# Code-in-the-Loop Forensics: Agentic Tool Use for Image Forgery Detection

Fanrui Zhang[1,2†], Qiang Zhang[1†], Sizhuo Zhou[1,2], Jianwen Sun[2], Chuanhao Li[3], Jiaxin Ai[2], Yukang Feng[2], Yujie Zhang[2], Wenjie Li[2], Zizhen Li[2], Yifan Chang[1,2], Jiawei Liu[1*], and Kaipeng Zhang[2,3*]

[1] University of Science and Technology of China, China
[2] Shanghai Innovation Institute, China
[3] Shanghai Artificial Intelligence Laboratory, China
zfr888@mail.ustc.edu.cn, jwliu6@ustc.edu.cn, kp_zhang@foxmail.com

**Abstract.** Existing image forgery detection (IFD) methods either exploit low-level, semantics-agnostic artifacts or rely on multimodal large language models (MLLMs) with high-level semantic knowledge. Although naturally complementary, these two information streams are highly heterogeneous in both paradigm and reasoning, making it difficult for existing methods to unify them or effectively model their cross-level interactions. To address this gap, we propose ForenAgent, a multi-round interactive IFD framework that enables MLLMs to autonomously invoke, execute, and iteratively refine Python-based low-level tools around the detection objective, thereby achieving more flexible and interpretable forgery analysis. ForenAgent adopts a two-stage training pipeline with Cold Start and Reinforcement Fine-Tuning to progressively improve tool interaction and reasoning adaptability. We design a human-inspired dynamic reasoning loop with global perception, local focusing, iterative probing, and holistic adjudication, and implement it as both a data-sampling strategy and a task-aligned process reward. To support training and evaluation, we built FABench, a heterogeneous agent-forensics dataset with 100k images and about 200k agent-interaction question-answer pairs. Experiments show that ForenAgent exhibits emergent tool-use competence and reflective reasoning on challenging IFD tasks when assisted by low-level tools, charting a promising route toward general-purpose IFD. The code is available at https://github.com/zfr00/ForenAgent.

**Keywords:** Image forgery detection · Multimodal reasoning · Agent

## 1 Introduction

Advances in image editing and easy-to-use software have made low-cost manipulation and synthesis widely accessible. This growing democratization greatly boosts personal expression but also enables the malicious fabrication of multimedia

† Equal contribution.
* Corresponding author.

content. [36–39, 65]. As a result, Image Forgery Detection (IFD) has become a critical research frontier, essential for mitigating the societal risks of large-scale visual manipulation and preserving information integrity.

Researchers have proposed a wide range of deep learning-based image forgery detection methods, achieving strong performance across various benchmarks. Current approaches can be broadly categorized into two paradigms: (1) Low-level feature-based methods: These approaches identify forgeries by capturing non-semantic inconsistencies between manipulated and authentic regions, focusing on subtle visual artifacts. Depending on the characteristics of the forged image, a wide range of low-level cues, such as JPEG compression artifacts, edge discontinuities, and camera model traces, have been utilized to enhance forensic perception. Such methods embody careful algorithmic design and strong domain priors, offering interpretability and effectiveness in specific scenarios. However, depending solely on low-level inconsistencies restricts these methods to simple artifact patterns, making it difficult for them to handle diverse or subtle manipulation scenarios. (2) MLLMs-based approaches: Recently, Multimodal Large Language Models (MLLMs) have achieved significant progress on tasks requiring integrated visual and textual understanding [55]. Methods such as FakeShield [51] and SIDA [14] fine-tune MLLMs for IFD and demonstrate strong potential, benefiting from large-scale data to learn generalizable representations. Nonetheless, these approaches still exhibit several critical limitations: weak interaction with forensic tools, limited capability in fine-grained manipulation analysis, and insufficient transparency and controllability in sensitive scenarios. Fundamentally, these issues arise because their end-to-end learning paradigm does not encode structured forensic procedures or explicit tool-aware reasoning mechanisms.


**Fig. 1:** ForenAgent autonomously composes a global-to-local Python toolchain, accurately delivers a tampered verdict with precise localization of the forged region, and further demonstrates reflective self-correction by carefully revising an initially mislocalized crop to the appropriate region of interest.

Recent progress in MLLMs has shown that they are increasingly capable of complex reasoning and interaction with external tools [16, 59, 60]. However, extending this mechanism to image forensics remains challenging: Current MLLMs lack a dynamic framework that connects high-level semantic reasoning with the control and interpretation of diverse low-level forensic tools, making task-adaptive integration difficult. Moreover, designing a training paradigm that guides the

model toward logically consistent, self-directed reasoning and purposeful tool use rather than passive imitation remains an open challenge. Addressing these challenges is key to building truly interpretable and highly adaptive intelligent forensic systems.

