properties, task conditions, available models, and execution resources.
Expected Effects	Describes the intended generation or transformation result when the preconditions hold.
Dependencies	Records required ComfyUI nodes, custom nodes, generative and auxiliary models, and external resources.
Usage Count u_w	Counts invocations of workflow w during inquiry and downstream OmniHarness execution.
Success Count s_w	Counts invocations verified as successful. The empirical success rate is $p_w = s_w/u_w$.
Reliability Tier	A new workflow starts as *Provisional*. It becomes *Validated* when $u_w \geq 3$ and $p_w \geq 0.5$, and *Suspended* when $u_w \geq 10$ and $p_w \leq 0.2$.

Table 11: Schema of the failure library $\mathcal{F}_t$.

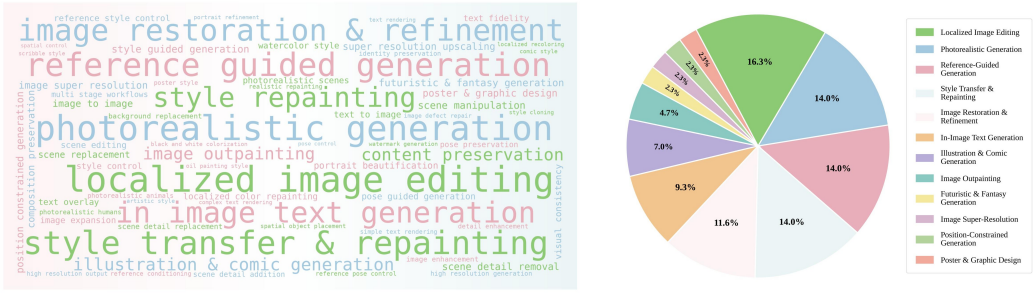
Entry Type	Field	Description
Execution Failure Record	Task τ_t	Stores the description q_{τ_t}, modality m_{τ_t}, required capabilities $\mathcal{G}_{\tau_t}$, and source image x_{τ_t} when applicable.
	Failure Context	Records the input conditions and generation constraints under which the failure occurred.
	Failed Workflow π_t	Stores the failed task's ComfyUI workflow, including nodes, parameters, connections, and execution settings.
	Failure Evidence o_t	Preserves execution status, intermediate and final outputs, source images when applicable, execution feedback, and verifier results.
Corrective Strategies	Root Cause	Identifies the workflow or generation factor responsible for the failure.
	Workflow Antipattern	Describes workflow structures, node configurations, parameter settings, or generation strategies to avoid.
	Remedy	Specifies a correction or alternative workflow strategy for the identified failure.
	Applicable Scope	Defines the modalities, capabilities, workflow stages, nodes, or models to which the corrective strategy applies.

and applicability conditions support reuse across a visual generation task family. As summarized in Table 10, each entry records its modality m_{τ_t}, capability categories $\mathcal{G}_{\tau_t}$, reusable ComfyUI workflow template, name, description, preconditions, expected effects, dependencies, and reliability statistics.

Before insertion, the distilled entry is compared with workflows in $\mathcal{L}_t$ by intended function and executable structure. If an equivalent workflow exists, the verified execution updates that entry, incrementing its usage count u_w and success count s_w. The empirical success rate $p_w = s_w/u_w$,

Table 12: Fields of the current scene context c_t.

Field	Description
Modality Coverage	Counts stored T2I and I2I workflows to identify imbalances in modality coverage.
Capability Coverage	Counts workflows associated with each capability in $\mathcal{G}$ to identify gaps in workflow coverage.
Reliability Distribution	Counts *Provisional*, *Validated*, and *Suspended* workflows in $\mathcal{L}_t$.
Source Image Pool $\mathcal{X}$	Contains all original ComfyBench images for I2I practice. Video inputs are excluded.
Source Image Metadata	Records image file names, resolutions, and formats to support image selection and task generation.



(a) Word Cloud of Workflow Library Capabilities (b) Workflow Library Distribution

Figure 7: Visualization of the workflow library after 50 iterations of self-directed inquiry. (a) Word cloud summarizing the capabilities represented by stored policies. (b) Distribution of workflow entries across capability categories.

conservative reliability estimate $\widehat{r}(w)$, and reliability tier are then updated. Otherwise, w is inserted into $\mathcal{L}_{t+1}$ as a new *Provisional* workflow entry.

After $N = 50$ inquiry iterations, the workflow library $\mathcal{L}_{N+1}$ contains 43 symbolic policies stored as workflow entries, as visualized in Figure 7. The curriculum does not enforce a uniform category distribution. Candidate generation considers the modality and capability coverage in c_t, while task selection maximizes $\mathcal{N}(\tau)\mathcal{C}(\tau)$. Capability novelty favors underexplored context–capability combinations. The competence frontier score favors intermediate estimated competence, serving as a learnability heuristic. A category may therefore be revisited when it contains novel task variants near the current competence frontier.

Localized Image Editing forms the largest category, accounting for 16.3% of the library. Reference-Guided Generation and Style Transfer & Repainting each account for 14.0%. Policies in these categories support localized control, content preservation, reference conditioning, and multi-stage workflow composition. Photorealistic Generation also accounts for 14.0%, covering diverse subjects and scenes that require high perceptual fidelity.

As reported in Table 6, removing self-directed inquiry reduces Creative Total Resolve from 95.0% to 72.5%. The corresponding T2I and I2I Resolve declines are 33.3 and 20.0 percentage points, respectively. These results support acquiring reusable symbolic policies before downstream Creative tasks arrive. Table 7 isolates the two task selection criteria. All variants retain a 100.0% Pass rate, with differences appearing in Resolve, which measures whether outputs satisfy all task requirements. Removing capability novelty lowers Creative Total Resolve to 87.5%. T2I Resolve decreases from 83.3% to 66.7%, while I2I Resolve decreases from 93.3% to 86.7%. These declines support $\mathcal{N}(\tau)$ in directing exploration toward underexplored context–capability pairs. Removing the competence frontier score lowers Creative Total Resolve to 85.0%. T2I Resolve remains at 83.3%, but I2I Resolve decreases to 80.0%. This larger I2I decline supports using $\mathcal{C}(\tau)$ to guide practice with source images according to estimated capability support. The combined objective $\mathcal{N}(\tau)\mathcal{C}(\tau)$ achieves the highest Creative Total Resolve, matches the best T2I result, and improves I2I Resolve among the

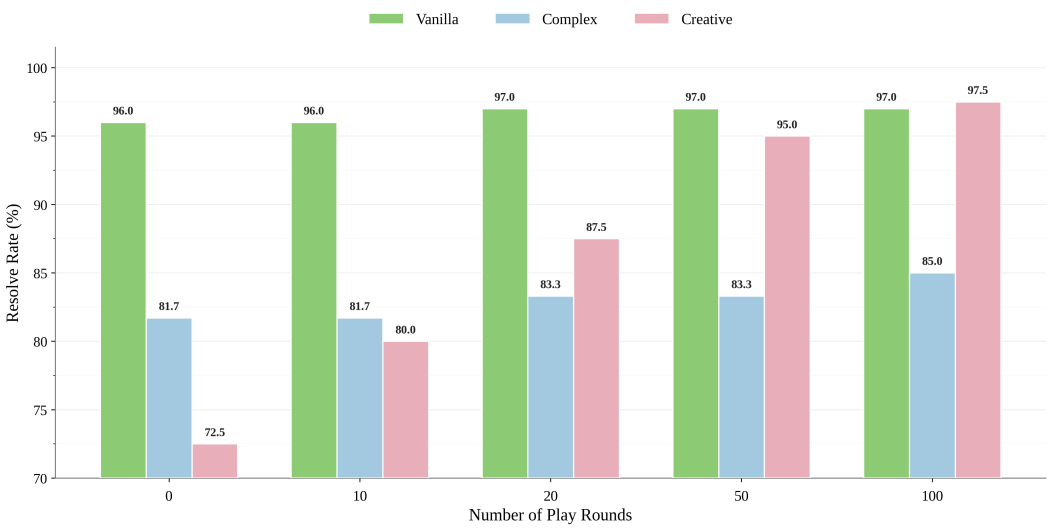


Figure 8: Resolve rates with different numbers of self-directed inquiry iterations on ComfyBench (Xue et al., 2025).

tested variants. This supports the complementary roles of novelty and estimated learnability in task selection.

As shown in Figure 8, increasing self-directed inquiry iterations improves task resolution, with the largest gains on Creative tasks. Without inquiry, OmniHarness achieves Resolve rates of 96.0%, 81.7%, and 72.5% on Vanilla, Complex, and Creative tasks, respectively, yielding 87.0% overall Resolve. Ten iterations increase Creative Resolve to 80.0%, while 20 and 50 iterations further improve it to 87.5% and 95.0%. The corresponding overall Resolve rates are 88.5%, 91.0%, and 92.5%. Vanilla reaches 97.0%, while Complex improves modestly to 83.3% after 50 iterations. These results suggest that policy acquisition before downstream tasks is particularly beneficial for creative workflow discovery and composition. Increasing inquiry from 50 to 100 iterations raises Creative Resolve to 97.5% and overall Resolve to 93.5%, adding only 1.0 percentage point overall. This diminishing performance gain supports using 50 iterations as a practical balance between task resolution and additional practice cost.

A.4.2 Fine-Grained Ablation of Feedback-Guided Execution

As shown in Table 8, the full OmniHarness achieves a 100.0% Pass rate and an 83.3% Resolve rate on the Complex subset. Removing planning causes the largest Resolve decline, reducing Complex Total Resolve from 83.3% to 55.0%. T2I and I2I Resolve fall to 40.0% and 56.5%, respectively. These results support the role of the ordered plan ρ_t in decomposing multi-step objectives, maintaining dependencies, and guiding workflow composition and synthesis. Removing intermediate verification causes the largest Pass decline, reducing Complex Total Pass from 100.0% to 75.0%, while Resolve falls to 68.3%. Without intermediate checks, incorrect outputs from $a_{t,j}$ can propagate to later steps. Removing localized recovery produces smaller but consistent declines, with Pass and Resolve decreasing to 93.3% and 76.7%, respectively. This supports targeted repair that preserves verified steps when adapting symbolic policies to downstream tasks.

