Open Evaluation Agent: Efficient and Promptable Evaluation of Visual Generative Models

Shulin Tian, Ziqi Huang, Fan Zhang, Hongyuan Zhu, Yu Qiao, Ziwei Liu ✉

Abstract—Recent advancements in visual generative models have enabled high-quality image and video generation, opening diverse applications. However, evaluating these models often demands sampling hundreds or thousands of images or videos, making the process computationally expensive, especially for diffusion-based models with inherently slow sampling. Moreover, existing evaluation methods rely on rigid pipelines that overlook specific user needs and provide numerical results without clear explanations. In contrast, humans can quickly form impressions of a model's capabilities by observing only a few samples. To mimic this, we propose the **Evaluation Agent** framework, which employs human-like strategies for efficient, dynamic, multi-round evaluations using only a few samples per round, while offering detailed, user-tailored analyses. Given a natural-language evaluation request, the agent decomposes it into sub-aspects, generates targeted prompts, samples images or videos from the evaluated model, invokes suitable evaluation tools, and iteratively updates its plan from the observed evidence. This tool-grounded loop enables the framework to evaluate both predefined benchmark dimensions and open-ended user concerns without requiring a fixed prompt set or a single handcrafted metric. It offers four key advantages: *1)* **efficiency**, *2)* **promptable evaluation** tailored to diverse user needs, *3)* **explainability** beyond single numerical scores, and *4)* **scalability** across various models and tools. Experiments show that Evaluation Agent reduces evaluation time to 10% of traditional methods while delivering comparable results. We introduce **Open Evaluation Agent (Open-EA)** by constructing EA-CoT-10K, a corpus of history-conditioned step-level instruction-tuning records derived from multi-round evaluation rollouts, and training EA-3B from Qwen2.5-3B-Instruct [1] as a local planning backbone. The resulting model preserves the structured reasoning, tool invocation, observation analysis, and summary protocol of the API-based agent, while reducing dependence on proprietary planning backbones. Experiments validate the API-based Evaluation Agent on established T2I/T2V benchmarks and open-ended queries, and evaluate Open-EA on four in-domain and three out-of-domain T2V generator families. The results demonstrate efficient, promptable, and interpretable visual-model evaluation while showing partial cross-family transfer of the learned Open-EA policy. We release the code, data, and models through our GitHub repository at github.com/Vchitect/Evaluation-Agent.

Index Terms—Evaluation, Autonomous Agent, Visual Generation

arXiv:2608.09666v1 [cs.AI] 10 Aug 2026

1 Introduction

Visual generative models have made significant progress in recent years, particularly driven by the advancement of diffusion models [7] and the availability of internet-scale datasets [8]–[10]. These advancements enable the generation of high-quality images and videos, opening up a wide range of applications in content creation, design inspiration, and beyond.

With the advancement of visual generative models, effective evaluation is crucial for understanding their strengths, limitations, and areas for improvement. Existing evaluation frameworks, such as VBench [6], [11], EvalCrafter [12], and T2I-CompBench [5], assess models across multiple dimensions using specific prompts and tailored metrics to ensure comprehensive performance analysis. However, these approaches often demand generating numerous samples, resulting in long evaluation time and high computational costs, particularly for diffusion-based models where sampling is inherently slow due to iterative sampling. Furthermore, these evaluation frameworks are constrained by rigid evaluation pipelines and predefined dimensions, making them less adaptable to open-ended inputs or diverse user needs. Additionally, these methods often produce single numerical scores as outcomes, requiring users to invest additional effort to extract meaningful insights.

In contrast, human evaluators can quickly gain a general understanding of a model's performance by interactively testing a few prompts, forming a sufficient impression without taking too much time. This type of evaluation has several unique advantages for assessing visual generative models. First, it is **fast**, requiring only a small number of samples to assess overall performance. Second, it is **flexible**, allowing intuitive evaluation of various aspects, such as realism, creativity, prompt adherence, or other user-defined criteria. Third, it is **dynamic**, enabling deeper, hierarchical evaluations through continuous adjustments in exploration.

To leverage the strengths of human-like evaluations, we introduce the **Evaluation Agent** [2], a paradigm that mimics human strategies for assessing visual generative models. The Evaluation Agent offers four key features: ***1) Efficiency*:** It dynamically adjusts its evaluation pathway based on intermediate results, uncovering subtle model behaviors and limitations while avoiding redundant test cases for efficient evaluation. ***2) Promptable Evaluation*:** Unlike existing benchmarks with fixed prompts and evaluation metrics, it accepts open-ended user input, allowing for flexible and customized assessments tailored to specific user needs. ***3) Detailed and Interpretable Results*:** It provides interpretable, detailed insights beyond single numerical scores, making results accessible to both experts and non-experts. ***4) Scalability*:** The framework supports seamless integration of new metrics and evaluation tools, ensuring adaptability and growth.

- ✉ *Corresponding author.*
- *Email:* {shulin002, ziqi.huang, ziwei.liu}@ntu.edu.sg
- *A preliminary version of this work was published in ACL 2025 [2].*
- *The authors are with S-Lab, Nanyang Technological University; Agency for Science, Technology and Research (A*STAR); Shanghai Artificial Intelligence Laboratory.*

TABLE 1
Comparison of Open-EA with Traditional T2I and T2V Benchmarks. Open-EA supports customized user queries in natural language and works with both T2I and T2V models. Unlike traditional benchmarks, it dynamically updates the evaluation process using multiple tools, providing comprehensive and explainable results with detailed textual analysis. The Open-EA row encompasses both the API-based Evaluation Agent and the local EA-3B instantiation. The released EA-3B checkpoint is trained and evaluated on T2V, while T2I records form a separate companion set.

Benchmark	Analysis	Customized Queries	Supported Models	# Required Samples	Open Evaluation Request Support	Dynamic Evaluation	Open Tool-Use
FID / FVD [3], [4]	✗	✗	T2I / T2V	2,048	✗ (Fixed-Form)	✗	✗
T2I-CompBench [5]	✗	✗	T2I	18,000	✗ (Pre-Defined)	✗	✗
VBench [6]	✗	✗	T2V	4,730	✗ (Pre-Defined)	✗	✗
Open-EA (Ours)	✓	✓	T2I & T2V	≈ 400	✓ (Open-Ended)	✓	✓

The Evaluation Agent begins by accepting open-ended user input, specifying what to evaluate and which model(s) to assess. Based on this input, it identifies initial evaluation aspects and leverages appropriate tools to conduct the assessment. It then observes the intermediate results and dynamically refines the direction of further exploration. In the end, it generates a detailed natural language response summarizing the evaluation results, providing a comprehensive analysis of the evaluation process and a clear summary of the model's capabilities as specified in the user input. The Evaluation Agent can also automate various applications, including: *1) Model Comparison*: Allowing users to compare models based on specific criteria to determine which performs better in a given aspect. *2) Model Recommendation*: Suggesting the most suitable model for the user's needs by evaluating models against personalized criteria.

We demonstrate the versatility of the Evaluation Agent through experiments on diverse scenarios, including the evaluation of image and video generation models. The results indicate that it delivers performance comparable to full benchmark pipelines while significantly reducing evaluation time. Furthermore, we create an open-ended user query dataset to showcase the Evaluation Agent's flexibility, depth, and accuracy in addressing open-ended queries.

To reduce dependence on API-based agents, we study whether the evaluation behavior can be transferred to a compact open-source backbone. We build EA-CoT-10K by unfolding recorded multi-round evaluation rollouts into history-conditioned, step-level instruction-tuning records that preserve the user query, intermediate observations, reasoning traces, tool-use decisions, and final summaries. We then fine-tune Qwen2.5-3B-Instruct into EA-3B using LLaMA-Factory [13], allowing the same evaluation protocol to be executed by a local model with structured reasoning and tool-call outputs. We refer to this open-source local instantiation as **Open Evaluation Agent (Open-EA)**.

We summarize our contributions as follows:

- We propose the **Evaluation Agent**, a human-like evaluation framework that overcomes the limitations of existing methods in flexibility and efficiency. Our approach will be fully open-sourced.
- We release **EA-CoT-10K**, a first-of-its-kind dataset comprising 10,042 history-conditioned T2V step-level instruction-tuning records derived from multi-round evaluation rollouts, together with a separate 986-record T2I companion set. The corpus captures the decision-making logic of large-scale proprietary models, serving as a critical resource for training autonomous evaluation systems.
- We develop **EA-3B**, a compact open-source planning backbone for Open-EA. Across four in-domain and three out-of-domain T2V generator families, EA-3B follows the same reasoning, tool-selection, observation, and summary protocol as Evaluation Agent and exhibits partial cross-family transfer while reducing dependence on external model APIs.
- We validated our approach on several widely adopted benchmarks, demonstrating that it achieves evaluation accuracy comparable to standard benchmarks while reducing evaluation time by over 90%. We also built an open-ended user query dataset to demonstrate our method's flexibility, depth, and accuracy in handling open-ended evaluation queries.
- Through a comprehensive analysis of how standard benchmarks, human evaluators, and our Evaluation Agent perform evaluations, we provide in-depth insights that serve as important cornerstones for future research in evaluating visual generative models.