In this work, we propose ForenAgent, an interactive multi-turn framework that enables MLLMs to autonomously invoke, execute, and iteratively refine Python-based low-level tools for IFD. To this end, we abstract and generalize common low-level forensics operators, including frequency residual, noise residual, and high-pass filtering, and package them into a toolbox of 12 utilities for future extension. For efficiency, only basic image processing operations, such as cropping, require code generation, while the remaining low-level forensics tools are exposed via direct tool calls. As illustrated in Figure 1, ForenAgent autonomously orchestrates Python tools to verify a forged image from global screening to local inspection, ultimately classifying it as tampered and accurately localizing the forged region. The agent further demonstrates reflective self-correction by recovering from an initially misfocused crop to the correct area of interest, an "aha moment" observed in IFD agents.

The development of ForenAgent involves two key components: (1) Forgery Agent Benchmark (FABench), a high-quality and heterogeneous forensic agent dataset constructed using state-of-the-art generative models (*e.g.*, GPT-4o [1], Nano-Banana [9], and Midjourney-v7 [28])). It contains 100k images (40k real, 30k synthetic, and 30k tampered) and serves as a comprehensive benchmark for training and evaluation in IFD. (2) A Cold-Start and Reinforcement Fine-Tuning (RFT) framework, designed to train MLLMs to function as reliable and autonomous agents. During the Cold-Start stage, ForenAgent adopts a self-exploration and experience-distillation paradigm. Specifically, GPT-4.1 [32] observes a large collection of forgery samples from FABench under system prompts that provide procedural guidance and executable code examples, distilling operational patterns into structured agent–interaction training data for initialization. During RFT, we abstract the human IFD workflow into four reasoning stages: global perception, local focusing, iterative probing, and holistic adjudication. Correspondingly, we design four Forgery Process Rewards that together form the overall tool reward. By incorporating these verifiable reward components into the reinforcement learning process, ForenAgent develops a more interpretable and systematic forensic reasoning mechanism, effectively integrating basic image processing with low-level forensic analysis in a coherent investigative workflow. This design enables ForenAgent to explore diverse reasoning strategies and optimize for long-term process quality rather than simply imitating predefined answers.

Extensive experiments demonstrate that ForenAgent significantly outperforms existing state-of-the-art IFD methods. Moreover, the model exhibits emergent multimodal reasoning behaviors such as visual search for forged regions, cross-region comparison, and even self-reflective correction. These intertwined reasoning patterns resemble human cognitive processes, contributing to stronger interpretability and forensic reliability for the IFD task. Our main contributions are summarized as follows:

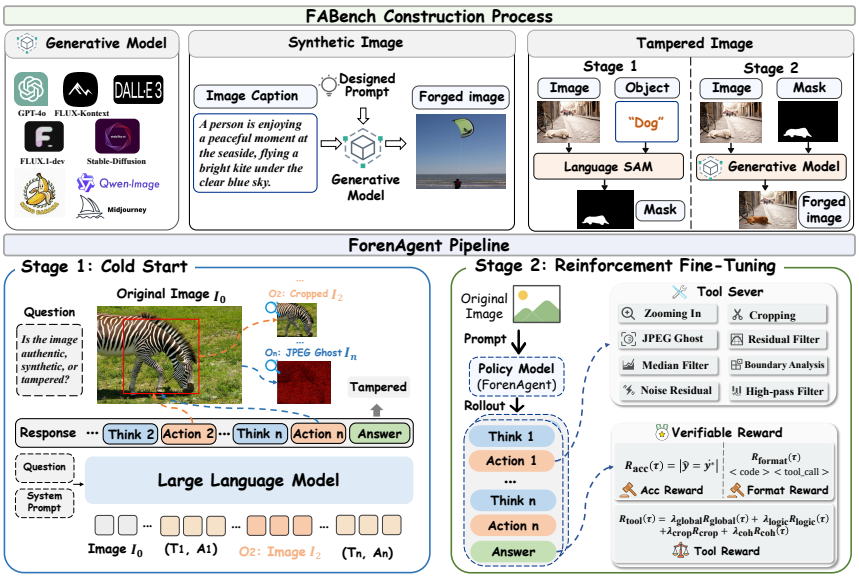


**Fig. 2:** The overall architecture of the ForenAgent is illustrated, with the upper part showing the FABench construction process and the lower part presenting the training pipeline of ForenAgent.

(1) We propose ForenAgent, a novel interactive, multi-turn framework that enables an MLLM to autonomously invoke, execute, and iteratively refine Python-based low-level tools for image forgery detection, thereby taking the first step toward intelligent, tool-augmented IFD systems. (2) We construct FABench, a large-scale, high-quality, and heterogeneous forensic agent dataset comprising 100k images and 200k interactive QA pairs, focusing on forgery detection from cutting-edge generative models. (3) We formulate a dynamic reasoning loop comprising global perception, local focusing, iterative probing, and holistic adjudication into a data-sampling strategy and task-aligned process reward that foster flexible, tool-adaptive evidence reasoning and enable robust, interpretable decisions.