A.4.3 Symbolic Policy Learning Analysis

To examine symbolic policy learning during downstream execution, we keep library updates active while OmniHarness processes ComfyBench (Xue et al., 2025). Execution starts from the policy library state acquired after 50 inquiry iterations. We track the workflow library's size, reliability, modality coverage, reuse patterns, and capability distribution during downstream execution.

As shown in Figure 9, the workflow library grows from 43 entries after inquiry to 48, 62, and 81 entries after the Vanilla, Complex, and Creative subsets, respectively. The number of Validated

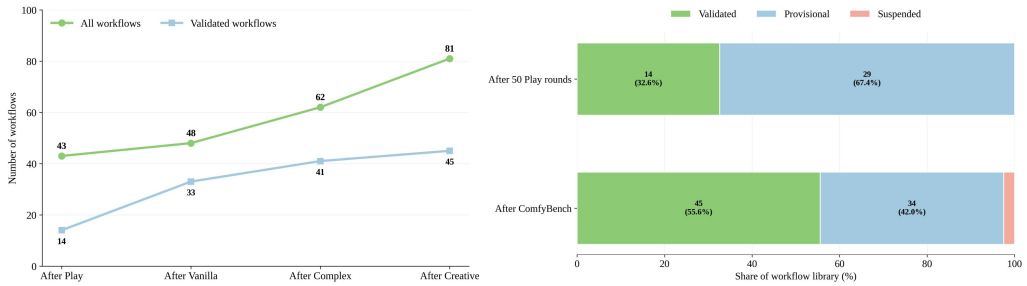


Figure 9: Symbolic policy learning during downstream execution on ComfyBench (Xue et al., 2025). (a) Growth in total and validated workflow entries in the workflow library, starting after 50 inquiry iterations and continuing through the Vanilla, Complex, and Creative subsets. (b) Changes in the proportions of *Validated*, *Provisional*, and *Suspended* workflows before and after ComfyBench execution.

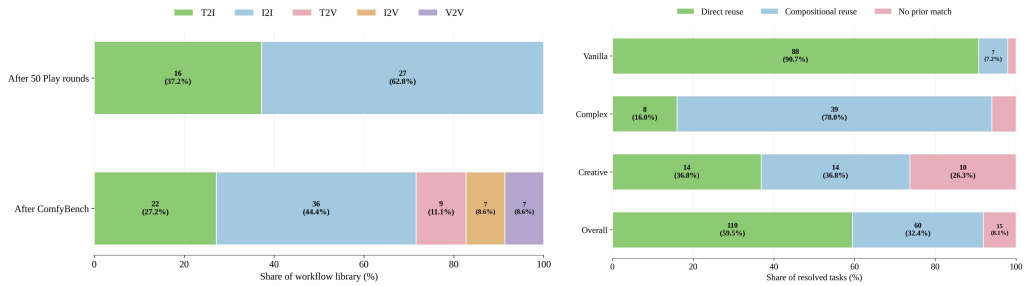


Figure 10: Evolution and reuse of symbolic policies during downstream execution on ComfyBench (Xue et al., 2025). (a) Expansion of the image-only workflow library acquired after 50 inquiry iterations to include T2I, I2I, T2V, I2V, and V2V workflows after ComfyBench execution. (b) Distribution of direct reuse, compositional reuse, and no-prior-match cases among resolved Vanilla, Complex, and Creative tasks, together with the overall distribution.

workflows follows a different trend. Vanilla adds only five workflows but increases Validated workflows from 14 to 33. This is consistent with repeated executions adding validation evidence to existing policies through updates to u_w and s_w. Complex and Creative add 14 and 19 workflows, while Validated workflows increase by only eight and four. Library expansion therefore outpaces the accumulation of sufficient validation evidence for newly acquired policies. After ComfyBench execution, the Validated proportion increases from 32.6% to 55.6%, while the Provisional proportion decreases from 67.4% to 42.0%. Two workflows are assigned the Suspended tier. These updates expand the workflow library and adjust retrieval priorities according to accumulated execution evidence.

Figure 10 shows how symbolic policies are expanded and reused. The initial library contains 16 T2I and 27 I2I workflows. After ComfyBench execution, it contains 22 T2I, 36 I2I, 9 T2V, 7 I2V, and 7 V2V workflows. The 23 newly acquired video workflows account for 28.4% of the final library. These additions extend the live workflow library from image generation to video generation during downstream execution. Among the 185 resolved tasks, 110 use direct reuse and 60 use compositional reuse, yielding a combined reuse rate of 91.9%. Reuse patterns also vary with task difficulty. Direct reuse accounts for 90.7% of resolved Vanilla tasks, while compositional reuse accounts for 78.0% of resolved Complex tasks. For Creative tasks, the no-prior-match rate rises to 26.3%. These patterns suggest that simpler tasks mainly invoke existing policies, while harder tasks more often require composition or workflow construction when no applicable policy is available.

Figure 11 presents the capability distribution shift within the image workflow library, which grows from 43 to 58 workflows. Localized Image Editing increases from 7 to 10 workflows, and Image

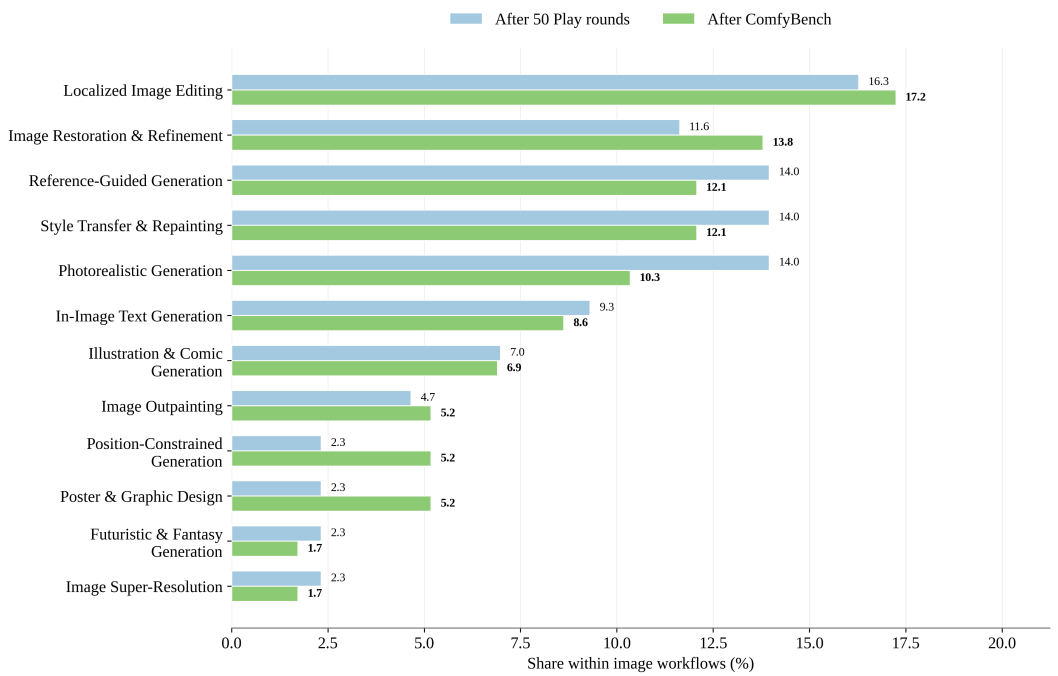


Figure 11: Capability distribution shift in the image workflow library during downstream execution on ComfyBench (Xue et al., 2025). The bars compare workflow proportions across capability categories after 50 inquiry iterations and after ComfyBench execution. Percentages are normalized over T2I and I2I workflows only. The comparison shows how the capabilities represented by stored symbolic policies change during downstream execution.

Restoration & Refinement increases from 5 to 8. Position-Constrained Generation and Poster & Graphic Design each increase from 1 to 3. These additions expand the library's representation of localized editing, restoration, spatial constraints, and layout-aware generation. Category proportions can decrease even when their workflow counts remain stable or increase. Reference-Guided Generation and Style Transfer & Repainting each grow from 6 to 7 workflows, although their shares decrease from 14.0% to 12.1%. Photorealistic Generation remains at 6 workflows, while its share decreases from 14.0% to 10.3%. The distribution therefore reflects both library growth and changes in relative capability coverage.

A.4.4 Reasoning Efficiency Analysis

As shown in Figure 12, OmniHarness reasoning cost varies with task difficulty and modality. Across all 200 ComfyBench tasks, it consumes an average of 12.03K tokens and 52.9 s of agent-side reasoning time per task. For Tasks 1–100 in the Vanilla subset, token consumption ranges from 6.74K to 9.08K, while reasoning time ranges from 30.8 s to 45.9 s. The lowest cost occurs in Tasks 1–20, which contain simple T2I requests. Cost increases for Tasks 21–60, where T2V and I2V tasks require new modality-specific workflow construction. It decreases for Tasks 61–100, which mainly contain I2I tasks suitable for reusing symbolic policies learned through self-directed inquiry.

Tasks 101–160 in the Complex subset require 14.36K–16.28K tokens and 57.9–66.8 s per task. Their dependent operations require an ordered plan, workflow composition, and intermediate verification. Tasks 121–140 have the highest cost within this subset and include many video conversion and interpolation workflows. The decrease for Tasks 141–160 coincides with their larger share of I2I tasks, for which symbolic policies learned through self-directed inquiry are available.