2 Related Work

2.1 Visual Generation and Evaluation

Visual generative models [14]–[21] have gained significant attention in recent years. However, unlike perception tasks, which have clear evaluation metrics such as accuracy, evaluating visual generative tasks is more challenging due to the absence of a definitive "ground truth" or single correct answer. Metrics such as FID [4] and FVD [3] are commonly used to measure the distance between generated samples and reference datasets. Recent benchmarks [5], [6], [11], [12], [22]–[24] provide multi-dimensional evaluations tailored to specific model capabilities, while benchmarks for unified multimodal models probe the coupling between understanding and generation [25]–[27]. However, whether relying on metrics like FID [4] and FVD [3] or predefined benchmarks such as VBench, these protocols use fixed test sets and scoring pipelines rather than adaptively choosing what to probe next.

2.2 LLM as a Judge

Recent advances in the understanding and reasoning capabilities of Large Language Models (LLMs) have enabled their use as powerful evaluators [28]–[31]. For instance, Li et al. [32] study customized LLM evaluators across diverse NLP tasks and show that LLM-generated criteria can reduce annotation variance while still requiring careful human scrutiny. Pan et al. [33] demonstrate how LLM-based evaluations enhance downstream tasks for digital agents. These studies showcase the ability of LLMs to reason about criteria and explain judgments. However, they typically score already supplied responses or trajectories under a fixed judging prompt, rather than deciding what evidence to acquire or which evaluation action to execute next. Moreover, devising effective

strategies for domain-specific problems remains challenging [34]. A recent survey systematizes this literature by what to judge, how to judge, and how to benchmark, and identifies dynamic examiner-style pipelines and complex judging agents as an emerging direction [35].

2.3 Agentic Evaluation

Beyond single-pass LLM judges, recent work frames the evaluator itself as an agent that decomposes objectives, acquires evidence, and inspects intermediate states. A recent survey organizes this shift into procedural, reactive, and self-evolving judge agents, spanning multi-agent collaboration, planning, tool integration, memory, and optimization [36]. Zhuge et al.'s Agent-as-a-Judge evaluates complete code-agent trajectories against hierarchical requirements [37]. Task elicitation adaptively proposes and verifies questions to discover target-model failure modes [38], while Mind2Web 2 instantiates task-specific judge agents from tree-structured rubrics for long-horizon, time-varying search tasks [39]. One-Eval instead compiles natural-language evaluation requests into traceable benchmark, metric, execution, and reporting workflows [40]. For safety evaluation, AgenticEval turns regulatory documents and evaluation feedback into a self-evolving LLM-safety test suite [41].

Agentic evaluation has also expanded to visual generation. Scene-graph question answering diagnoses hallucinations in text-to-image models [42], and CIGEval decomposes conditional-image evaluation into fine-grained checks and selects specialized visual tools [43]. For image editing, EdiVal-Agent combines object-centric decomposition with specialist evaluators for multi-turn evaluation [44], while RewardHarness evolves a library of tools and skills from preference demonstrations to judge instruction-guided edits [45]. For video, VideoGen-Eval combines LLM-based content structuring, MLLM judgment, and temporal patch tools in a dynamic evaluation pipeline [46]; VQQA generates visual questions and uses the resulting critiques to refine generation prompts [47]. Most recently, VideoArgus constructs output-blind, input-specific rubrics and executes criterion-specific evidence plans across video generation and editing settings [48]. These systems primarily operate at the instance or predefined-task level, with some additionally optimizing a particular generated sample. In contrast, Evaluation Agent treats the generator itself as the object of investigation: from an open-ended user request, it repeatedly designs prompts, samples new outputs, selects heterogeneous tools, and changes the next probe or termination decision according to accumulated observations.

The reliability of automated evaluators has consequently become a first-class evaluation target. AgentRewardBench finds that no single LLM judge is consistently strongest across web-agent benchmarks [49], while AJ-Bench directly evaluates judge agents' ability to acquire information and verify environment states and execution processes [50]. Controlled meta-evaluation in REFLECT further exposes substantial difficulty in detecting reasoning, tool-use, and evidence failures in research-agent trajectories [51]. More broadly, audits of agentic benchmarks show that task setup, reward design, and outcome-verification procedures can materially distort measured performance [52]. These findings motivate tool-grounded, auditable evaluation traces and caution against treating model-based judgments as infallible ground truth.

2.4 Agent in Planning & Reasoning

Agents and agentic systems are gaining attention for their ability to automate complex tasks and design customized trajectories based on user queries. They have been explored across various domains, including web, mobile, desktop, and operating systems (OS) [53]–[57], showing effectiveness in improving long-horizon task completion. For example, Chain-of-Thought (CoT) [58] and Zero-shot-CoT [59] use prompting techniques to enable step-by-step reasoning. Similarly, ReAct [60] introduces a general paradigm for agent prompting by integrating reasoning traces with task-specific actions through interleaved triplets of "thought-action-observation," thereby incorporating environmental feedback. This action-observation loop provides a natural foundation for an evaluator that revises its tests in response to intermediate evidence. Beyond single-chain reasoning, recent methods explore richer deliberation structures, including Tree of Thoughts [61], Algorithm of Thoughts [62], Graph of Thoughts [63], self-consistency [64], and Reflexion [65]. Tool-use oriented systems and benchmarks, such as API-Bank [66], ToolLLM [67], Toolformer [68], and Gorilla [69], further show that explicit action selection is important for agents operating over external resources. Beyond evaluation, similar reasoning paradigms have also been used to improve generation itself [70], [71]. Evaluation Agent adopts this reasoning-and-action view but grounds each action in visual generation sampling or benchmark/VQA-based assessment.

Recent agent benchmarks have likewise shifted toward stateful, executable, and auditable evaluation. τ^2-Bench models both the agent and the user as tool-using actors that modify a shared environment [72]; Online-Mind2Web evaluates agents on live websites and shows that offline benchmarks can overstate capability [73]; and PaperBench combines hierarchical rubrics co-developed with paper authors with a judge evaluated on a separate benchmark for long-horizon research replication [74]. These benchmarks primarily evaluate agents as task performers, whereas our framework uses an agent to adaptively design and execute evaluations of visual generative models.

3 METHODS

3.1 Preliminaries: Evaluation of Visual Generative Models

Evaluation Benchmark. Let $C = \{c_j \mid j \in \{1, 2, 3, \ldots, N\}\}$ denote the condition set (*i.e.*, test cases), whose elements can be text prompts, input images, class labels, or conditions in other formats. For unconditional generation, C is empty or consists of random seeds. Existing evaluation approaches typically pre-define hundreds or thousands of conditions, making visual sampling computationally expensive. In the Evaluation Agent framework, C is determined dynamically during evaluation and usually contains only a small number of cases.

Sampling. $v_j = G(c_j)$ where G is the visual generative model, which generates the visual output v_j (*i.e.*, images or videos) given an optional condition c_j. $V = G(C)$ where $V = \{v_j \mid j \in \{1, 2, 3, \ldots, N\}\}$ represents the set of generated visuals for the entire condition set C.

Evaluation Pipeline. Existing evaluation methods usually follow a fixed pipeline to evaluate all the images or videos V sampled from the pre-defined benchmark C.