## 2 Related Work

### 2.1 Image Forgery Detection

Early IFD research primarily relies on low-level feature extractors that model non-semantic inconsistencies between manipulated and authentic regions. Typical frequency and residual cues include FFT-based high-frequency modeling in FreqNet [42], SRM-filtered noise residuals in RGB-N [27], multi-scale high-frequency noise suppression in HFF [27], and transformation-driven feature learning in IT-Detector [20]. Related signals further exploit camera fingerprints (PRNU) [58], inconsistent compression (JPEG ghost) [35], DCT high-pass localization (ObjectFormer) [46], and boundary-sensitive operators (MVSS-Net, HiFi-Net) [8, 11].

Additional work broadens these cues and improves robustness [4, 6, 23]. ST-Trace analyzes images through spatial-temporal evolution patterns [49]. Despite progress, generalization remains challenging: Chameleon highlights failures on high-quality synthetic images [52], while Effort [53] improves robustness by separating content from artifact features, better isolating forgery traces. More recently, vision-language models and LLM-augmented frameworks have been introduced to incorporate semantic reasoning and natural-language explanations. DD-VQA [57] enables fine-tuning BLIP on tampering data to improve both detection and explanations; FakeShield [51] and ForgeryGPT [24] leverage LLMs for multimodal understanding and interactive, explainable analysis. For synthetic image detection, FakeScope [21] and LEGION [17] further demonstrate the promise of MLLMs, while SIDA [14] and So-Fake-R1 [15] unify tampered and synthetic detection in a multi-class setting to probe MLLM capabilities. Overall, the field is moving from artifact-centric detection toward multimodal reasoning with improved interpretability. AIGI-Holmes [62] employs MLLMs for explainable AI-generated image detection, unifying high-level reasoning with fine-grained artifact analysis to boost generalization. Loki [54] establishes a comprehensive benchmark for synthetic data detection, exposing the disparity between MLLMs' high-level reasoning and fine-grained artifact perception. In line with this, our work also focuses on comprehensive detection across both tampered and synthetic images.

### 2.2 Thinking with Images

The "Thinking with Images" paradigm is pushing MLLMs beyond passive visual description toward interactive and iterative agents that can actively manipulate visual inputs and refine their reasoning processes [64]. Early work typically relied on predefined CoT formats and static toolsets (*e.g.*, VisProg [12] and ViperGPT [41]) to prompt models to invoke fixed tools for specific vision tasks. However, such designs often limit flexibility and generalization to unseen or complex scenarios. To address this limitation, METATOOL [48] introduces meta-task augmentation to improve tool mastery and transferability. In contrast, PyVision [59] enables dynamic Python code generation to flexibly invoke and compose complex tools for more versatile visual reasoning. Pushing autonomy further, recent studies optimize tool use through reinforcement learning (RL) [26]. DeepEyes [60] performs end-to-end RL with tool-oriented data selection and reward design, while DeepEyesV2 [13] advances multimodal reasoning toward a more agentic paradigm, achieving state-of-the-art results on perception- and search-intensive tasks in RealX-Bench. Thyme [56] shifts the focus from static image perception to comprehensive reasoning by integrating temporal and logical context. V-TOOLRL [40] directly maximizes task success based on tool-interaction feedback, and ReVPT [61] adopts a two-stage scheme with multiple visualization tools to enhance general multimodal capabilities. Collectively, these methods demonstrate a transition from fixed tool pipelines toward adaptive visual agents that learn more effective tool-use strategies through interaction and feedback. Despite significant progress in numerous domains, exploration within the specialized field of image forgery

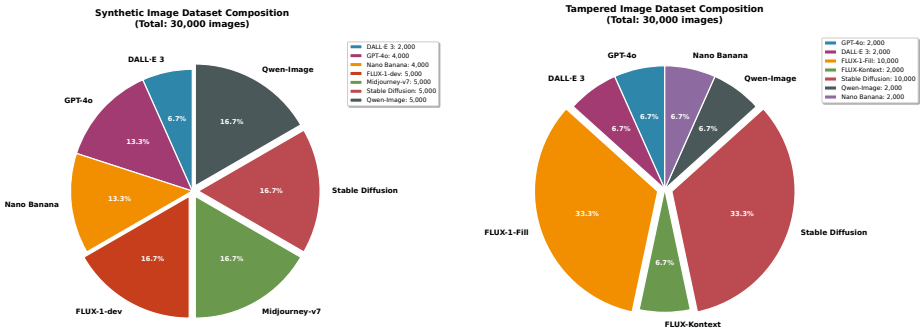


**Fig. 3:** The composition of the FABench training set.

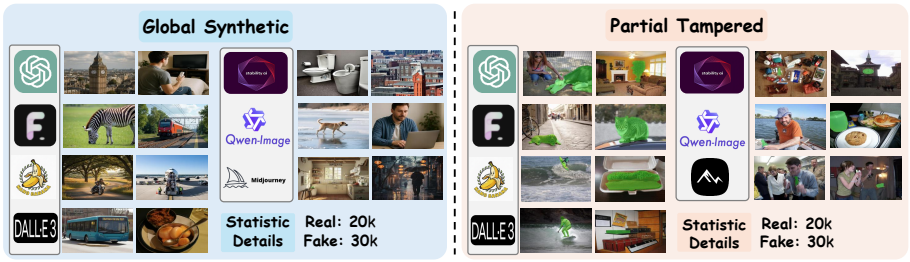


**Fig. 4:** Examples of tampered and synthetic images from diverse FABench generators.

detection remains nascent. To address this gap, we propose ForenAgent, a forensic agent based on a "Code-in-the-Loop" design. ForenAgent actively generates and iteratively optimizes low-level forensic analysis code, thereby achieving accurate detection of forged images.