Tasks 161–200 in the Creative subset incur the highest reasoning cost. Average token consumption reaches 16.47K and 18.63K in the final two groups, while reasoning time increases to 69.8 s and 83.5 s. These tasks involve longer instructions, stricter content constraints, complex text and layout requirements, and identity-preserving transformations. The final group is also dominated by

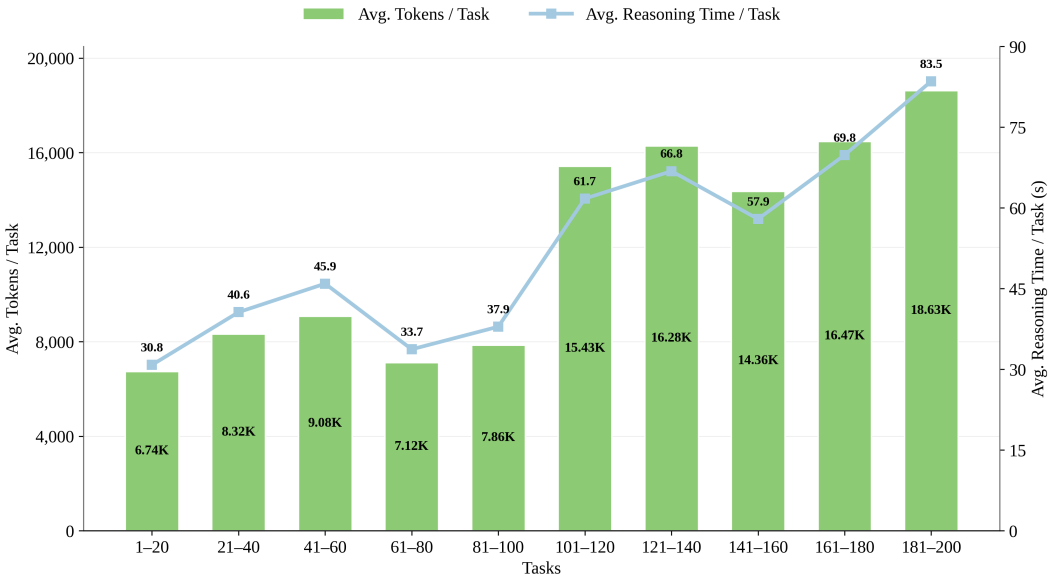


Figure 12: Reasoning efficiency of OmniHarness on the 200 ComfyBench tasks (Xue et al., 2025), reported over consecutive groups of 20 tasks. The bars show the average token consumption per task, while the line shows the average reasoning time per task. Tasks 1–100, 101–160, and 161–200 correspond to the Vanilla, Complex, and Creative subsets, respectively. Reasoning time includes agent-side planning, workflow synthesis, verification, and recovery, but excludes visual generation model execution.

Table 13: Backbone ablation of OmniHarness on ComfyBench (Xue et al., 2025). The first three rows report representative existing agents, followed by OmniHarness instantiated with different reasoning backbones and planner configurations. Pass and Resolve are reported as percentages, with higher values indicating better performance.

Agent	Vanilla		Complex		Creative		Total	
	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑
GPT-4o + ComfyAgent (Xue et al., 2025)	67.0	46.0	48.3	21.7	40.0	15.0	56.0	32.5
GPT-4o + ComfyMind (Guo et al., 2025)	**100.0**	92.0	**100.0**	85.0	**100.0**	57.5	**100.0**	83.0
Gemini-2.5-Flash + SymbOmni (Liu et al., 2026a)	**100.0**	95.0	**100.0**	83.3	**100.0**	67.5	**100.0**	86.0
Gemini-2.5-Flash + OmniHarness	**100.0**	96.0	**100.0**	81.7	**100.0**	87.5	**100.0**	90.0
GPT-4o + OmniHarness	**100.0**	95.0	**100.0**	76.7	**100.0**	95.0	**100.0**	89.5
GPT-5.5 + OmniHarness	**100.0**	97.0	**100.0**	81.7	**100.0**	**97.5**	**100.0**	92.5
Codex GPT-4o + OmniHarness	**100.0**	97.0	**100.0**	83.3	**100.0**	95.0	**100.0**	92.5
Codex GPT-5.5 + OmniHarness	**100.0**	**98.0**	**100.0**	**88.3**	**100.0**	**97.5**	**100.0**	**95.0**

video generation and V2V editing, requiring workflow composition, intermediate verification, and localized recovery. Token consumption and reasoning time follow similar trends across these task groups.

A.4.5 Backbone Ablation

As shown in Table 13, same-backbone comparisons evaluate agent frameworks while holding the underlying reasoning model fixed. With GPT-4o, OmniHarness achieves a 100.0% Total Pass rate and an 89.5% Total Resolve rate, compared with 56.0% and 32.5% for ComfyAgent, and 100.0% and 83.0% for ComfyMind. Its largest advantage over ComfyMind appears on Creative tasks, where Resolve increases from 57.5% to 95.0%, although ComfyMind performs better on Complex tasks. With Gemini-2.5-Flash, OmniHarness improves Total Resolve over SymbOmni from 86.0% to 90.0% and

Table 14: Source image pool isolation study on ComfyBench (Xue et al., 2025). $\mathcal{X}_{\text{ComfyBench}}$ denotes the original ComfyBench source image pool, while $\mathcal{X}_{\text{Gen}}$ replaces it with independently generated images from the same generative capability space.

Agent	Vanilla		Complex		Creative		Total	
	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑
OmniHarness with $\mathcal{X}_{\text{ComfyBench}}$	**100.0**	**97.0**	**100.0**	**83.3**	**100.0**	**95.0**	**100.0**	**92.5**
OmniHarness with $\mathcal{X}_{\text{Gen}}$	**100.0**	**97.0**	**100.0**	**83.3**	**100.0**	90.0	**100.0**	91.5

Creative Resolve from 67.5% to 87.5%, despite a small decrease on Complex tasks from 83.3% to 81.7%. These comparisons show performance differences across frameworks using the same reasoning model. Across OmniHarness configurations, all variants achieve a 100.0% Pass rate, while Resolve varies with the model and planner configuration. Replacing GPT-4o with GPT-5.5 increases Total Resolve from 89.5% to 92.5%. Codex GPT-4o also reaches 92.5%, outperforming GPT-5.5 on Complex tasks but trailing it on Creative tasks. Codex GPT-5.5 achieves the highest overall result of 95.0% among the tested configurations. These results support using OmniHarness across reasoning models and planner configurations. Its symbolic policies capture shared procedures for visual generation task families, while feedback-guided execution adapts them to new tasks and supplies feedback for continual policy learning.

A.4.6 Source Image Pool Isolation Study

To assess whether the benefits of inquiry depend on ComfyBench source images, we replace the original pool with 33 independently generated images from the same generative capability space. As shown in Table 14, OmniHarness with $\mathcal{X}_{\text{Gen}}$ achieves a Total Resolve rate of 91.5%, compared with 92.5% using $\mathcal{X}_{\text{ComfyBench}}$. Vanilla and Complex results remain unchanged, while Creative Resolve decreases. These results suggest that inquiry can acquire useful symbolic policies from independently generated images, with limited dependence on the original ComfyBench source pool in this evaluation.

A.5 Comprehensive Benchmark Evaluation

We evaluate OmniHarness on six complementary benchmarks covering diverse visual generation task families. ComfyBench (Xue et al., 2025) assesses autonomous ComfyUI workflow construction and execution across Vanilla, Complex, and Creative generation and editing tasks. GenEval (Ghosh et al., 2023) measures compositional text-to-image fidelity through object generation, object co-occurrence, counting, color, relative position, and attribute binding. GenEval2 (Kamath et al., 2025) provides a finer-grained assessment of object generation, attribute rendering, counting, spatial relations, and transitive verb relations. Reason-Edit (Huang et al., 2024) evaluates explicit instruction understanding and commonsense-driven image editing, emphasizing target localization and preservation of unrelated content. WISE (Niu et al., 2025) examines knowledge-guided synthesis across cultural, temporal, spatial, biological, physical, and chemical domains. KRIS-Bench (Wu et al., 2025d) evaluates factual, conceptual, and procedural knowledge in generation and editing. Together, these benchmarks assess performance across workflow construction, compositional generation, reasoning-guided editing, and knowledge-grounded visual synthesis.

A.5.1 Evaluation on ComfyBench

Figures 13 and 4 present qualitative comparisons on representative ComfyBench tasks. Figure 13 compares localized I2I editing using the same input image. In these examples, OmniHarness removes the plate, fork, or glass as instructed and replaces the fork with a spoon while preserving unrelated objects and the scene layout. Some compared outputs retain the target object, modify non-target content, or change the overall appearance. Figure 4 illustrates multi-stage and appearance-sensitive tasks. OmniHarness reconstructs a scribble as a realistic red flower and transfers its visual style to a new flower-field image. It also restores and colorizes an old photograph while preserving

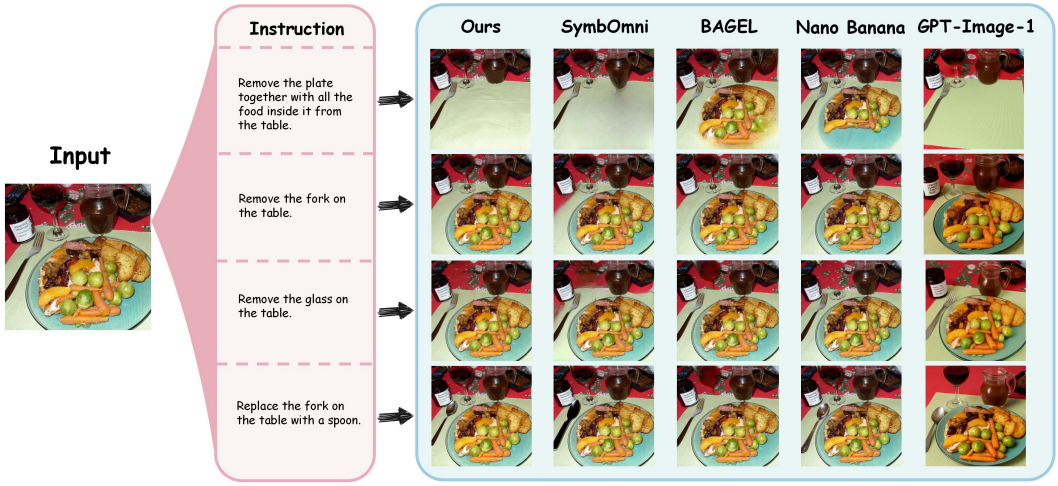


Figure 13: Qualitative comparison on representative I2I tasks from ComfyBench (Xue et al., 2025). The examples illustrate target localization, instruction following, and preservation of unrelated scene content.

identity and clothing details. In facial appearance editing, it changes the requested attributes while retaining facial structure, pose, and clothing.