$$y_j = e_k(v_j, c_j) \tag{1}$$

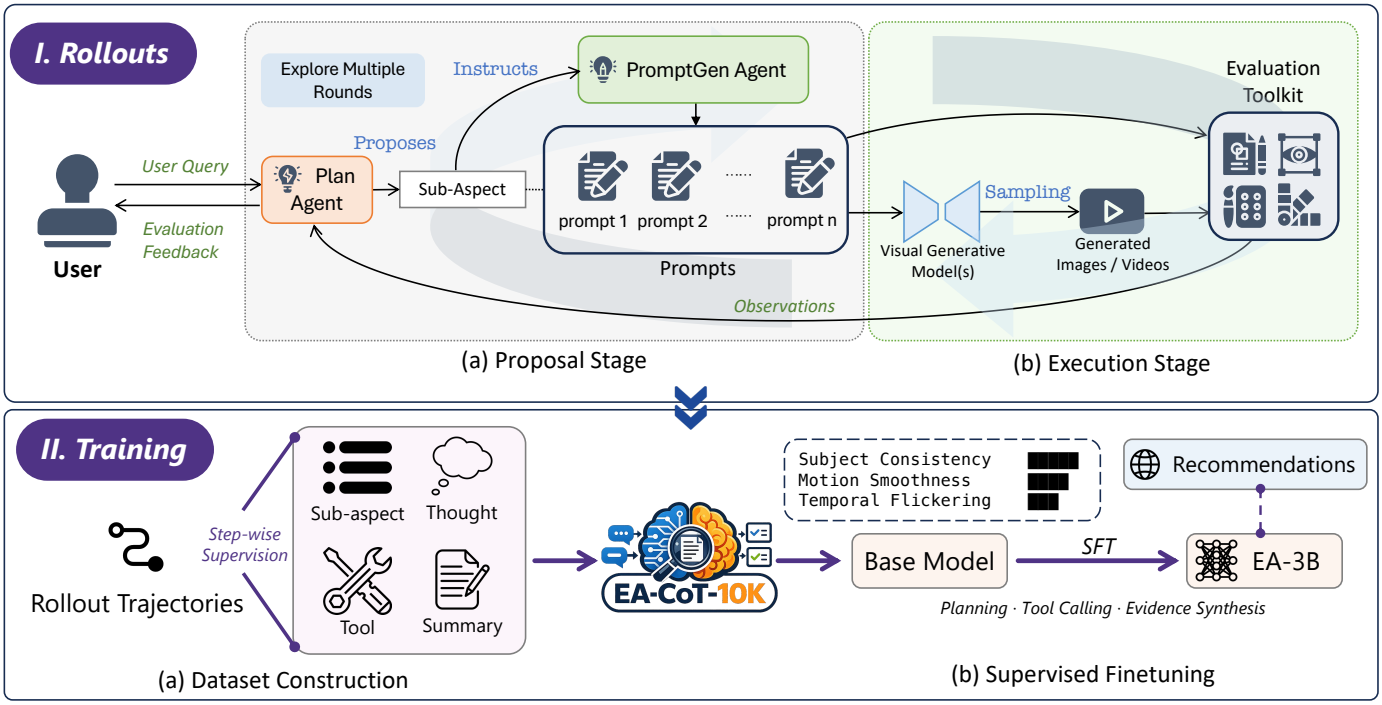

Fig. 1. **Overview of the Evaluation Agent Framework.** The upper rollout pipeline contains a proposal stage, where the Plan Agent decomposes the user request and the PromptGen Agent prepares evaluation prompts, and an execution stage, where visual outputs are sampled and assessed by selected tools. These stages form an iterative observation loop for dynamic, user-conditioned evaluation. The bottom training pipeline converts completed rollouts into step-wise supervision. Sub-aspects, thoughts, tool calls, observations, and final summaries are organized into EA-CoT-10K and used to fine-tune a base model into EA-3B, allowing Open-EA to follow the same evaluation protocol with a compact local planning backbone.

TABLE 2
Evaluation Results Comparison with VBench [6]. We evaluated 15 specific ability dimensions in VBench using our Evaluation Agent and compared its results against VBench in terms of conclusion accuracy. The numerical results show the percentages of the Evaluation Agent's correct predictions falling either within the exact range (left) or within an error margin of one range (right) across ten trials.

Models	Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Object Class
Latte-1 [17]	50% / 80%	0% / 30%	40% / 70%	30% / 70%	60% / 100%	70% / 100%	40% / 50%
ModelScope [18]	80% / 80%	80% / 90%	60% / 80%	60% / 100%	60% / 100%	100% / 100%	0% / 50%
VideoCrafter-0.9 [20]	100% / 100%	80% / 100%	70% / 100%	80% / 100%	90% / 100%	20% / 100%	20% / 60%
VideoCrafter-2 [19]	10% / 100%	60% / 100%	30% / 90%	30% / 80%	80% / 100%	50% / 100%	70% / 100%

Multiple Objects	Human Action	Color	Spatial Relationship	Scene	Appearance Style	Temporal Style	Overall Consistency
40% / 100%	10% / 10%	30% / 70%	10% / 80%	20% / 40%	70% / 90%	40% / 100%	70% / 100%
50% / 100%	10% / 40%	0% / 20%	10% / 30%	20% / 100%	90% / 100%	50% / 90%	20% / 100%
80% / 100%	10% / 30%	10% / 40%	20% / 100%	30% / 100%	60% / 100%	80% / 100%	0% / 80%
20% / 60%	10% / 90%	90% / 100%	0% / 70%	0% / 10%	80% / 100%	80% / 100%	60% / 100%

where $e_k \in E$ is an evaluation function for an aspect such as aesthetics or compositionality. Given the generated visual v_j and the optional condition c_j, it produces the evaluation result y_j.

Some reference/statistics-based evaluation frameworks like FID and FVD also use reference datasets V_r for calculating the results.

$$Y = E(V, V_r, C) \tag{2}$$

In existing evaluation approaches, E is a pre-defined set of evaluation dimensions, or a single evaluation metric, which limits the possible evaluation aspects from the beginning, and requires assessing all the aspects even if some are not needed in some cases. In our Evaluation Agent framework, the evaluation tool e_k is dynamically determined during the evaluation process.

3.2 The Evaluation Agent Framework

Our Evaluation Agent framework is powered by LLM-based agents, leveraging their advanced planning capabilities to simulate human-like behaviors for efficient and flexible visual model assessments. As illustrated in Figure 1, the rollout component of the framework operates in two stages: the proposal stage and the execution stage. By iteratively interacting and looping between these stages, the framework dynamically evaluates models based on user queries. A training component reuses the rollout traces as supervision. Each trajectory records the selected sub-aspects, reasoning thoughts, tool interactions, observations, and summaries; these structured records are assembled into EA-CoT-10K and used to train EA-3B. This connects the API-based Evaluation Agent with Open-EA, where the trained local backbone follows the same propose–sample–evaluate–

TABLE 3
Evaluation Results Comparison with T2I-CompBench [5]. We evaluated four ability dimensions in T2I-CompBench using our Evaluation Agent and compared its results with those of T2I-CompBench in terms of conclusion accuracy. The numerical results show the percentages of the Evaluation Agent's correct predictions falling either within the exact range (left) or within an error margin of one range (right) across ten trials.

Models	Color Binding	Shape Binding	Texture Binding	Non-Spatial Relationships
SD1.4 [14]	50% / 100%	100% / 100%	0% / 100%	50% / 100%
SD2.1 [14]	100% / 100%	60% / 100%	80% / 100%	60% / 100%
SDXL [15]	100% / 100%	20% / 100%	80% / 100%	60% / 100%
SD3.0 [16]	20% / 90%	0% / 90%	0% / 70%	80% / 90%

observe protocol during deployment.

3.2.1 Proposal Stage

The Proposal Stage consists of two agents: the Plan Agent and the PromptGen Agent. The Plan Agent is responsible for planning, observing, and summarizing the evaluation process based on the user's query, while the PromptGen Agent focuses specifically on the design aspects.

Plan Agent. We design the Plan Agent to simulate human behavior during the evaluation process, including planning and adjusting the evaluation direction, observing intermediate results, and summarizing the final outcomes. As the core component of the framework, the Plan Agent not only interacts with the user but also drives the entire evaluation process. Specifically, upon receiving a user query, the Plan Agent identifies an initial sub-aspect for evaluation and iteratively refines it based on feedback from intermediate results. This process continues until sufficient information is collected, after which the agent provides a detailed analysis and summary. At each step, the Plan Agent is required to provide an explicit rationale for the selected sub-aspect and, when it terminates the loop, to justify why the collected observations are sufficient to answer the user query. For closed-domain benchmark queries, it also selects the corresponding evaluation tool; for open-ended queries, it formulates the evaluation direction that will later be converted into VQA-style observations.

PromptGen Agent. The PromptGen Agent mimics human behavior in designing prompts for visual generative models based on the plan developed during the evaluation process. Specifically, it generates tailored prompts for each sub-aspect identified by the Plan Agent, enabling focused content generation and evaluation. Additionally, the PromptGen Agent can reference and utilize well-established prompts from existing benchmarks. In benchmark-based evaluation, PromptGen selects prompts from the official prompt lists and keeps the evaluation comparable to the original benchmark protocol. In open-ended evaluation, it jointly designs generation prompts and VQA questions so that each sampled visual output can be converted into textual evidence for the Plan Agent.

3.2.2 Execution Stage

The Execution Stage is responsible for sampling and evaluating the model using the appropriate tools, as specified in the Proposal Stage, and for returning the final evaluation results.

Visual Generative Models. This component takes prompts designed by the PromptGen Agent as input and generates corresponding visual content, which is then used for subsequent evaluation.

Evaluation Toolkit. The Evaluation Toolkit consists of a set of elementary evaluation tools for visual generative models. This module is open and extensible, allowing for continuous expansion. We have integrated several existing evaluation tools from well-known benchmarks for different modalities of visual generation models. To support the evaluation of open-ended user queries, we have introduced a paradigm based on vision-language models (VLMs), leveraging the Visual Question Answering (VQA) format to enable flexible assessments across various aspects of the models. Upon receiving the generated samples from visual generative models along with the corresponding prompts, the module utilizes the appropriate tools to evaluate each sample. All evaluation results are then compiled and returned to the Plan Agent for further proposals or summarization. The implemented toolkit covers VBench dimensions for text-to-video evaluation, T2I-CompBench tools for compositional text-to-image evaluation, and a VLM-based VQA tool for open-ended image-model evaluation. This design keeps the tool interface modular: each evaluation round returns structured observations rather than only final benchmark scores.

3.2.3 Overall Pipeline

The Evaluation Agent's process is dynamic and multi-round, with each round comprising a proposal stage and an execution stage. By interacting and looping through these stages, we achieve dynamic evaluation, where the evaluation process adapts based on intermediate observations and initial user query. This dynamic approach allows the Evaluation Agent to refine its focus iteratively, adjusting its exploration direction and prompt design based on an evolving understanding of the model's capabilities. Consequently, the evaluation process becomes more efficient and targeted, systematically identifying the strengths and limitations of generative models.