## 3 Method

In this section, as shown in Figure 2, we first introduce FABench, a comprehensive benchmark consisting of multi-type, high-difficulty forgery images. We then describe ForenAgent's two-stage training framework.

### 3.1 FABench

Recent advances in generative AI have enabled the easy creation of sophisticated synthetic and tampered content, while existing detection datasets exhibit notable limitations: (i) Outdated synthetic content. Benchmarks grounded in early GANs (*e.g.*, StyleGAN [18]) mainly contain low-fidelity generations that are significantly easier than modern photorealistic outputs (*e.g.*, GPT-4o-image [1], Midjourney-v7 [28]). (ii) Fixed tampering pipelines. Many datasets rely on a narrow set of inpainting models (*e.g.*, Stable Diffusion [34]) and rarely explore newer pipelines

(*e.g.*, FLUX-Kontext [19], Qwen-image [50]), limiting heterogeneity and inducing repeated artifacts. We build FABench via a strict, modular pipeline designed to maximize diversity across contemporary generators. FABench contains authentic, synthetic, and tampered images to reflect open-world scenarios and comprehensively evaluate forensic reasoning. Beyond simple objects and portraits, it also covers a wide range of naturalistic scenes:

Authentic (40k): COCO [22] images spanning a broad spectrum of real-world scenes and everyday contexts.

Synthetic (30k): Two-step pipeline (as shown in Figure 2): caption enrichment (30k COCO images; GPT-4o-mini generates detailed, compositional captions), followed by image synthesis with a diverse set of generators to maximize architectural diversity and realism.

Tampered (30k): Starting from COCO sources with instance masks, we derive object masks via SAM-guided text prompts [22], then perform object-level inpainting using (i) strict-mask models (FLUX-1-Fill, Stable Diffusion; inputs: image/mask/prompt) and (ii) soft/no-mask models (*e.g.*, GPT-4o-image, Qwen-Image), where the edited regions are composited back into the original image to suppress unintended global micro-changes. The comprehensive construction pipeline and detailed analysis are provided in the Appendix.

We adopt a multi-stage pipeline consisting of quality validation (file integrity, mask legality, etc.) and deduplication, followed by stratified human auditing to filter low-quality samples. Samples that fail inspection are removed, while borderline cases are re-synthesized or re-inpainted accordingly. For the tampered split, we construct a 700-image tampered test set using seven generators: GPT-4o, DALL·E 3 [30], FLUX-1-Fill, FLUX-Kontext, Stable Diffusion, Qwen-Image, and Nano Banana [9]. Similarly, for the synthetic split, we generate a 700-image synthetic test set using GPT-4o, DALL·E 3, FLUX-1-dev, Midjourney-v7, Stable Diffusion, Qwen-Image, and Nano Banana. For the authentic split, we randomly sample 700 real images from COCO to form the authentic test set.

We visualize the composition of the FABench training set in Figure 3. For synthetic images, the dataset includes samples generated by GPT-4o (4 k), DALL·E 3 (2 k), FLUX-1-dev (5 k), Midjourney-v7 (5 k), Stable Diffusion (5 k), Qwen-Image (5 k), and Nano Banana (4 k). For tampered images, it includes GPT-4o (2 k), DALL·E 3 (2 k), FLUX-1-Fill (10 k), FLUX-Kontext (2 k), Stable Diffusion (10 k), Qwen-Image (2 k), and Nano Banana (2 k). This diverse composition demonstrates that FABench integrates a wide spectrum of generation and inpainting paradigms, covering both text-to-image and mask-based pipelines. Figure 4 further showcases examples of tampered and synthetic images produced by different generators in FABench. We observe that advanced models, such as Nano Banana and Qwen-Image, produce more photorealistic and harder-to-discriminate forgeries.

### 3.2 ForenAgent

ForenAgent enables MLLMs to dynamically invoke and execute low-level tools throughout their reasoning process. In each session, the MLLM receives the input

and responds with either executable Python code or structured tool calls, which are run in an isolated Python sandbox. The resulting outputs, including text and or visualizations, are then fed back into the MLLM context, allowing iterative multi-turn refinement until the final prediction is produced.

**Tool Boxes** ForenAgent adopts Python code and tool calls as the fundamental primitives for tool construction. Accordingly, we categorize the tools into two major types as follows:

**(1) Basic Image Processing:** These two categories of tools form the basis for visual manipulation and perception. They enable the agent to clean, align, and highlight image content to improve downstream reasoning.