As shown in Table 1, both OmniHarness variants achieve a 100.0% Pass rate across all task categories. Codex GPT-4o + OmniHarness obtains a Total Resolve rate of 92.5%, exceeding SymbOmni and ComfyMind by 6.5 and 9.5 percentage points, respectively. Its largest advantage appears on Creative tasks, where Resolve reaches 95.0%, compared with 67.5% for SymbOmni and 57.5% for ComfyMind. The ablation results in Tables 6 and 7 further support the contribution of self-directed inquiry and its task selection criteria to Creative performance. Before downstream objectives are specified, OmniHarness selects practice tasks using capability novelty $\mathcal{N}(\tau)$ and the competence frontier score $\mathcal{C}(\tau)$. Their product favors underexplored context–capability pairs near the estimated competence frontier. Verified executions are abstracted into symbolic policies that capture shared procedures and applicability conditions for visual generation task families. The harness instantiates, adapts, and composes these policies for downstream requirements.

A.5.2 Evaluation on GenEval

As shown in Table 2 and Figure 14, OmniHarness achieves an overall GenEval score of 0.997, exceeding SymbOmni, ComfyMind, and GPT-Image-1 by 0.017, 0.097, and 0.157, respectively. It reaches 1.00 on single-object generation, two-object co-occurrence, counting, color, and relative position, and 0.98 on attribute binding. Compared with SymbOmni, Position improves from 0.97 to 1.00 and Attribute Binding from 0.95 to 0.98. GPT-Image-1 obtains 0.75 and 0.61 on these categories. The qualitative examples illustrate these compositional requirements. OmniHarness generates a clear parking meter, preserves bottle–refrigerator co-occurrence, produces exactly two bears, renders the requested red zebra, places the bird below the skateboard, and binds the purple and brown attributes to the elephant and sports ball. Some compared outputs violate counts, spatial relations, or attribute assignments. These results indicate high compositional fidelity across the evaluated dimensions. OmniHarness uses feedback-guided execution to adapt and compose symbolic policies for task requirements, with intermediate verification guiding workflow refinement.

A.5.3 Evaluation on GenEval2

As shown in Table 3 and Figure 15, OmniHarness achieves the best results on four of the five GenEval2 skills. It obtains 94.0 on Attribute and Count, 76.9 on Position, and 89.0 on Verb, exceeding the strongest competing results by 2.6, 19.2, 6.7, and 2.3 points, respectively. Compared with SymbOmni, the gains are 10.4 points on Attribute, 19.2 on Count, 8.1 on Position, and 24.5 on Verb. Its Object score of 95.0 matches SymbOmni but remains below Qwen-Image and Gemini

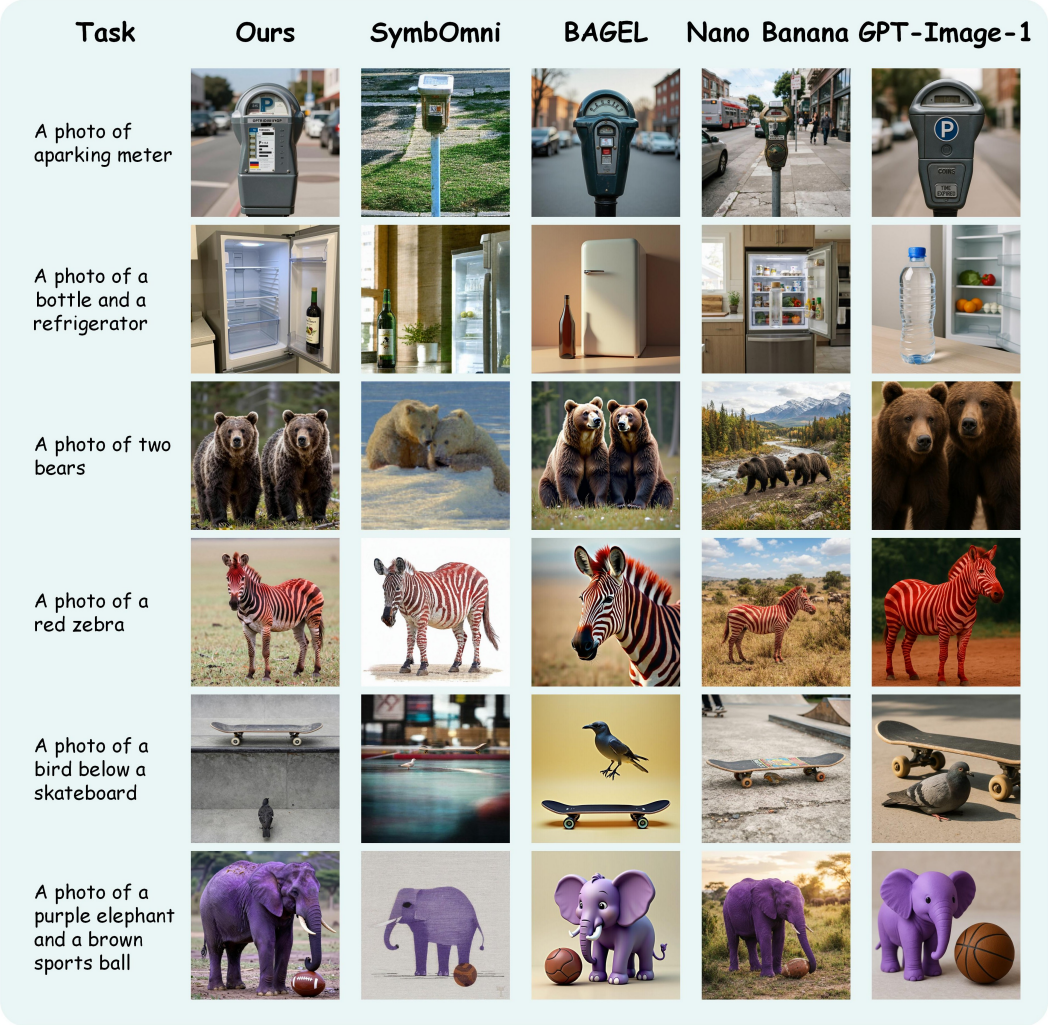


Figure 14: Qualitative comparison on representative GenEval (Ghosh et al., 2023) tasks covering single-object generation, two-object co-occurrence, counting, color, spatial relations, and attribute binding.

2.5 Flash Image. The strongest advantages therefore appear on tasks with interacting compositional constraints. In the presented examples, OmniHarness generates six cars with one kangaroo, a glass lion with one bear and four kangaroos, five checkered umbrellas with a black candle, a lion jumping over a sparkling dog, and five penguins to the left of a striped violin. Some competing outputs violate counts, assign attributes to the wrong objects, or miss spatial and action relations. These results support strong performance on counting, attributes, spatial relations, and transitive verb relations, while isolated object generation remains less competitive.

A.5.4 Evaluation on Reason-Edit

As shown in Table 5, Figure 16, and Figure 6, OmniHarness performs well on instruction understanding and reasoning-guided editing. In the Understanding Scenarios, it leads all four metrics, with PSNR of 23.894 dB, SSIM of 0.856, LPIPS of 0.053, and CLIP Score of 24.554. Compared with the strongest baseline for each metric, PSNR improves by 0.298 dB, SSIM by 0.105, and CLIP Score by 0.820, while LPIPS decreases by 0.014. In the Reasoning Scenarios, OmniHarness achieves the highest SSIM of 0.796 and CLIP Score of 21.318. Its LPIPS is 0.052, compared with the best result of 0.049, and its PSNR remains below SmartEdit-13B and InsightEdit. The GPT-score comparison provides another assessment. OmniHarness ranks first on Left-Right, Mirror, Color,

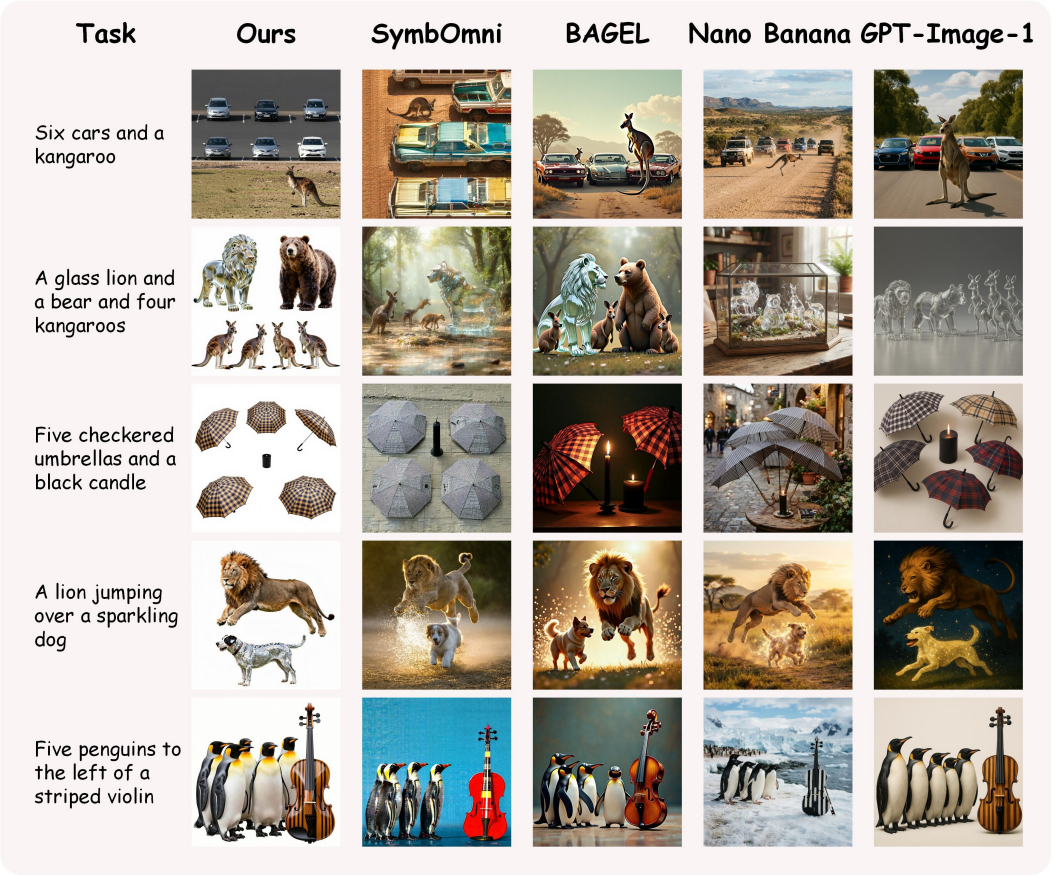


Figure 15: Qualitative comparison on representative GenEval2 (Kamath et al., 2025) tasks covering object generation, attribute rendering, counting, spatial relations, and transitive verb relations.