Operationally, each round follows a propose–sample–evaluate–observe loop. The agent proposes one sub-aspect, samples a small set of images or videos, invokes one selected tool, receives per-sample or aggregate observations, and then decides whether to continue probing or produce a final answer. This loop allows the evaluation to expand breadth-wise across categories or depth-wise toward harder cases, depending on the evidence gathered in previous rounds.

3.3 EA-CoT Data Construction and Open-EA Training

EA-CoT Data Construction. To transfer the evaluation protocol into a trainable local agent, we construct EA-CoT from recorded evaluation trajectories rather than isolated question-answer pairs. Each trajectory preserves the user query, the previous observations available to the agent, the reasoning segment used to select the next sub-aspect, the selected tool when applicable, the returned information, and the final evidence-based summary. The post-processing pipeline converts these trajectories into instruction-tuning records with explicit tags such as `<think>`, `<tool>`, `<information>`, and `<summary>`. This format exposes both the intermediate decision process and the terminal evaluation judgment to supervised training. As illustrated in Figure 2(a), post-processing

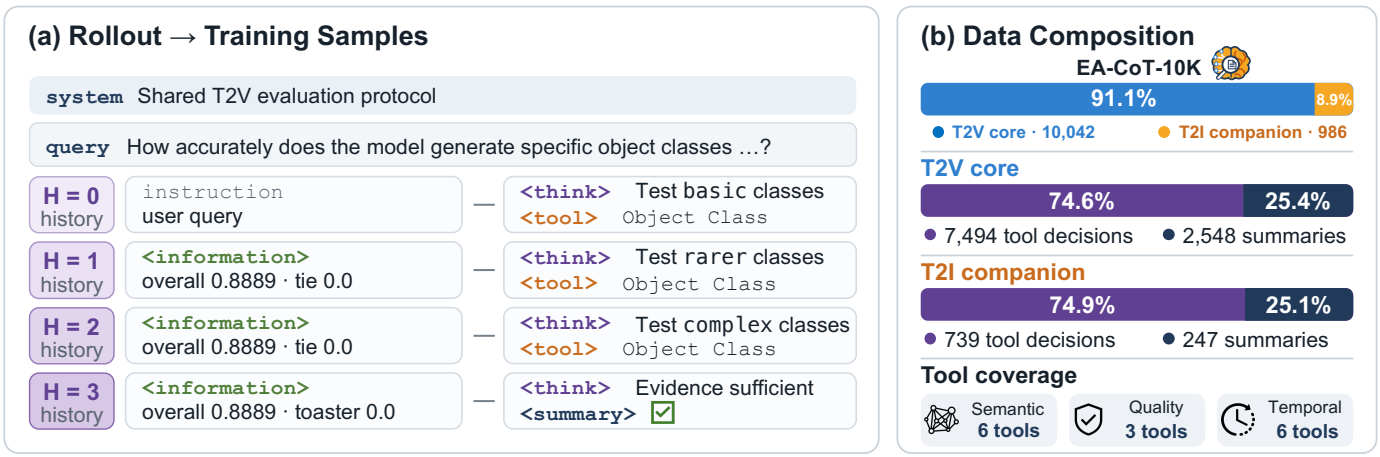

Fig. 2. **EA-CoT-10K Construction and Data Composition.** (a) A representative T2V evaluation rollout is unfolded into history-conditioned step-level instruction-tuning records (training samples). Under a shared system protocol, each record retains preceding turns in its history and supervises either the next reasoning-and-tool decision or, after sufficient evidence has been collected, the terminal summary; text is abbreviated for clarity. (b) The upper bar compares the relative sizes of the separately processed T2V core (10,042 records; 91.1%) and T2I companion set (986 records; 8.9%). The lower bars show their within-set tool-decision/summary composition: 7,494/2,548 (74.6%/25.4%) for T2V and 739/247 (74.9%/25.1%) for T2I. The T2V core covers 15 VBench tools grouped into six semantic, three quality, and six temporal tools.

unfolds each rollout into history-conditioned step-level instruction-tuning records, each serving as a training sample whose target is either the next reasoning-and-tool decision or the terminal summary. Figure 2(b) compares the scale and within-set record composition of the T2V core and the separate T2I companion set and summarizes the T2V tool coverage.

Data Filtering. Before packaging EA-CoT-10K, we apply a two-stage filtering procedure to improve the faithfulness of the supervision. First, we perform deterministic score-based filtering using the tool scores stored in each rollout and the corresponding dimension-specific scoring reference. Trajectories whose generated images or videos exhibit severe prompt-visual misalignment, such as extremely low aggregate scores or per-prompt scores in the lowest-quality range for the selected dimension, are removed because their visual evidence is too noisy to support reliable reasoning. Second, we deploy an MLLM as a judge to verify whether the textual reasoning is grounded in the original visual outputs. For each candidate trajectory, the judge is given the user query, generated prompts, sampled visual evidence, tool observations, and the agent's reasoning and summary. It then checks whether the stated objects, actions, attributes, temporal changes, and final conclusion are supported by the generated video or image. We discard trajectories where the reasoning contradicts the visual content, hallucinates absent details, overlooks obvious generation failures, or maps the score evidence to an inconsistent qualitative judgment. In our processed artifacts, the T2V table-context variant contains 10,042 history-conditioned step-level records (7,494 tool-decision records and 2,548 summary records) spanning 15 VBench evaluation tools. A separate T2I companion set contains 986 records (739 tool-decision records and 247 summary records) across four T2I-CompBench tools. The companion set is not mixed into the current EA-3B training run. Each Alpaca-style record contains `instruction`, `input`, `output`, `system`, and `history` fields.

Open-EA Training. We obtain EA-3B by full-parameter supervised fine-tuning of Qwen2.5-3B-Instruct [1] on the 10,042-record T2V core using LLaMA-Factory [13]. We train for three epochs with a maximum sequence length of 4096, an effective batch size of 64, and a learning rate of 1×10^{-5}, using cosine decay and a warmup ratio of 0.1. The structured reasoning, tool, information, and summary fields teach EA-3B to choose the next evaluation action from accumulated evidence and to terminate with a grounded assessment. EA-3B thereby provides Open-EA with a compact local planning backbone while preserving the interaction protocol and tool workflow of the API-based Evaluation Agent.

4 Experiments

We first validate the efficiency of our Evaluation Agent on established benchmarks for visual generative models and then demonstrate the flexibility, depth, and accuracy of our approach in handling open-ended user queries on our self-constructed dataset. We evaluate the local Open-EA instantiation on four in-domain and three out-of-domain T2V generator families, analyze the dimensions on which its tier predictions remain difficult, and characterize its end-to-end operational cost separately from that of the API-based Evaluation Agent.

4.1 Experiments on Existing Benchmarks

We validate the effectiveness of our framework on both the Text-to-Video (T2V) and Text-to-Image (T2I) tasks. In this subsection, all closed-domain experiments use LLM agents with `gpt-4o-2024-08-06` as the backbone and a temperature of 0.7. The system prompts follow a CoT/ReAct-style protocol [58], [60], requiring the agent to reason before selecting a tool, interpret intermediate observations, and justify its final response.

4.1.1 Experimental Setup

Visual Generative Models. For the T2V task, we select four open-source models: VideoCrafter-0.9 [20], VideoCrafter-2 [19], Latte-1 [17], and ModelScope [18]. Similarly, for the T2I task, we choose four well-known open-source models: SD(Stable Diffusion)1.4 [14], SD2.1 [14], SDXL [15], and SD3.0 [16]. We validate the efficiency and accuracy of our method using these models within the evaluation frameworks for both T2I and T2V tasks.

Visual Generation Benchmarks. For the T2V and T2I tasks, we adopt well-established and comprehensive evaluation frameworks

TABLE 4
Evaluation Cost Comparison with Standard Benchmarks. We compare the original VBench and T2I-CompBench pipelines with the API-based Evaluation Agent. Each entry reports wall-clock evaluation time and the number of generated samples. Benchmark costs are shown both for the complete benchmark and as an average per evaluated dimension, while the Evaluation Agent cost corresponds to one user-requested dimension.

Task / Benchmark	Model	Standard Benchmark Total Cost ↓	Standard Benchmark Avg. Cost per Dimension ↓	Evaluation Agent Cost per Query ↓
T2V / VBench	Latte-1 [17]	2557 min / 4355 samples	170 min / 290 samples	15 min / 25 samples
	ModelScope [18]	1160 min / 4355 samples	77 min / 290 samples	6 min / 23 samples
	VideoCrafter-0.9 [20]	1459 min / 4355 samples	97 min / 290 samples	9 min / 24 samples
	VideoCrafter-2 [19]	4261 min / 4355 samples	284 min / 290 samples	24 min / 23 samples
T2I / T2I-CompBench	SD1.4 [14]	563 min / 12000 samples	141 min / 3000 samples	5 min / 26 samples
	SD2.1 [14]	782 min / 12000 samples	196 min / 3000 samples	6 min / 26 samples
	SDXL [15]	1543 min / 12000 samples	386 min / 3000 samples	8 min / 26 samples
	SD3.0 [16]	1410 min / 12000 samples	353 min / 3000 samples	7 min / 25 samples

tailored to their respective domains. Specifically, for the T2V task, we use VBench [6], a comprehensive and fine-grained evaluation framework for video generation. For the T2I task, we select T2I-CompBench [5], which assesses the compositional generation capabilities of T2I models across multiple dimensions. We validate the effectiveness of our method using these evaluation frameworks.