Cropping: For high-resolution or cluttered inputs, the agent typically crops and zooms into regions of interest. By reasoning about the coordinates, it effectively performs soft object localization and forensic analysis, directing attention to the most informative regions.

Enhancement: In visually subtle domains like tampered imaging, the agent autonomously applies adaptive contrast adjustments and multi-scale enhancements to amplify latent structures and imperceptible forensic traces, making them significantly more prominent.

**(2) Low-Level Forensics Tools:** Based on the related work, we constructed a candidate pool of 12 low-level, code-based forensic methods. The agent can generate and deploy these tools as needed. We categorize them as follows:

(a) Frequency Domain Analysis: Tools that analyze artifacts in transformed domains. (1) FFT High-Frequency Residual: Emphasizes forgery boundaries and texture anomalies in the frequency domain. (2) DWT High-Frequency Subbands: Uses wavelet decomposition to reveal high-frequency differences from synthesis or upsampling. (3) Resampling Periodicity: Detects spectral peaks introduced by interpolation (scaling/rotation). (4) DCT-based High-Pass Filter: Extracts high-frequency components to highlight edges and tampering traces.

(b) Noise & Residual Analysis: Tools that extract subtle noise patterns typically suppressed by image content. (5) SRM: Uses a bank of high-pass and directional filters to extract robust noise residuals. (6) Bayar Constrained Convolution: Employs a specific convolutional kernel to suppress image content and amplify manipulation traces. (7) PRNU (Photo-Response Non-Uniformity): Extracts the camera sensor's unique fingerprint noise to find local inconsistencies (splices) via block correlation.

(c) Edge & Boundary Analysis: Methods to pinpoint inconsistent edges or gradients. (8) Sobel Edge Detector: Identifies splicing boundaries or anomalous edge patterns. (9) Laplacian High-Pass: Extracts high-frequency components to detect tampering artifacts.

(d) Specific Artifact Detection: Tools targeting the byproducts of distinct manipulations. (10) JPEG Ghost: Detects recompression artifacts by analyzing the error layer difference between multiple compression qualities. (11) Median Filtering Traces: measures artifacts and suspicious smoothing patterns.

(e) Statistical Analysis: (12) Local Correlation Map: Quantifies enhanced correlations within pixel neighborhoods, often indicative of manipulation.

For efficiency, only basic image processing operations such as cropping require Python code generation and execution, while the remaining low level forensics tools are exposed via direct tool calls.

**Cold Start** The training process for ForenAgent consists of two sequential stages, designed to progressively equip the MLLM with the capabilities to handle complex image forgery detection tasks.

**System Prompt Design**: To steer the MLLM's reasoning, code generation, and low-level tool usage, ForenAgent employs a carefully designed system prompt in addition to user queries. The prompt defines how to access inputs, invoke tools, structure code, and return results: (i) prefer executable code or tool calls over free-form text; (ii) preload images or frames as `image_clue_i` (with resolution metadata) to enable direct referencing for operations such as cropping; (iii) standardize outputs via `print(...)` for text and `plt.show()` for visualizations; (iv) wrap each code block with `<code>...</code>` for reliable parsing, and each tool invocation with `<tool_call>...</tool_call>` to enable structured execution; (v) place the final class token inside `<answer>...</answer>` for consistent evaluation. This design produces parsable and executable outputs with reduced runtime errors. The full system prompt is provided in the Appendix.

**Correct Reasoning Trajectories**: Built on FABench, we curate a long-horizon, multi-turn instruction-tuning set for IFD to inject domain reasoning and long-CoT skills into open-source MLLMs. For each sample, we provide the System Prompt, question, and images to GPT-4.1 to obtain an authenticity judgment and a multi-step chain. We retain a response only if: (1) the predicted label is correct; (2) It includes a code sandbox that enforces the enclosure of executable Python within `<code>...</code>` and tool invocations within `<tool_call>...</tool_call>`; (3) for tampered cases, the answer explicitly names the forged object. The filtered subset, containing approximately 200k agent–interaction question–answer pairs, is used for supervised Cold-Start tuning.

**Reinforcement Fine-Tuning** In this section, we investigate how MLLMs acquire tool-calling and reasoning capabilities without the need for supervised labels, leveraging pure RL for continuous self-improvement. End-to-end, outcome-rewarded RL jointly optimizes textual CoT and action planning over full trajectories. The agent interacts for multiple turns until producing an answer or exhausting the tool-call budget. States interleave text tokens $X$ and image tokens $I$; all observation tokens are inputs only and do not contribute to the loss.

**Group Relative Policy Optimization (GRPO)**: With GRPO [10], we sample a candidate set for each input, compute relative rewards through within-group normalization, and update the policy without a separate critic network. By employing clipped importance weights alongside a KL divergence penalty relative to a reference model, we stabilize the training process while effectively distilling preference signals from both model-generated and human-annotated answers.