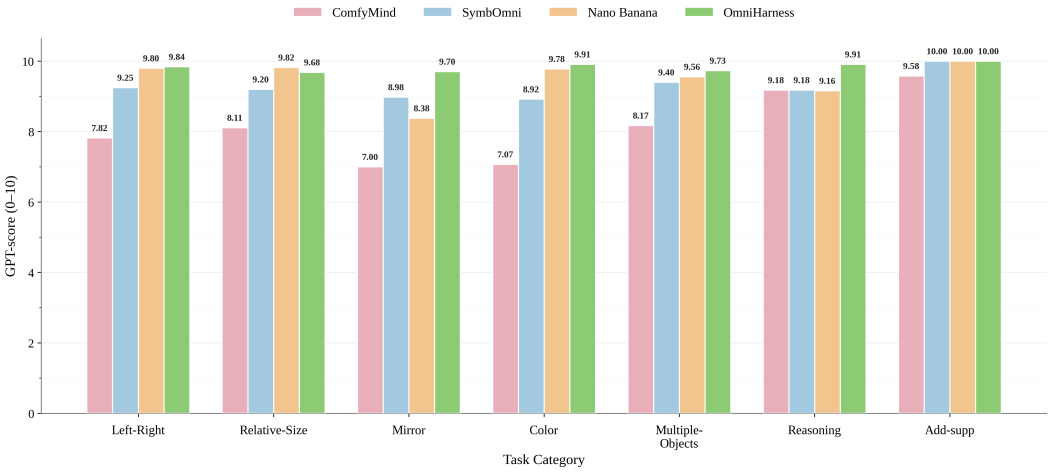


Figure 16: GPT-score comparison on Reason-Edit (Huang et al., 2024) across seven reasoning-intensive image-editing categories. Higher scores indicate better performance.

Multiple-Objects, and Reasoning, reaches 10.00 on Add-supp, and ranks second to Nano Banana on Relative-Size. The Understanding examples illustrate target identification through spatial position, color, mirror relations, and relative size. OmniHarness replaces or adds the specified object while



Figure 17: Qualitative comparison on representative WISE (Niu et al., 2025) tasks spanning cultural, temporal, spatial, biological, physical, and chemical knowledge.

preserving other instances and scene structure. In the Reasoning examples, it identifies implicit targets from commonsense descriptions, including a dog bowl, a fruit-cutting knife, vitamin-rich food, an animal lying on grass, a game ball, and a warning sign. It then performs the requested removal or replacement with limited changes to unrelated regions. These results support the effectiveness of OmniHarness on both explicit and reasoning-intensive editing instructions.

A.5.5 Evaluation on WISE

As shown in Table 4 and Figure 17, OmniHarness achieves the highest overall WiScore of 0.86, exceeding SymbOmni and GPT-Image-1 by 0.06 and ComfyMind by 0.10. It ranks first in Time, Biology, Physics, and Chemistry, with scores of 0.85, 0.84, 0.82, and 0.86, respectively. Compared with the strongest competing result in each category, the gains are 0.14, 0.01, 0.03, and 0.08, respectively. Its Cultural and Space scores reach 0.88 and 0.86, remaining 0.02 and 0.03 below the corresponding best results. The qualitative examples illustrate the range of knowledge requirements. OmniHarness generates Russian nesting dolls as the requested cultural craft, depicts the Sydney Opera House at night, and preserves the depth relation between a nearby coin and a distant cityscape. It also depicts the formation of migrating geese, condensation on a cold can in humid air, and the precipitate formed by mixing copper sulfate and sodium hydroxide. Some compared outputs capture the main objects but miss the implied temporal condition, spatial relation, physical state, or

Table 15: Quantitative comparison on KRIS-Bench (Wu et al., 2025d). Scores evaluate factual, conceptual, and procedural knowledge, together with overall performance. Higher values indicate better performance.

Method	Factual↑	Conceptual↑	Procedural↑	Overall↑
Closed-Source Models				
GPT-Image-1	**79.80**	81.37	**78.32**	**80.09**
Gemini 2.0 Flash Experimental	65.26	59.65	62.90	62.41
Doubao	63.30	62.23	54.17	60.70
Open-Source Models				
BAGEL-Think (Deng et al., 2025)	55.77	59.44	39.26	53.36
BAGEL (Deng et al., 2025)	47.71	52.17	40.23	47.76
Step1X-Edit (Liu et al., 2025b)	45.52	48.01	31.82	43.29
Emu2 (Sun et al., 2024b)	45.40	37.54	34.91	39.70
AnyEdit (Yu et al., 2025)	39.26	41.88	31.74	38.55
MagicBrush (Zhang et al., 2023)	41.84	39.24	26.54	37.15
OmniGen (Xiao et al., 2025)	33.11	28.02	23.89	28.85
InsPix2Pix (Brooks et al., 2023)	23.33	25.59	17.28	22.82
Collaborative AI Systems				
SymbOmni (Liu et al., 2026a)	73.33	72.28	70.29	72.18
OmniHarness	74.81	**81.89**	73.22	77.33

Table 16: Quantitative plug-and-play evaluation on ComfyBench (Xue et al., 2025). The first three rows report the original host agents, while the last three augment them with the same frozen policy snapshot $\mathcal{K}_{\text{inquiry}}$ learned through self-directed inquiry, without model fine-tuning.

Agent	Vanilla		Complex		Creative		Total	
	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑
ComfyAgent (Xue et al., 2025)	67.0	46.0	48.3	21.7	40.0	15.0	56.0	32.5
ComfyMind (Guo et al., 2025)	**100.0**	92.0	**100.0**	85.0	**100.0**	57.5	**100.0**	83.0
SymbOmni (Liu et al., 2026a)	**100.0**	95.0	**100.0**	83.3	**100.0**	67.5	**100.0**	86.0
ComfyAgent (Xue et al., 2025) + $\mathcal{K}_{\text{inquiry}}$	87.0	69.0	73.3	46.7	67.5	42.5	79.0	57.0
ComfyMind (Guo et al., 2025) + $\mathcal{K}_{\text{inquiry}}$	**100.0**	94.0	**100.0**	**86.7**	**100.0**	75.0	**100.0**	88.0
SymbOmni (Liu et al., 2026a) + $\mathcal{K}_{\text{inquiry}}$	**100.0**	**96.0**	**100.0**	85.0	**100.0**	**77.5**	**100.0**	**89.0**

scientific consequence. These results support the effectiveness of OmniHarness across the evaluated cultural, spatiotemporal, and scientific generation tasks.

A.5.6 Evaluation on KRIS-Bench

As shown in Table 15 and Figure 18, OmniHarness achieves an overall score of 77.33 on KRIS-Bench (Wu et al., 2025d), exceeding SymbOmni by 5.15 points and BAGEL-Think, the strongest open-source baseline overall, by 23.97 points. It trails GPT-Image-1 by 2.76 points. OmniHarness achieves the best Conceptual score of 81.89, exceeding GPT-Image-1 and SymbOmni by 0.52 and 9.61 points, respectively. Its Factual and Procedural scores reach 74.81 and 73.22, improving over SymbOmni by 1.48 and 2.93 points. The qualitative examples illustrate diverse knowledge requirements. OmniHarness completes numerical patterns, applies scientific, geographic, and biological knowledge, reasons about physical processes, performs landmark and color transformations, composes reference images, annotates geometric and medical content, controls object counts, completes missing structures, and follows poster-design constraints. These tasks require knowledge to be translated into visual operations under specific constraints. The harness instantiates, adapts, and composes symbolic policies to meet these requirements. Intermediate verification guides workflow refinement and localized recovery during execution.

Table 17: Quantitative cross-modal generalization results on the video-generation subset of ComfyBench (Xue et al., 2025). T2V, I2V, and V2V denote text-to-video, image-to-video, and video-to-video tasks, respectively. GPT-4o + OmniHarness uses the frozen policy snapshot $\mathcal{K}_{\text{inquiry}}$, learned through self-directed inquiry in an image-only capability space without exposure to video tasks during inquiry.