Comparison Setup. To ensure comparability and fairness in our experiments, we restrict the evaluation tools and prompts available to the Evaluation Agent. For T2V tasks, we used only the metrics and prompt lists from VBench, and for T2I tasks, only those from T2I-CompBench. For comparison, we categorize performance into five levels based on the performance density distribution. For VBench, we evaluate 15 dimensions and map raw metric outputs into five categorical tiers: `Very High`, `High`, `Moderate`, `Low`, and `Very Low`. For proportion-style dimensions, such as `Dynamic Degree`, the tiering thresholds are constructed from leaderboard statistics. For T2I-CompBench, we use four dimensions, `Color Binding`, `Shape Binding`, `Texture Binding`, and `Non-Spatial Relationships`; we exclude `Spatial Relationships` because its score distribution and statistical interpretation are not directly comparable under our small-sample setting.

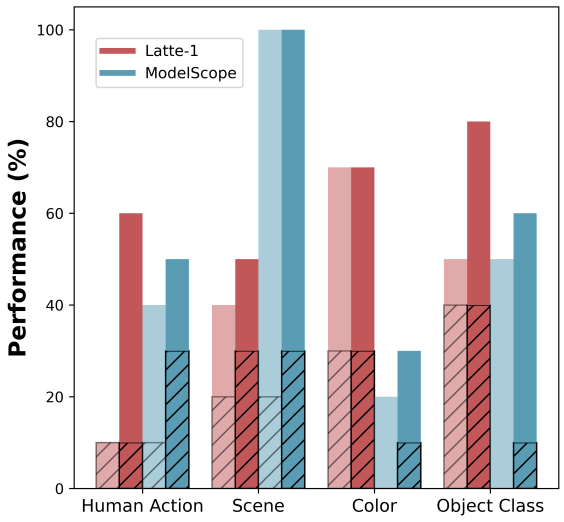

Fig. 3. **Effect of Prompt Budget on Percentage-Based VBench Dimensions.** Lighter bars use the original prompt budget and darker bars use 30 prompts. Hatched portions denote exact-tier predictions and solid portions denote predictions within one neighboring tier.

4.1.2 Results Analysis

API-Based Evaluation Cost. Table 4 summarizes the evaluation-cost comparison between standard benchmarks and the API-based Evaluation Agent. Relative to the average cost of one standard benchmark dimension, the API-based Evaluation Agent reduces both wall-clock time and the number of generated samples across all evaluated T2V and T2I models.

Validation on VBench. To highlight the efficiency of our proposed methods, we compare the time consumption and sample counts between VBench and our approach. As shown in Table 4, our method reduces evaluation time by more than $10\times$. Additionally, Table 2 confirms the consistency of our evaluation results with VBench across various dimensions. The quantitative results show that the Evaluation Agent achieves high prediction accuracy across most dimensions, demonstrating that our approach maintains accuracy while significantly reducing evaluation time.

Validation on T2I-CompBench. We evaluate the Evaluation Agent's performance on T2I tasks, as shown in Table 4. Instead of requiring thousands of samples and several hours, the Evaluation Agent completes evaluations with just about 26 samples in five to eight minutes per dimension. Table 3 compares the evaluation results, showcasing high accuracy within an error margin of one range.

Effect of the Prompt Budget. The lower accuracy on Human Action, Scene, Color, and Object Class is consistent with their reliance on percentage estimates formed from binary sample-level outcomes. We therefore evaluate Latte-1 and ModelScope with 30 prompts instead of the default budget, which ranges from three to nine prompts per round. Figure 3 shows that Latte-1 improves from 25.0% to 27.5% in Exact accuracy and from 42.5% to 65.0% in Within-1 accuracy. ModelScope improves from 7.5% to 20.0% in Exact accuracy and from 52.5% to 60.0% in Within-1 accuracy. When averaged across the two generators, Within-1 improves by 15.0 percentage points, which is twice the 7.5-point improvement in Exact accuracy. The larger gain under the one-tier tolerance suggests that additional prompts chiefly stabilize the estimated capability region, while the remaining Exact gap shows that a larger budget alone does not resolve errors near tier boundaries.

Backbone Robustness of the API-Based Evaluation Agent. Keeping the evaluation protocol fixed, we replace `gpt-4o-2024-08-06` with `claude-3-5-sonnet-20241022`. Across all evaluated model–dimension pairs, GPT-4o and Claude obtain 44.8% / 81.3% versus

TABLE 5
Open-EA Evaluation across VBench Dimensions. Each entry reports Exact / Within-1 tier accuracy over ten trials. Four in-domain generators are evaluated alongside three out-of-domain families to measure cross-family transfer.

Models	Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Object Class
In-Domain Models							
Latte-1	60% / 100%	40% / 70%	90% / 100%	80% / 90%	30% / 80%	40% / 80%	0% / 0%
ModelScope	30% / 50%	40% / 60%	80% / 90%	60% / 80%	40% / 90%	70% / 100%	0% / 30%
VideoCrafter-0.9	100% / 100%	100% / 100%	100% / 100%	100% / 100%	100% / 100%	90% / 100%	0% / 10%
VideoCrafter-2	40% / 100%	30% / 100%	40% / 80%	0% / 100%	50% / 100%	80% / 100%	100% / 100%
Out-of-Domain Models							
Show-1	0% / 40%	30% / 80%	10% / 60%	40% / 80%	50% / 100%	30% / 100%	10% / 30%
CogVideoX-2B	70% / 100%	50% / 90%	80% / 80%	0% / 20%	50% / 80%	20% / 50%	60% / 60%
CogVideoX-5B	50% / 80%	30% / 80%	10% / 90%	30% / 80%	50% / 70%	20% / 70%	0% / 20%

Models	Multiple Objects	Human Action	Color	Spatial Relationship	Scene	Appearance Style	Temporal Style	Overall Consistency
In-Domain Models								
Latte-1	20% / 100%	0% / 70%	0% / 0%	0% / 0%	80% / 80%	40% / 80%	60% / 90%	20% / 100%
ModelScope	20% / 100%	0% / 0%	10% / 30%	30% / 30%	0% / 70%	10% / 100%	10% / 60%	30% / 100%
VideoCrafter-0.9	100% / 100%	0% / 0%	60% / 80%	30% / 100%	0% / 100%	60% / 100%	40% / 100%	80% / 100%
VideoCrafter-2	10% / 80%	0% / 60%	100% / 100%	0% / 60%	40% / 40%	20% / 100%	0% / 100%	20% / 100%
Out-of-Domain Models								
Show-1	30% / 80%	0% / 60%	10% / 40%	0% / 40%	0% / 0%	10% / 60%	20% / 90%	70% / 100%
CogVideoX-2B	40% / 40%	60% / 60%	20% / 30%	0% / 0%	60% / 60%	30% / 80%	20% / 90%	10% / 80%
CogVideoX-5B	10% / 30%	50% / 50%	20% / 100%	100% / 100%	0% / 0%	40% / 70%	30% / 100%	80% / 100%

41.5% / 77.2% Exact / Within-1 accuracy on VBench, and 53.8% / 96.3% versus 54.4% / 100.0% on T2I-CompBench. Their close aggregate results indicate that the dynamic evaluation loop is not tied to GPT-4o, although individual dimensions differ. These experiments assess the API-based Evaluation Agent rather than EA-3B.

4.2 Open-EA Quantitative Evaluation

We evaluate EA-3B on all 15 VBench dimensions using four in-domain and three out-of-domain generators. The EA-CoT-10K training traces include Latte-1, ModelScope, VideoCrafter-0.9, and VideoCrafter-2, whereas Show-1 and the two CogVideoX variants do not appear in the training traces and therefore serve as out-of-domain generator families. Table 5 reports Exact and Within-1 tier accuracy over ten trials for every model–dimension pair.

Generalization and Tier Calibration. Open-EA retains partial cross-family generalization: the micro-averaged Exact / Within-1 score decreases from 41.3% / 77.3% in domain to 31.1% / 64.9% out of domain. The persistent Exact–Within-1 gap suggests that coarse capability placement transfers more reliably than exact tier calibration; VideoCrafter-2 is the clearest example, with 35.3% Exact but 88.0% Within-1 accuracy.