**Reward Modeling**: In addition to the correctness reward $R_{\text{acc}}(\tau)$ and the format reward $R_{\text{format}}(\tau)$ that ensures the use of valid `<code>`,`<tool_call>` and `<answer>` tags, we introduce a tool usage reward $R_{\text{tool}}$ to evaluate how effectively the model applies external tools. The tools are categorized into Basic Image Processing $\mathcal{T}_{\text{basic}}$ and Low-Level Forensics $\mathcal{T}_{\text{low}}$. The reward $R_{\text{tool}}(\tau)$ integrates four components to assess the logical use of these tools.
*(i) Global Forensic Priority ($R_{global}$):* Encourages the model to first apply low-level forensic tools for global image analysis before using basic image processing for local operations. $T$ represents the total number of interaction turns. Let $t$ denote the index of the current reasoning step and $a_t$ represent the action. Define the first-use steps:

$$\begin{aligned} t_{\text{low}} &= \min\{t : a_t \in \mathcal{T}_{\text{low}}\}, \\ t_{\text{basic}} &= \min\{t : a_t \in \mathcal{T}_{\text{basic}}\}. \end{aligned} \tag{1}$$

The global priority reward is:

$$R_{\text{global}}(\tau) = [t_{\text{low}} < t_{\text{basic}}] \cdot \gamma^{t_{\text{low}}-1}, \quad \gamma \in (0,1). \tag{2}$$

*(ii) Tool Logic ($R_{logic}$):* This component motivates the model to optimize its behavior by rewarding syntactically correct and logically coherent tool invocations.
*(iii) Crop Sensitivity ($R_{crop}$):* Reward a single occurrence of `Crop` with class-specific weights. Define the indicator

$$\mathbb{I}_{\text{crop}} \;=\; \mathbb{1}\{\exists\, t \in \{1,\dots,T\} :\; a_t = \texttt{Crop}\}\,, \tag{3}$$

$$R_{\text{crop}}(\tau) = \begin{cases} b_{\text{tamper}}\, \mathbb{I}_{\text{crop}}, & \text{if } \hat{y} = \texttt{tampered}, \\ b_{\text{auth}}\, \mathbb{I}_{\text{crop}}, & \text{if } \hat{y} = \texttt{authentic}, \\ b_{\text{syn}}\, \mathbb{I}_{\text{crop}}, & \text{if } \hat{y} = \texttt{synthetic}. \end{cases} \tag{4}$$

*(iv) Reasoning Coherence ($R_{coh}$):* Reward a "locate-then-investigate" pair once (at most): a low-level tool immediately after `Crop` that consumes its output. Let

$$\begin{aligned} R_{\text{coh}} = \mathbb{1}\Big\{ &\exists\, t \in \{1,\dots,T-1\} :\; a_t = \texttt{Crop}, \\ &a_{t+1} \in \mathcal{T}_{\text{low}},\; \text{chain}(a_t, a_{t+1}) \Big\}. \end{aligned} \tag{5}$$

The tool usage reward aggregates the four sub-rewards:

$$\begin{aligned} R_{\text{tool}}(\tau) = \lambda_{\text{global}} R_{\text{global}}(\tau) + \lambda_{\text{logic}} R_{\text{logic}}(\tau) \\ + \lambda_{\text{crop}} R_{\text{crop}}(\tau) + \lambda_{\text{coh}} R_{\text{coh}}(\tau). \end{aligned} \tag{6}$$

Finally, the overall reward function $R$ is defined as:

$$\begin{aligned} R(\tau) = \lambda_{\text{acc}} \cdot R_{\text{acc}}(\tau) + \lambda_{\text{format}} \\ \cdot R_{\text{format}}(\tau) + \lambda_{\text{tool}} \cdot R_{\text{tool}}(\tau). \end{aligned} \tag{7}$$

**Table 1:** Comparison of ForenAgent with state-of-the-art methods on FABench, reporting per-class and overall performance across three categories.