Agent	T2V		I2V		V2V		Total	
	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑	Pass↑	Res.↑
GPT-4o + ComfyAgent (Xue et al., 2025)	68.4	47.4	48.0	20.0	40.0	13.3	56.4	32.1
GPT-4o + ComfyMind (Guo et al., 2025)	**100.0**	71.1	**100.0**	76.0	**100.0**	53.3	**100.0**	69.2
Gemini-2.5-Flash + SymbOmni (Liu et al., 2026a)	**100.0**	81.6	**100.0**	88.0	**100.0**	66.7	**100.0**	80.8
GPT-4o + OmniHarness	**100.0**	**84.2**	**100.0**	**92.0**	**100.0**	**80.0**	**100.0**	**85.9**

A.6 Detailed Plug-and-Play Analysis

To evaluate policy portability, we keep the snapshot $\mathcal{K}_{\text{inquiry}}$ learned through self-directed inquiry frozen and map its entries to each host agent's native knowledge interface. The snapshot contains reusable workflow templates, failure evidence, and corrective strategies from the workflow and failure libraries. Host model weights and original control flow remain unchanged. Host agents retrieve, instantiate, and adapt the exported policies through their existing execution procedures. For ComfyAgent (Xue et al., 2025), each workflow template is converted into its native Python-like representation and inserted into the *Reference* memory with its name, description, preconditions, expected effects, and dependencies. The RetrieveAgent selects relevant entries, while the CombineAgent and AdaptAgent compose or modify the retrieved workflows. The RefineAgent checks the adapted workflow before it enters the *Workspace*. Failure evidence and corrective strategies provide correction references for the PlanAgent and RefineAgent. For ComfyMind (Guo et al., 2025), each compatible workflow entry is wrapped as a Semantic Workflow Interface module. Its description, preconditions, and expected effects define the semantic function interface, while its ComfyUI graph template provides the canonical JSON implementation. The Planning Agent selects relevant modules, and the Execution Agent supplies task-specific parameters before execution. Corrective strategies are appended to the local-feedback context as correction guidance. For SymbOmni (Liu et al., 2026a), each workflow entry is converted into a Symbolic Concept. Its description and preconditions define the concept semantics, its reusable procedure defines the Symbolic Workflow Instruction, its dependencies and parameters define the parameter configuration, and its reliability statistics define the concept score. Failure library entries become negative or refinement concepts. The resulting concepts are inserted into the Symbolic Concept Box and accessed through the original retrieval and transduction process. The exported snapshot receives no updates during evaluation.

As shown in Table 16, the same $\mathcal{K}_{\text{inquiry}}$ improves task resolution in all three host agents without model fine-tuning. For ComfyAgent, Total Pass increases from 56.0% to 79.0%, while Total Resolve increases from 32.5% to 57.0%. Creative Resolve rises from 15.0% to 42.5%, indicating gains in workflow executability and task completion. ComfyMind and SymbOmni retain their 100.0% Pass rates, with improvements appearing in Resolve. ComfyMind increases Total Resolve from 83.0% to 88.0% and Creative Resolve from 57.5% to 75.0%. SymbOmni increases Total Resolve from 86.0% to 89.0% and Creative Resolve from 67.5% to 77.5%. The larger Creative gains suggest that the exported symbolic policies are particularly useful for these tasks. Improvements through code references, semantic workflow modules, and symbolic concepts support plug-and-play reuse across the three evaluated visual agent frameworks.

A.7 Cross-Modal Generalization Evaluation

As shown in Table 17 and Figures 19 and 20, GPT-4o + OmniHarness applies the frozen policy snapshot $\mathcal{K}_{\text{inquiry}}$, learned through image-only inquiry, to video generation tasks not encountered during self-directed inquiry. The snapshot remains unchanged during evaluation. OmniHarness achieves a 100.0% Pass rate across T2V, I2V, and V2V, with Resolve rates of 84.2%, 92.0%, and 80.0%, respectively. Overall Resolve reaches 85.9%, exceeding SymbOmni by 5.1 percentage points. The gains are 2.6, 4.0, and 13.3 percentage points on T2V, I2V, and V2V, with the largest improvement on V2V

tasks combining temporal processing and content-preserving editing. The sampled frames illustrate T2V generation, source-conditioned I2V generation, and motion-transfer and material-editing tasks in V2V. Invocation statistics document the reuse of image-domain symbolic policies within video workflows. Across all video tasks, OmniHarness invokes image workflows 90 times, including 48 calls for T2V, 27 for I2V, and 15 for V2V. Photorealistic Generation and Reference-Guided Generation contribute 33 and 28 calls, while Localized Image Editing and Image Super-Resolution contribute 10 and 7 calls. These policies support visual content construction, reference preservation, local editing, and quality enhancement, while video-specific components handle temporal operations. The results support reuse beyond image generation through task-specific adaptation and composition of symbolic policies with video-specific components, while the exported snapshot remains fixed.

A.8 User Study

We conduct a blind user study on 40 tasks selected from ComfyBench (Xue et al., 2025). A total of 36 participants evaluate anonymized outputs from OmniHarness, SymbOmni, BAGEL, and GPT-Image-1. As shown in Figure 21, each questionnaire presents the task instruction, the input image when required, and one anonymized generation result. Participants rate each result on a five-point Likert scale from 1 (*Strongly disagree*) to 5 (*Strongly agree*) across Task Description Alignment, Input-Structure Fidelity, Photorealism, Visual Quality, and Overall Satisfaction. Figure 22 reports mean ratings over tasks and participants. OmniHarness achieves the highest ratings for Task Description Alignment, Input-Structure Fidelity, and Overall Satisfaction, reaching 4.72, 4.63, and 4.62, respectively. GPT-Image-1 receives higher Photorealism and Visual Quality ratings of 4.68 and 4.64, compared with 4.50 and 4.56 for OmniHarness. Averaged across all five criteria, OmniHarness obtains the highest score of 4.61, followed by GPT-Image-1, SymbOmni, and BAGEL at 4.43, 4.35, and 3.89, respectively. These results indicate favorable user assessments of OmniHarness, particularly for instruction alignment, input-structure preservation, and overall satisfaction.

A.9 Limitations

OmniHarness combines symbolic policy learning, feedback-guided execution, and self-directed inquiry while keeping model parameters fixed. Verified executions yield symbolic policies for visual generation task families, which are adapted and refined during downstream execution. However, both self-directed inquiry and downstream execution incur additional computation. Self-directed inquiry requires repeated task generation, workflow construction, execution, verification, and library consolidation. Downstream tasks may trigger multiple planning, verification, and recovery cycles as the harness adapts and composes policies for new requirements. These operations increase agent-side token consumption and reasoning time, especially for complex tasks. Future work should explore lighter execution frameworks, selective verification, adaptive recovery, efficient policy retrieval and composition, and reasoning budgets that adapt to task difficulty.

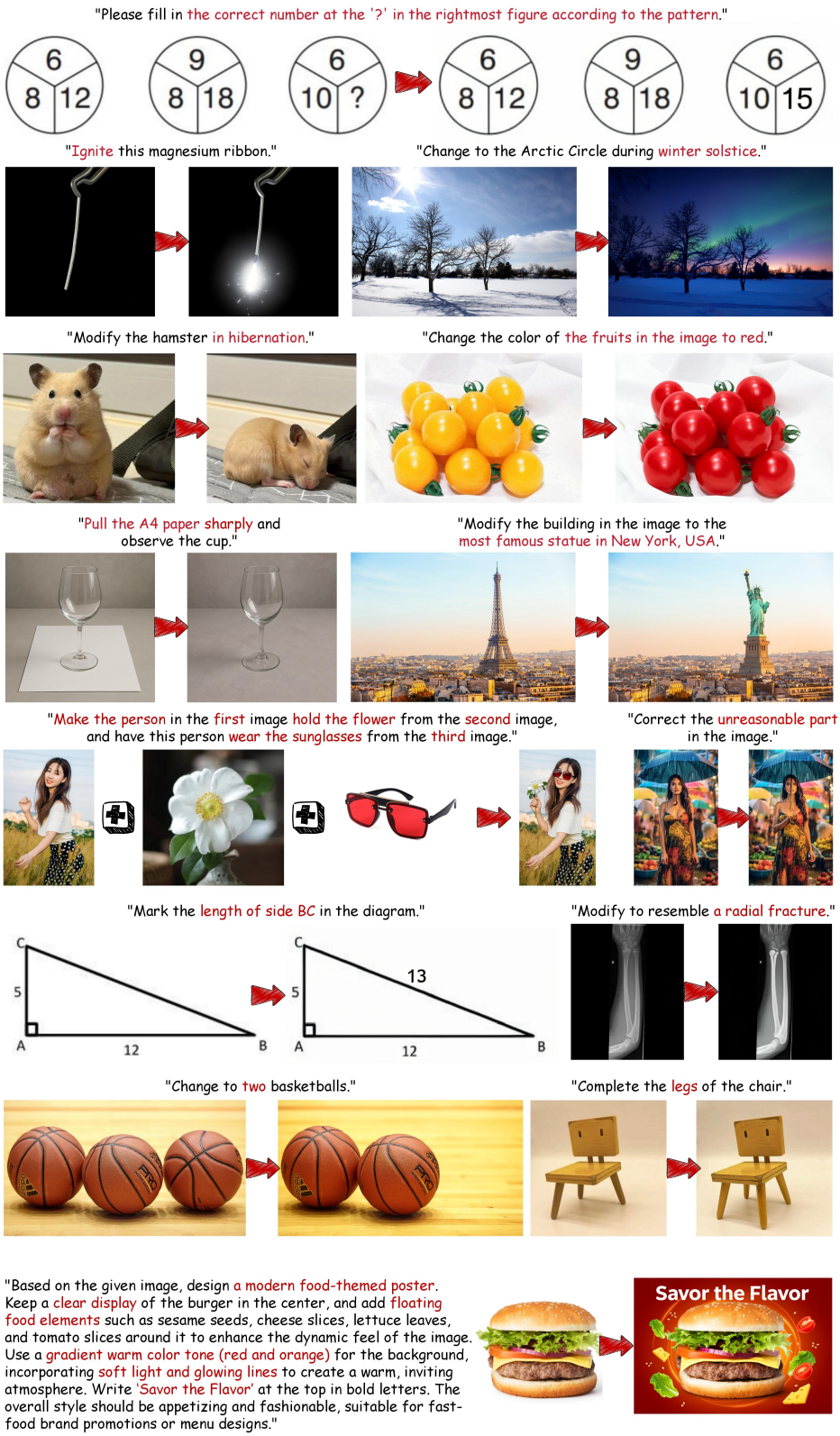


Figure 18: Qualitative results of OmniHarness on representative KRIS-Bench tasks (Wu et al., 2025d), covering factual, conceptual, and procedural knowledge through knowledge-guided generation, editing, visual reasoning, and multi-image composition.