Difficulty across Evaluation Signals. Table 5 shows that performance varies sharply with the evaluation signal. Across all generators, Overall Consistency and Temporal Style achieve 97.1% and 90.0% Within-1 accuracy, whereas Object Class, Human Action, and Spatial Relationship reach only 35.7%, 42.9%, and 47.1%, respectively. VideoCrafter-0.9 makes this contrast especially clear: it attains at least 80% Within-1 accuracy on 13 of the 15 dimensions, yet scores 0% / 0% on Human Action and 0% / 10% on Object Class. A plausible explanation is that global quality and temporal properties provide repeated video-level evidence, while class-specific or relational dimensions infer a model-level tier from a small number of prompt-specific successes and are therefore more sensitive to concept coverage and tier boundaries. Together with the prompt-budget ablation in Figure 3, this pattern suggests that Open-EA transfers broad quality and consistency judgments more reliably than fine-grained semantic coverage, making prompt diversity and tier calibration the clearest opportunities for improvement.

TABLE 6
Open-EA End-to-End Inference Speed. Wall-clock runtime over all successful saved traces, including planning, visual generation, tool execution, observation aggregation, and summarization. This trace pool is summarized independently of the fixed ten-trial accuracy evaluation.

Models	Saved Runs	Median Time per Query ↓	Mean Time per Query ↓
In-Domain Models with Saved Traces			
Latte-1	148	7.87 min	9.23 min
ModelScope	147	1.86 min	2.41 min
Out-of-Domain Models			
Show-1	117	144.35 min	176.64 min
CogVideoX-2B	155	20.29 min	23.89 min
CogVideoX-5B	149	52.42 min	57.02 min

End-to-End Runtime. Table 6 reports wall-clock time for the complete local Open-EA pipeline, rather than EA-3B latency alone.

Median runtime spans 1.86–144.35 minutes, a 77.6× range

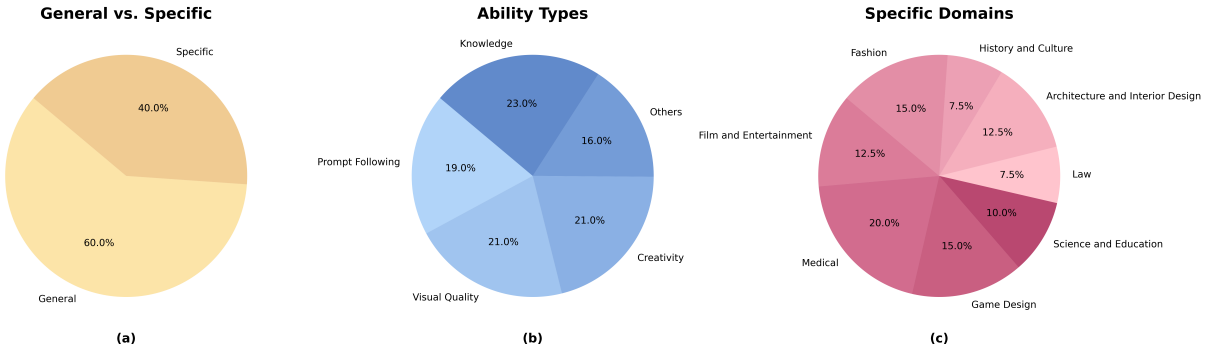

Fig. 4. **Data Distribution of Open-Ended User Query Dataset.** We analyze the constructed open-ended user query dataset from three aspects: General/Specific, Ability, and Specific Domain. The distribution shows that the collected queries cover both broad capability questions and application-oriented use cases.

despite the shared planning backbone, while consistently higher means indicate a right tail of slower runs. Because the saved traces differ in generator and tool mix, these measurements describe full-pipeline deployment cost; they cannot rank EA-3B planning speed or be compared directly with the API-based Evaluation Agent costs in Table 4.

4.3 Experiments on Open-Ended User Query

We demonstrate the flexibility of our framework and the benefits of its dynamic evaluation through experiments on an open-ended user query dataset that we collected and curated. Unless otherwise stated, the cases in this subsection use the API-based Evaluation Agent rather than EA-3B.

4.3.1 Open-Ended User Query Dataset

We create an open-ended user query dataset comprising 100 user queries focused on evaluating generative model capabilities. Each query is manually labeled with `Ability`, `General/Specific`, and `Specific Domain` tags. The dataset is collected through a user study and then cleaned, filtered, and expanded into 100 final queries. The ability labels cover `Prompt Following`, `Visual Quality`, `Creativity`, `Knowledge`, and `Others`; the scope labels distinguish general questions from application-specific questions; and the domain labels cover areas such as law, film and entertainment, fashion, game design, architecture and interior design, medical, science and education, and history and culture. In the current dataset, 60 queries are general and 40 are specific, and the ability categories are distributed across knowledge, visual quality, creativity, prompt following, and other model capabilities.

4.3.2 Experimental Setup

The Evaluation Agent demonstrates strong planning capabilities and accepts any input, but its effectiveness is limited by restrictive evaluation tools, which hinder its ability to handle open-ended queries. To overcome this limitation, we propose a simple yet intuitive solution: leveraging a VLM as an evaluation tool in the form of VQA. During each evaluation round, the PromptGen Agent not only designs prompts for specific sub-aspects but also generates corresponding questions based on the content of each prompt and the aspects to be evaluated. The generated sample and questions are input into the VLM, which provides answers that are then fed back to the Plan Agent for further analysis and planning. This VQA formulation converts open-ended visual inspection into a structured textual observation. As a result, the Plan Agent can compare evidence across rounds, decide whether to broaden the tested scenarios or increase prompt complexity, and produce a final answer grounded in the generated samples rather than in a single scalar metric.

4.3.3 Open-Ended User Query Evaluation

Most visual generation benchmarks use predefined dimensions and prompts to evaluate models, but this fixed approach often overlooks users' specific needs, such as handling unique scenarios or objects. A user study found that 67.44% (29 of 43) participants prioritized models meeting their specific needs over general performance. Additionally, fixed prompts can lead to targeted optimization, resulting in misleading evaluations.

We address these issues using the Evaluation Agent, which conducts dynamic, multi-round assessments of a model's capabilities for open-ended queries, with flexible prompt design at each stage. Figure 5 presents a complete evaluation trajectory for the query: "How precisely can the user specify object relationships?" The agent first evaluates simple spatial relationships between two objects, then increases the difficulty to relationships involving three or more objects, and finally probes non-standard and imaginative configurations. The intermediate VQA observations show that the evaluated model captures common spatial relationships but may omit fine-grained constraints, such as a leaf held in a bird's beak or a ladder reaching the ground. By adapting its focus across rounds and synthesizing the accumulated evidence, the agent distinguishes reliable performance on simple and moderately complex relationships from persistent limitations on highly detailed or unconventional arrangements. Additional evaluated queries include artistic-style preservation and specific object-count generation, where the agent records the sub-aspects, thoughts, observations, and final conclusions in the same trajectory format. As shown, our Evaluation Agent framework provides precise, detailed, and user-focused evaluations for open-ended queries.

5 Further Discussions

In this section, we discuss the unique aspects of the Evaluation Agent compared to traditional benchmarks, as well as its potential broader applications.

Dynamic and Multi-Step Evaluation. One of the core features of the Evaluation Agent is its dynamic, multi-step evaluation process. This structured evaluation paradigm enables the discovery of nuanced differences in model capabilities and provides a more detailed analysis of a model's strengths and weaknesses.

Fig. 5. **A Complete Case of Open-Ended User Query Evaluation.** Given the query "How precisely can the user specify object relationships?", the Evaluation Agent progressively probes simple two-object relations, complex relations among three or more objects, and non-standard imaginative configurations. Round-wise VQA observations guide subsequent probes and support the final conclusion: the evaluated model handles simple and moderately complex relations but remains less reliable under highly detailed or unconventional constraints.

Specifically, it allows for a hierarchical, step-by-step evaluation, progressing from simple to complex tasks, as well as category-based assessments that measure performance across different content types within the same dimension. In contrast, traditional visual generation evaluation benchmarks, while incorporating diverse prompts carefully designed for each dimension, suffer from limitations such as fixed prompts and a lack of fine-grained prompt categorization. These constraints reduce flexibility, make it harder to draw valuable insights, and increase the risk of models being over-optimized for specific prompts. Furthermore, dynamic evaluations help avoid redundant testing, significantly enhancing efficiency.

Open-Ended Evaluation Toolkit. Our framework for evaluating open-ended queries relies on two key aspects: the planning and reasoning capabilities of LLM Agents and the evaluation toolkit's ability to assess diverse dimensions. One approach to building the evaluation toolkit is integrating various tools, allowing the agent to select the most suitable one for each task. However, current evaluation tools are limited, often focusing only on general evaluations and lacking sensitivity to fine-grained details. For instance, CLIPScore [75] captures the general similarity between an image and a caption but fails to detect subtle changes, such as variations in object counts, limiting its effectiveness in specific evaluations. An alternative approach is to design a versatile evaluation tool capable of assessing multiple aspects. A VQA-based format using VLMs is particularly promising, enabling fine-grained evaluation through targeted questions and providing detailed textual outputs that integrate well with LLM Agents. While this method's effectiveness depends on the VLM's capabilities, current VLMs already demonstrate impressive results, with future advancements poised to further enhance performance.

Agent Reliability and Error Handling. Although the Plan Agent follows a structured multi-step protocol, agentic evaluation can still fail through unreasonable tool choices, repetitive proposals, or premature/non-terminating decisions. We therefore combine prompt-level safeguards, such as requiring an explicit thought before each action and before termination, with system-level constraints such as a maximum number of rounds. These mechanisms do not remove the need for stronger reliability design, but they make the evaluation trace auditable and prevent unbounded evaluation cost.