| Methods | Authentic | | | | Synthetic | | | | Tampered | | | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | Prec | Rec | F1 | Acc | Prec | Rec | F1 | Acc | Prec | Rec | F1 | Acc | F1 |
| Gemini2.5-flash [44] | 47.2 | 36.2 | 98.6 | 52.9 | 75.2 | 88.2 | 38.5 | 53.6 | 68.2 | 75.0 | 3.9 | 7.4 | 45.3 | 38.0 |
| Gemini2.5-Pro [44] | 46.9 | 38.3 | 96.9 | 54.9 | 72.6 | 83.2 | 22.0 | 34.8 | 70.0 | 74.8 | 15.3 | 25.4 | 44.7 | 38.4 |
| GPT-4o [1] | 46.6 | 38.1 | 96.3 | 65.8 | 72.5 | 83.5 | 21.7 | 34.5 | 69.8 | 71.8 | 15.3 | 25.4 | 44.4 | 38.1 |
| GPT-o3-mini [33] | 46.6 | 38.1 | 96.0 | 54.5 | 72.5 | 83.5 | 21.7 | 34.5 | 69.9 | 72.4 | 15.7 | 25.8 | 44.5 | 38.3 |
| GPT-4.1 [32] | 54.0 | 41.1 | 95.9 | 57.5 | 80.0 | 90.5 | 46.6 | 61.5 | 68.8 | 65.3 | 13.0 | 21.6 | 51.4 | 46.9 |
| GPT-5 [31] | 46.5 | 38.1 | 96.3 | 54.6 | 72.6 | 83.6 | 21.9 | 34.7 | 69.8 | 73.1 | 15.1 | 25.1 | 44.5 | 38.1 |
| InternVL3-78B [63] | 46.7 | 37.3 | 87.9 | 52.4 | 66.0 | 46.0 | 12.1 | 19.2 | 67.9 | 54.9 | 20.9 | 30.2 | 40.3 | 33.9 |
| Qwen2.5-VL-72B [3] | 63.3 | 47.4 | 90.7 | 62.3 | 70.3 | 56.4 | 48.3 | 52.0 | 66.3 | 47.8 | 11.0 | 17.9 | 50.0 | 44.1 |
| QVQ-72B-preview [45] | 59.5 | 43.3 | 69.3 | 53.3 | 66.4 | 49.4 | 38.1 | 43.1 | 65.2 | 46.6 | 29.3 | 36.0 | 45.6 | 44.1 |
| InternVL2.5-78B-MPO [47] | 60.1 | 45.0 | 88.4 | 59.6 | 64.1 | 44.4 | 30.1 | 35.9 | 62.0 | 30.2 | 10.7 | 15.8 | 43.1 | 37.1 |
| Qwen3-VL-30B [2] | 55.8 | 42.4 | 90.6 | 57.7 | 65.4 | 45.5 | 19.7 | 27.5 | 71.7 | 67.7 | 29.0 | 40.6 | 46.4 | 41.9 |
| GPT-4.1 (with Tools) [32] | 77.8 | 65.5 | 70.4 | 67.9 | 75.1 | 64.1 | 57.9 | 60.8 | 76.4 | 64.3 | 65.7 | 65.0 | 64.7 | 64.6 |
| Qwen2.5-VL-72B (with Tools) [3] | 74.2 | 61.5 | 60.1 | 60.8 | 71.2 | 56.7 | 57.4 | 57.1 | 69.5 | 54.2 | 54.7 | 54.4 | 57.4 | 57.4 |
| Gemini2.5-Pro (with Tools) [44] | 71.3 | 57.0 | 57.1 | 57.1 | 69.2 | 54.1 | 50.9 | 52.4 | 68.3 | 52.3 | 55.3 | 53.8 | 54.4 | 54.4 |
| Qwen2.5-VL-7B [3] | 86.2 | 80.2 | 78.0 | 79.1 | 86.1 | 79.5 | 78.4 | 78.9 | 87.1 | 79.5 | 82.7 | 81.1 | 79.7 | 79.9 |
| Qwen3-VL-8B [2] | 80.1 | 66.2 | 81.9 | 73.2 | 86.9 | 88.7 | 69.6 | 78.0 | 85.2 | 78.4 | 76.9 | 77.6 | 76.1 | 76.3 |
| Gram-Net [25] | 75.7 | 58.4 | 94.4 | 72.2 | 74.6 | 67.2 | 46.3 | 54.8 | 75.5 | 69.1 | 48.0 | 56.7 | 62.9 | 61.2 |
| SIDA [14] | 86.6 | 74.8 | 90.3 | 81.8 | 81.8 | 74.4 | 69.1 | 71.7 | 85.1 | 82.0 | 70.7 | 75.9 | 76.7 | 76.5 |
| AIGI-Holmes [62] | 81.0 | 71.2 | 72.4 | 71.8 | 80.1 | 68.1 | 75.9 | 71.8 | 83.2 | 78.6 | 68.3 | 73.1 | 72.2 | 72.1 |
| LGrad [43] | 86.8 | 80.1 | 80.4 | 80.3 | 86.5 | 80.1 | 79.0 | 79.6 | 83.5 | 75.0 | 75.7 | 75.3 | 78.4 | 78.4 |
| LNP [5] | 80.1 | 70.8 | 68.4 | 69.6 | 76.0 | 63.0 | 67.6 | 65.2 | 82.8 | 75.2 | 72.1 | 73.6 | 69.4 | 69.5 |
| Effort [53] | 89.9 | 81.7 | 89.7 | 85.5 | 86.1 | 78.7 | 79.9 | 79.3 | 86.4 | 83.4 | 74.0 | 78.4 | 81.2 | 81.1 |
| DDA [7] | 87.0 | 76.4 | 88.1 | 81.8 | 84.0 | 77.0 | 74.4 | 75.7 | 85.0 | 81.3 | 71.4 | 76.0 | 78.0 | 77.8 |
| UnivFD [29] | **95.3** | **90.5** | **96.1** | **93.2** | 82.1 | 75.8 | 68.3 | 71.8 | 84.8 | 76.3 | 79.0 | 77.6 | 81.1 | 80.9 |
| **ForenAgent** | 94.0 | 87.9 | 95.3 | 91.4 | **93.6** | **91.3** | **89.4** | **90.3** | **91.3** | **89.5** | **83.7** | **86.5** | **89.5** | **89.4** |

By incorporating these verifiable reward components into the reinforcement learning process, ForenAgent achieves more interpretable and systematic reasoning for IFD, effectively learning to leverage both basic image processing and low-level forensic analysis tools in a coherent investigative workflow.