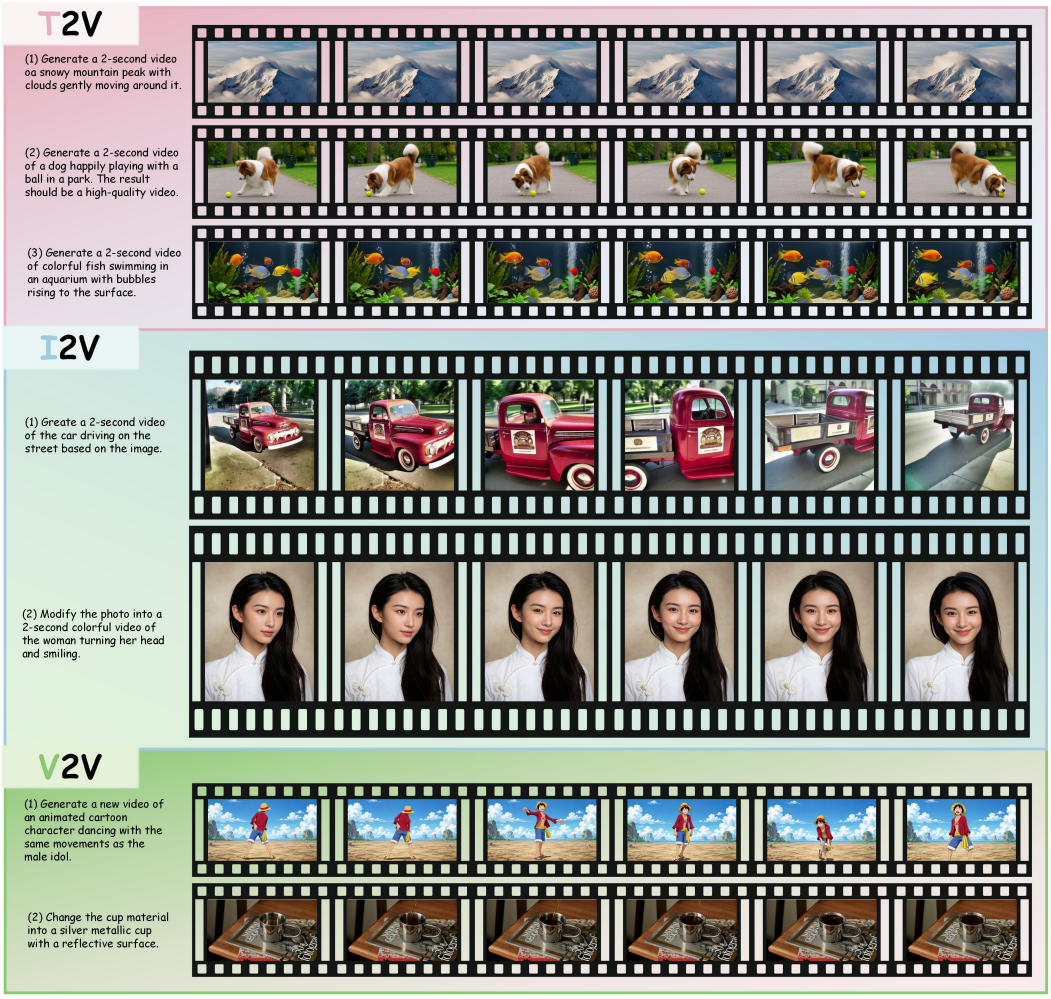


Figure 19: Qualitative cross-modal generalization results of OmniHarness on representative video-generation tasks from ComfyBench (Xue et al., 2025). Sampled frames illustrate generated content and source-content preservation in T2V, I2V, and V2V tasks using the frozen policy snapshot $\mathcal{K}_{\text{inquiry}}$ learned through image-only inquiry.

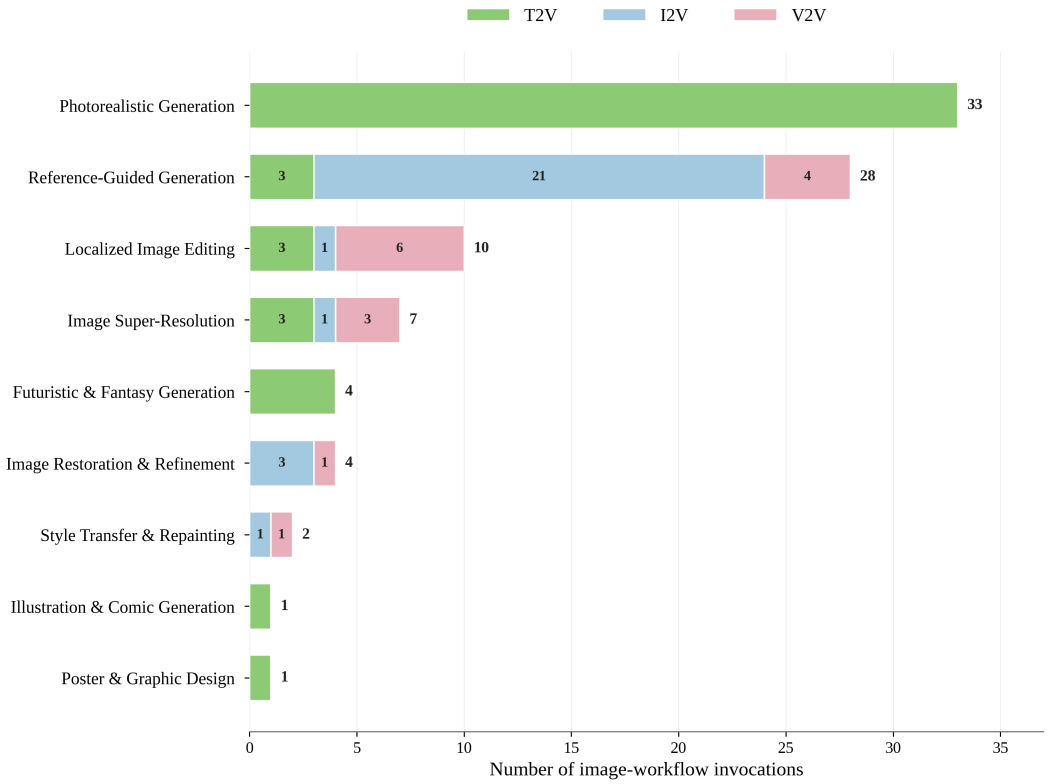


Figure 20: Reuse of symbolic policies learned through image-only inquiry during cross-modal generalization on ComfyBench (Xue et al., 2025). The bars report invocation counts of workflow entries from the frozen image-only workflow library for T2V, I2V, and V2V tasks.

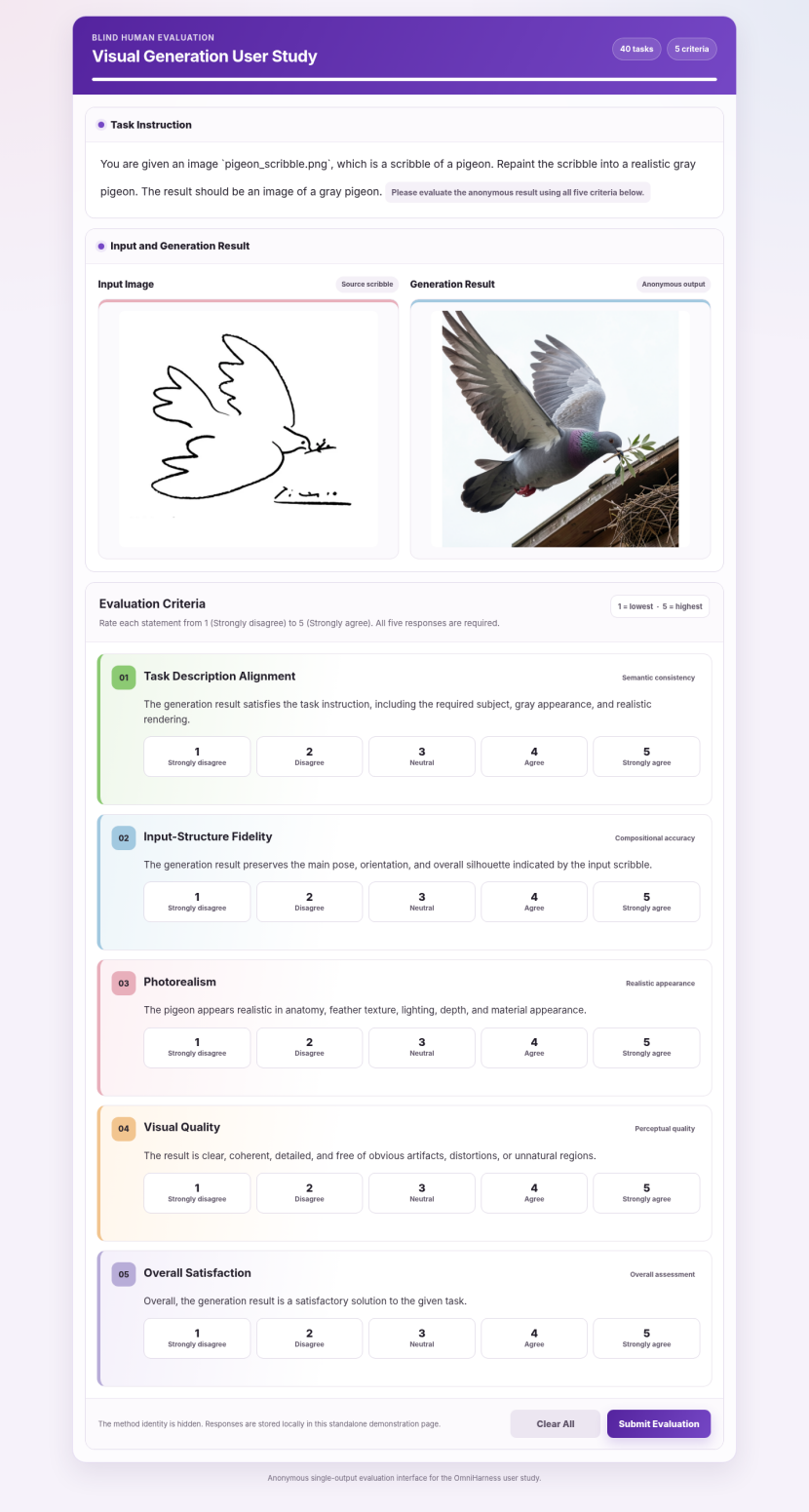


Figure 21: Web interface used in the 40-task user study. Participants assess an anonymized generation result against the task instruction and, when required, the input image. Ratings cover task description alignment, input-structure fidelity, photorealism, visual quality, and overall satisfaction on a five-point Likert scale.