Broader Applications. The Evaluation Agent not only evaluates the performance of a single model but also facilitates the direct comparison of two models' strengths and weaknesses in specific dimensions by assessing them simultaneously during execution. This feature enables users to determine which model excels in particular areas. Moreover, by accumulating evaluation results across various capabilities, a database can be constructed. Once sufficient information about multiple models is collected, this database can serve as the foundation for building a recommendation system, capable of suggesting the most suitable model based on the user's specific needs.

6 Conclusion

We present Evaluation Agent, a dynamic and promptable framework for assessing visual generative models beyond rigid evaluation pipelines. Unlike traditional methods that rely on fixed benchmarks and time-consuming sampling processes, the Evaluation Agent adapts its evaluation trajectory to intermediate evidence and user-specified criteria. This design reduces the number of required samples while supporting flexible integration of evaluation tools and visual generative models. By open-sourcing this framework, we aim to inspire further research in the development of more flexible and efficient evaluation methods for visual generative models. The Open-EA results show that this protocol can be distilled into a compact local agent trained from evaluation trajectories, reducing dependence on proprietary planning backbones while preserving promptable, tool-grounded evaluation.

7 Acknowledgment

This study is supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 (MOET2EP20221-0012, MOE-T2EP20223-0002), and under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

REFERENCES

[1] Qwen Team, "Qwen2.5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.

[2] F. Zhang, S. Tian, Z. Huang, Y. Qiao, and Z. Liu, "Evaluation Agent: Efficient and promptable evaluation framework for visual generative models," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2025, pp. 7561–7582.

[3] T. Unterthiner, S. Van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Towards accurate generative models of video: A new metric & challenges," *arXiv preprint arXiv:1812.01717*, 2018.

[4] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[5] K. Huang, K. Sun, E. Xie, Z. Li, and X. Liu, "T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[6] Z. Huang, Y. He, J. Yu, F. Zhang, C. Si, Y. Jiang, Y. Zhang, T. Wu, Q. Jin, N. Chanpaisit, Y. Wang, X. Chen, L. Wang, D. Lin, Y. Qiao, and Z. Liu, "VBench: Comprehensive benchmark suite for video generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[7] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[8] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "Frozen in time: A joint video and image encoder for end-to-end retrieval," in *IEEE International Conference on Computer Vision*, 2021.

[9] T.-S. Chen, A. Siarohin, W. Menapace, E. Deyneka, H.-w. Chao, B. E. Jeon, Y. Fang, H.-Y. Lee, J. Ren, M.-H. Yang, and S. Tulyakov, "Panda-70m: Captioning 70m videos with multiple cross-modality teachers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[10] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision (IJCV)*, vol. 127, no. 8, pp. 1106–1125, 2019.

[11] Z. Huang, F. Zhang, X. Xu, Y. He, J. Yu, Z. Dong, Q. Ma, N. Chanpaisit, C. Si, Y. Jiang, Y. Wang, X. Chen, Y.-C. Chen, L. Wang, D. Lin, Y. Qiao, and Z. Liu, "Vbench++: Comprehensive and versatile benchmark suite for video generative models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 48, no. 3, pp. 3268–3285, 2026.

[12] Y. Liu, X. Cun, X. Liu, X. Wang, Y. Zhang, H. Chen, Y. Liu, T. Zeng, R. Chan, and Y. Shan, "Evalcrafter: Benchmarking and evaluating large video generation models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 22 139–22 149.

[13] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, and Z. Luo, "Llamafactory: Unified efficient fine-tuning of 100+ language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, 2024, pp. 400–410.

[14] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[15] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," in *International Conference on Learning Representations*, 2024.

[16] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel *et al.*, "Scaling rectified flow transformers for high-resolution image synthesis," in *Forty-first International Conference on Machine Learning*, 2024.

[17] X. Ma, Y. Wang, G. Jia, X. Chen, Z. Liu, Y.-F. Li, C. Chen, and Y. Qiao, "Latte: Latent diffusion transformer for video generation," *Transactions on Machine Learning Research*, 2025.

[18] J. Wang, H. Yuan, D. Chen, Y. Zhang, X. Wang, and S. Zhang, "Modelscope text-to-video technical report," *arXiv preprint arXiv:2308.06571*, 2023.

[19] H. Chen, Y. Zhang, X. Cun, M. Xia, X. Wang, C. Weng, and Y. Shan, "Videocrafter2: Overcoming data limitations for high-quality video diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7310–7320.

[20] Y. He, T. Yang, Y. Zhang, Y. Shan, and Q. Chen, "Latent video diffusion models for high-fidelity video generation with arbitrary lengths," *arXiv preprint arXiv:2211.13221*, 2022.

[21] Y. Wang, X. Chen, X. Ma, S. Zhou, Z. Huang, Y. Wang, C. Yang, Y. He, J. Yu, P. Yang *et al.*, "Lavie: High-quality video generation with cascaded latent diffusion models," *International Journal of Computer Vision*, vol. 133, no. 5, pp. 3059–3078, 2025.

[22] D. Zheng, Z. Huang, H. Liu, K. Zou, Y. He, F. Zhang, Y. Zhang, J. He, W.-S. Zheng, Y. Qiao, and Z. Liu, "VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness," *arXiv preprint arXiv:2503.21755*, 2025.

[23] T. Lee, M. Yasunaga, C. Meng, Y. Mai, J. S. Park, A. Gupta, Y. Zhang, D. Narayanan, H. Teufel, M. Bellagente *et al.*, "Holistic evaluation of text-to-image models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 69 981–70 011, 2023.

[24] K. Sun, K. Huang, X. Liu, Y. Wu, Z. Xu, Z. Li, and X. Liu, "T2v-compbench: A comprehensive benchmark for compositional text-to-video generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 8406–8416.

[25] Y. Shi, Y. Dong, Y. Ding, Y. Wang, X. Zhu, S. Zhou, W. Liu, H. Tian, R. Wang, H. Wang, Z. Liu, B. Zeng, R. Chen, Q. Wang, Z. Zhang, X. Chen, C. Tong, B. Li, Q. Liu, H. Wang, W. Yang, Y. Zhang, P. Wan, Y.-F. Zhang, and Z. Liu, "RealUnify: Do unified models truly benefit from unification? A comprehensive benchmark," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 22 488–22 497.

[26] K. Zou, Z. Huang, Y. Dong, S. Tian, D. Zheng, H. Liu, J. He, B. Liu, Y. Qiao, and Z. Liu, "Uni-MMMU: A massive multi-discipline multimodal unified benchmark," in *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026, pp. 908–924.

[27] J. Liu, X. Shuai, H. Ding, and Y.-G. Jiang, "Unison: Benchmarking unified multimodal models via synergistic understanding and generation," in *International Conference on Machine Learning (ICML)*, 2026.

[28] S. Jain, V. Keshava, S. Mysore Sathyendra, P. Fernandes, P. Liu, G. Neubig, and C. Zhou, "Multi-dimensional evaluation of text summarization with in-context learning," in *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 2023, pp. 8487–8495. [Online]. Available: http://dx.doi.org/10.18653/v1/2023.findings-acl.537

[29] C.-H. Chiang and H. yi Lee, "Can large language models be an alternative to human evaluations?" in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023. [Online]. Available: https://arxiv.org/abs/2305.01937

[30] J. Fu, S.-K. Ng, Z. Jiang, and P. Liu, "Gptscore: Evaluate as you desire," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024, pp. 6556–6576. [Online]. Available: https://aclanthology.org/2024.naacl-long.365/

[31] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 595–46 623, 2023.

[32] Q. Li, L. Cui, L. Kong, and W. Bi, "Exploring the reliability of large language models as customized evaluators for diverse nlp tasks," in *Proceedings of the 31st International Conference on Computational Linguistics*, 2025, pp. 10 325–10 344. [Online]. Available: https://aclanthology.org/2025.coling-main.688/

[33] J. Pan, Y. Zhang, N. Tomlin, Y. Zhou, S. Levine, and A. Suhr, "Autonomous evaluation and refinement of digital agents," in *Proceedings of the First Conference on Language Modeling*, 2024.

[34] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[35] D. Li, B. Jiang, L. Huang, A. Beigi, C. Zhao, Z. Tan, A. Bhattacharjee, Y. Jiang, C. Chen, T. Wu, K. Shu, L. Cheng, and H. Liu, "From generation to judgment: Opportunities and challenges of LLM-as-a-judge," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 2757–2791.

[36] R. You, H. Cai, C. Zhang, Q. Xu, M. Liu, T. Yu, Y. Li, and W. Li, "Agent-as-a-Judge," *arXiv preprint arXiv:2601.05111*, 2026.

[37] M. Zhuge, C. Zhao, D. R. Ashley, W. Wang, D. Khizbullin, Y. Xiong, Z. Liu, E. Chang, R. Krishnamoorthi, Y. Tian, Y. Shi, V. Chandra, and J. Schmidhuber, "Agent-as-a-Judge: Evaluate agents with agents," in *Proceedings of the 42nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 267, 2025, pp. 80 569–80 611.