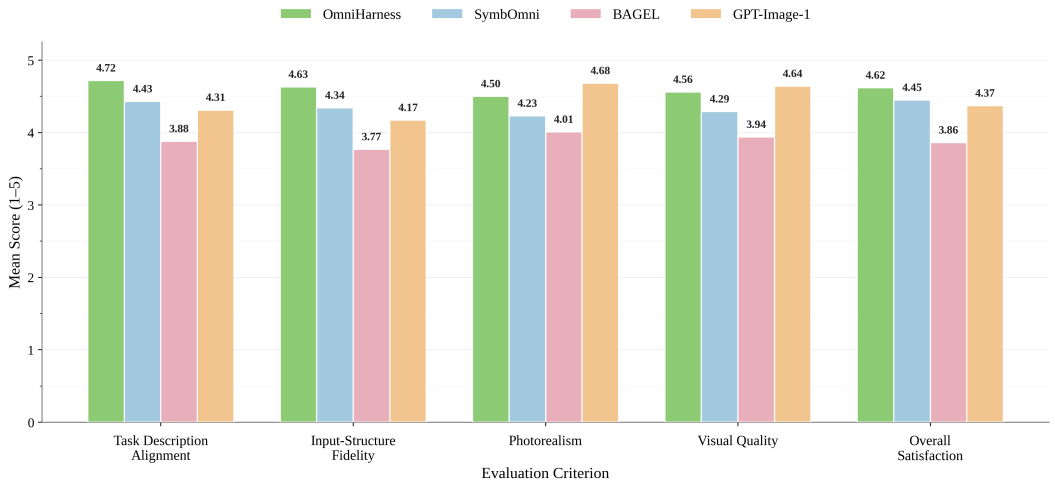


Figure 22: User-study comparison across five evaluation criteria. Mean ratings over 40 Comfy-Bench (Xue et al., 2025) tasks are reported on a five-point Likert scale, with higher scores indicating better performance.

A.10 SYSTEM PROMPTS

The following tables reproduce the two developer instructions and eight task-input templates used by the implementation. Braced Python expressions denote values inserted at runtime rather than literal prompt text. Candidate validation and scoring, context retrieval, structural plan and workflow checks, compilation, execution, evidence aggregation, symbolic policy input abstraction, library updates, consolidation, and snapshot export are implemented programmatically without separate model prompts. Candidate generation reuses a proposer thread across inquiry iterations. Execution roles share a task-scoped thread, while localized repair uses a fresh thread with the same generation-team developer instructions.

Table 18: Shared developer instructions for self-directed inquiry.

Self-Directed Inquiry Task Proposer
You are the Self-Directed Inquiry Task Proposer inside OmniHarness. Generate practice tasks before downstream objectives are specified. Each task must use the supplied capability space, be feasible with current resources, differ from recent tasks, and be independently verifiable. Use failure evidence and corrective strategies to avoid known antipatterns. For I2I, use exactly one supplied image ID and state preservation constraints. List applicable_workflow_ids only when the workflow's preconditions hold for the proposed task and source image. Exclude suspended workflows; return an empty list when none apply. Use only the supplied context and library. Do not inspect downstream task instructions, reference workflows, target outputs, or benchmark annotations. Return only the requested JSON object.

Table 19: Task-input template for candidate task generation.

Candidate Task Generation
Generate exactly {`candidate_count`} candidate tasks. Return {"candidates": [CandidateTask, ...]}. Each CandidateTask contains task_id, description, modality (T2I or I2I), capability_categories (one to three exact names), source_image_id (null for T2I), generation_constraints, preservation_constraints, success_criteria, required_resources, task_signature, proposal_rationale, safety_flags (empty for acceptable tasks), and applicable_workflow_ids. **State:** {`json.dumps(compact, ensure_ascii=False, indent=2)`}

Table 20: Shared developer instructions for feedback-guided execution.

OmniHarness Generation Team
You are the OmniHarness Generation Team. Operate through explicit roles: Planner, Plan Verifier, Workflow Writer, Executor observer, Step Verifier, Goal Verifier, and Failure Diagnoser. Retrieved workflows are symbolic policy templates for task families; retrieved failures contain evidence and corrective strategies. Bind the current task's inputs and adapt or compose templates before execution. Respect preconditions, dependencies and component-only retrieval roles. Image components require video-specific processing for video outputs. Return only the requested JSON. Never edit files, execute ComfyUI, change retry budgets, update memory, or declare technical success. Code-as-Policy is a safe DSL with one assignment and one allow-listed ComfyUI node call per line, keyword arguments only, literals or earlier variables as inputs, and no imports, attributes, control flow, nested calls, or arbitrary Python.

Table 21: Task-input template for planning.

Planner

Create a dependency-consistent ordered plan with explicit verification for every step. For each selected workflow, check its input roles, preconditions, and dependencies against the request and planned intermediate inputs. In task_analysis.policy_applicability return {workflow_id:{satisfied:true,evidence:[specific condition checks]}}. Select a policy only when its conditions can be met by the plan. Choose reuse, adapt, compose, or build. Return {plan_id, task_analysis, strategy, selected_workflow_ids, steps:[{step_id, objective, inputs, outputs, required_nodes, verification:{type,criteria}, expected_output_nodes, fallback}], rationale}.

Request:

{_json(request)}

Context:

{_json(context.compact(include_prompts=False))}

Table 22: Conditional task-input template for plan repair.

Plan Repair

Repair this plan and return the same schema.

Request:

{_json(request)}

Plan:

{_json(plan)}

Issues:

{_json(issues)}

Context:

{_json(context.compact(include_prompts=False))}

Table 23: Task-input template for workflow writing and policy adaptation.

Workflow Writer

Convert the verified plan to Code-as-Policy. Use $SOURCE_MEDIA_0 placeholders for media inputs; $SOURCE_IMAGE_0 remains accepted for compatibility. Retrieved graphs are templates: replace every $POLICY_* placeholder with task-specific text, numeric parameters, output names, or $SOURCE_MEDIA_n bindings. Check input-role preconditions and compose component_only image policies with video-specific nodes. Set _meta_title on every Code-as-Policy node to its owning plan step_id. Preserve existing assignment variable names during localized repairs. Each step must emit an inspectable intermediate output before dependent steps execute; expected_output_nodes should identify actual numeric node IDs of those output nodes, matching their one-based Code-as-Policy statement positions. A step title refers only to that step's output nodes; it does not require observing every internal node. Every node must contribute to at least one declared step output. If no applicable policy is available, construct a workflow from node knowledge. Describe the reusable task-family procedure, not the particular objects, prompt, or source filename of this execution, in workflow_name, description, preconditions and expected_effects. Return {ir_code, workflow_name, description, preconditions, expected_effects, dependencies:{Nodes,Models,Parameters}}.

Request:

{_json(request)}

Plan:

{_json(plan)}

Context:

{_json(context.compact(include_prompts=True))}

Table 24: Conditional task-input template for step verification.

Step Verifier

Inspect the supplied step output artifacts using image or video tools. Verify each listed criterion against these artifacts and the request. Do not infer an intermediate effect from the final result. Return {criteria:[{criterion,status:pass|fail|uncertain,evidence:[string]}]}. Copy each criterion exactly and use uncertain if its evidence cannot be inspected.

Request:
{_json(request)}

Step:
{_json(step)}

Step outputs:
{_json([item.path for item in result.outputs])}

Table 25: Conditional task-input template for goal verification.

Goal Verifier

Inspect every output path with the appropriate image or video tools. Evaluate every success and preservation criterion. Use uncertain when evidence is insufficient. Return {status:pass|fail|uncertain, verifier, scores, evidence, failed_criteria}.

Request:
{_json(request)}

Plan:
{_json(plan)}

Outputs:
{_json([item.path for item in result.outputs])}

Table 26: Conditional task-input template for failure diagnosis.

Failure Diagnoser

Identify the smallest failed stage. Return {failure_class:planning_failure|compilation_failure|infrastructure_failure|execution_failure|verifier_failure|semantic_failure, failed_stage, root_cause, workflow_antipattern, remedy, applicable_scope, retry_scope}.

Request:
{_json(request)}

Plan:
{_json(plan)}

Draft:
{_json(draft)}

Error:
{_json(error)}

Verification:
{_json(verification)}

Table 27: Conditional task-input template for localized repair.

Failure Subagent

Act as an isolated Failure Subagent. Repair only the diagnosed step and preserve unrelated workflow logic. Preserve the assignment variable names and node inputs outside that step and its dependent downstream nodes. Return {failed_step_id, root_cause, subworkflow_ir, revised_ir_code, reusable_name, rationale}. The full revised_ir_code must integrate the localized component.

Request:
{_json(request)}

Plan:
{_json(plan)}

Draft:
{_json(draft)}

Diagnosis:
{_json(diagnosis)}

Recovery context:
{_json(context.compact(include_prompts=True))}