[38] D. Brown, P. Balehannina, H. Jin, S. Havaldar, H. Hassani, and E. Wong, "Adaptively profiling models with task elicitation," in *Proceedings of the*

2025 Conference on Empirical Methods in Natural Language Processing, 2025, pp. 24 985–25 020.

[39] B. Gou, Z. Huang, Y. Ning, Y. Gu, M. Lin, W. Qi, A. Kopanev, B. Yu, B. Jiménez Gutiérrez, Y. Shu, C. H. Song, J. Wu, S. Chen, H. N. Moussa, T. Zhang, J. Xie, Y. Li, T. Xue, Z. Liao, K. Zhang, B. Zheng, Z. Cai, V. Rozgic, M. Ziyadi, H. Sun, and Y. Su, "Mind2Web 2: Evaluating agentic search with Agent-as-a-Judge," in *Advances in Neural Information Processing Systems*, vol. 38, 2025, pp. 192 393–192 451.

[40] C. Shen, Y. Hou, M. Pan, R. He, Z. H. Wong, M. Qiang, Z. Liu, H. Liang, P. Lai, Z. Sheng, and W. Zhang, "One-Eval: An agentic system for automated and traceable LLM evaluation," *arXiv preprint arXiv:2603.09821*, 2026.

[41] Y. Wang, X. Wang, Y. Yao, X. Li, X. Yang, Y. Teng, X. Ma, and Y. Wang, "AgenticEval: Toward agentic and self-evolving safety evaluation of large language models," in *Findings of the Association for Computational Linguistics: ACL 2026*, 2026, pp. 14 789–14 808.

[42] Z. Qin, D. Cheng, H. Wang, H. Yi, Y. Shao, Z. Fan, K. Li, and Q. Lao, "Evaluating hallucination in text-to-image diffusion models with scene-graph based question-answering agent," *arXiv preprint arXiv:2412.05722*, 2024.

[43] J. Wang, X. Yang, L. Wang, Z. Xu, Y. Wang, Y. Wang, W. Luo, K. Zhang, B. Hu, and M. Zhang, "A unified agentic framework for evaluating conditional image generation," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 12 626–12 646.

[44] T. Chen, Y. Zhang, Z. Zhang, P. Yu, S. Wang, Z. Wang, K. Lin, X. Wang, Z. Yang, L. Li, C.-C. Lin, J. Xie, O. Leong, L. Wang, Y. N. Wu, and M. Zhou, "EdiVal-Agent: An object-centric framework for automated, fine-grained evaluation of multi-turn editing," *arXiv preprint arXiv:2509.13399*, 2026.

[45] Y. Zhang, P. Du, B. Li, C. Wei, J. Miao, H. Zhang, S. Cai, Y. Wang, D. Jiang, Y. Zhang, P. Nie, W. Chen, C. Yu, and K. R. Allen, "RewardHarness: Self-evolving agentic post-training," *arXiv preprint arXiv:2605.08703*, 2026.

[46] Y. Yang, K. Fan, S. Sun, H. Li, A. Zeng, F. Han, W. Zhai, W. Liu, Y. Cao, and Z.-J. Zha, "VideoGen-Eval: Agent-based system for video generation evaluation," *arXiv preprint arXiv:2503.23452*, 2025.

[47] Y. Song, T. Pfister, and Y. Song, "VQQA: An agentic approach for video evaluation and quality improvement," *arXiv preprint arXiv:2603.12310*, 2026.

[48] Z. Zeng, Z. Wang, Y. Yu, H. Hua, and J. Luo, "VideoArgus: Agentic rubric-grounded unified evaluation for video generation and editing," *arXiv preprint arXiv:2608.05485*, 2026.

[49] X. H. Lù, A. Kazemnejad, N. Meade, A. Patel, D. Shin, A. Zambrano, K. Stańczak, P. Shaw, C. J. Pal, and S. Reddy, "AgentRewardBench: Evaluating automatic evaluations of web agent trajectories," in *Second Conference on Language Modeling*, 2025. [Online]. Available: https://openreview.net/forum?id=fQcUZMPIvu

[50] W. Shi, Y. Wang, Y. Zhao, Y. Chen, F. Feng, X. Hao, X. Su, Q. Gu, H. Su, X. Cai, and X. He, "AJ-Bench: Benchmarking Agent-as-a-Judge for environment-aware evaluation," in *Findings of the Association for Computational Linguistics: ACL 2026*, 2026, pp. 25 371–25 413.

[51] L. Wang, Y. He, P. Chen, A. Yehudai, Y. Liu, R. Ying, M. Shmueli-Scheuer, and A. Cohan, "Time to REFLECT: Can we trust LLM judges for evidence-based research agents?" *arXiv preprint arXiv:2605.19196*, 2026.

[52] Y. Zhu, T. Jin, Y. Pruksachatkun, A. Zhang, S. Liu, S. Cui, S. Kapoor, S. Longpre, K. Meng, R. Weiss, F. Barez, R. Gupta, J. Dhamala, J. Merizian, M. Giulianelli, H. Coppock, C. Ududec, A. Kellermann, J. Sekhon, J. Steinhardt, S. Schwettmann, A. Narayanan, M. A. Zaharia, I. Stoica, P. Liang, and D. Kang, "Establishing best practices in building rigorous agentic benchmarks," in *Advances in Neural Information Processing Systems*, vol. 38, 2025, pp. 184 435–184 475.

[53] S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, T. Ou, Y. Bisk, D. Fried *et al.*, "Webarena: A realistic web environment for building autonomous agents," in *Advances in Neural Information Processing Systems*, 2024.

[54] T. Xie, D. Zhang, J. Chen, X. Li, S. Zhao, R. Cao, T. J. Hua, Z. Cheng, D. Shin, F. Lei *et al.*, "Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments," in *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[55] J. Wang, H. Xu, J. Ye, M. Yan, W. Shen, J. Zhang, F. Huang, and J. Sang, "Mobile-agent: Autonomous multi-modal mobile device agent with visual perception," in *ICLR 2024 Workshop on Large Language Model Agents*, 2024.

[56] R. Kapoor, Y. P. Butala, M. Russak, J. Y. Koh, K. Kamble, W. Alshikh, and R. Salakhutdinov, "Omniact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web," in *Computer Vision – ECCV 2024*, 2024, pp. 161–178.

[57] S. Tian, Z. Zhang, L. Chen, and Z. Liu, "MMInA: Benchmarking multihop multimodal internet agents," in *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, pp. 13 682–13 697.

[58] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[59] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[60] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations*, 2023.

[61] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[62] B. Sel, A. Al-Tawaha, V. Khattar, R. Jia, and M. Jin, "Algorithm of thoughts: Enhancing exploration of ideas in large language models," in *International Conference on Machine Learning*, 2024.

[63] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk *et al.*, "Graph of thoughts: Solving elaborate problems with large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17 682–17 690.

[64] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in *International Conference on Learning Representations*, 2023.

[65] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[66] M. Li, Y. Zhao, B. Yu, F. Song, H. Li, H. Yu, Z. Li, F. Huang, and Y. Li, "Api-bank: A comprehensive benchmark for tool-augmented llms," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 3102–3116.

[67] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian *et al.*, "Toolllm: Facilitating large language models to master 16000+ real-world apis," in *International Conference on Learning Representations*, 2024.

[68] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[69] S. G. Patil, T. Zhang, X. Wang, and J. E. Gonzalez, "Gorilla: Large language model connected with massive apis," in *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[70] D. Jiang, Z. Guo, R. Zhang, Z. Zong, H. Li, L. Zhuo, S. Yan, P.-A. Heng, and H. Li, "T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot," *Advances in Neural Information Processing Systems*, vol. 38, pp. 39 856–39 890, 2026.

[71] Z. Huang, N. Yu, G. Chen, H. Qiu, P. Debevec, and Z. Liu, "VChain: Chain-of-visual-thought for reasoning in video generation," in *Annual Meeting of the Association for Computational Linguistics (ACL Findings), 2026*, 2026.

[72] V. Barres, H. Dong, S. Ray, X. Si, and K. Narasimhan, "τ^2-Bench: Evaluating conversational agents in a dual-control environment," *arXiv preprint arXiv:2506.07982*, 2025.

[73] T. Xue, W. Qi, T. Shi, C. H. Song, B. Gou, D. Song, H. Sun, and Y. Su, "An illusion of progress? assessing the current state of web agents," in *Second Conference on Language Modeling*, 2025. [Online]. Available: https://openreview.net/forum?id=6jZi4HSs6o

[74] G. Starace, O. Jaffe, D. Sherburn, J. Aung, J. S. Chan, L. Maksin, R. Dias, E. Mays, B. Kinsella, W. Thompson, J. Heidecke, A. Glaese, and T. Patwardhan, "PaperBench: Evaluating AI's ability to replicate AI research," in *Proceedings of the 42nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 267, 2025, pp. 56 843–56 873.

[75] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi, "Clipscore: A reference-free evaluation metric for image captioning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.