## 4 Experiments

### 4.1 Experimental Setup

**Baselines.** We compare ForenAgent against 24 baselines across four groups: (1) closed-source MLLMs evaluated zero-shot without fine-tuning, (2) large-scale open-source MLLMs evaluated zero-shot without fine-tuning, (3) tool-augmented MLLMs equipped with our toolbox, and (4) trained baselines, including MLLMs trained on the same image set as ForenAgent and representative state-of-the-art forgery detectors. Detailed baseline descriptions and configurations are provided in the *Supplementary Materials*. Notably, Our evaluation adopts a more challenging three-class protocol; most conventional detectors are binary by design and thus cannot be directly applied without substantial modification.

**Implementation Details.** All experiments are conducted with PyTorch on eight NVIDIA Tesla H200 GPUs; we adopt Qwen2.5-VL-7B as the base MLLM,

**Table 2:** Overall accuracy (%) and F1-score comparison with state-of-the-art methods on the SIDA-Test dataset.

| Methods | Accuracy | F1-score |
|---|---|---|
| Qwen2.5-VL-7B [3] | 72.9 | 69.9 |
| Qwen3-VL-8B [2] | 68.7 | 65.5 |
| Gram-Net [25] | 53.4 | 55.0 |
| SIDA [14] | 77.2 | 77.1 |
| Effort [53] | 78.5 | 78.4 |
| DDA [7] | 74.3 | 72.9 |
| LGrad [43] | 64.5 | 64.5 |
| LNP [5] | 53.3 | 53.2 |
| UnivFD [29] | 61.1 | 60.9 |
| **ForenAgent** | **83.1** | **82.9** |

**Table 3:** Evaluation of the influence of different components of ForenAgent on the FABench dataset.

| Methods | Accuracy | F1-score |
|---|---|---|
| w/o Cold Start | 78.7 | 76.9 |
| w/o RFT | 81.4 | 81.3 |
| w/o Tool Reward | 83.8 | 82.9 |
| w/o global | 86.4 | 86.3 |
| w/o logic | 84.4 | 84.3 |
| w/o crop | 87.8 | 86.5 |
| w/o coh | 88.0 | 87.9 |
| with InternVL3-8B | 79.8 | 79.9 |
| with LLaVA-7B | 75.8 | 75.5 |
| **ForenAgent** | **89.5** | **89.4** |

perform full-parameter finetuning in the Cold-Start stage with a learning rate of 1e-5 for two epochs using AdamW and a cosine-annealing scheduler with a maximum context length of 100k tokens, and then run RFT with GRPO on Qwen2.5-VL-7B for 80 iterations, sampling 256 prompts per batch with eight rollouts per prompt and at most seven tool calls and capping the response length at 20,480 tokens. More details are provided in the *Supplementary Materials*.

**Evaluation Metrics.** Following prior work [14], we evaluate detection at the image level using Accuracy and F1.

### 4.2 Detection Evaluation

As shown in Table 1, ForenAgent achieves state-of-the-art performance across all test splits, consistently outperforming baselines in both synthetic and tampered categories. In contrast, existing closed and open-source MLLMs struggle in zero-shot settings by frequently misclassifying manipulated images as authentic. This highlights a critical deficit of IFD-specific knowledge in their pretraining corpora. Notably, equipping models such as Gemini2.5-Pro and GPT-4.1 with specialized forensic tools consistently improves performance, underscoring the potential of tool-augmented IFDL. After supervised training, Qwen2.5-VL-7B outperforms Qwen3-VL-8B, supporting our choice of Qwen2.5-VL as the backbone. UnivFD achieves the best accuracy on the authentic class, suggesting strong capability in identifying unmanipulated images, but it struggles to distinguish between tampered and synthetic cases. Overall, these results underline the robustness of ForenAgent across manipulation types and domains, demonstrating its potential as a general-purpose and high-performance solution for real-world IFD.

### 4.3 Generalization

To assess generalization, we further evaluate ForenAgent on the SIDA-Test dataset in Table 2. Under identical training data, we compare ForenAgent with

other methods. ForenAgent achieves the highest scores, demonstrating its strong adaptive capacity. The *Supplementary Materials* further explore the limitations of current MLLMs in using bounding boxes for forgery localization and detail the optimal pipeline for adapting ForenAgent to pixel-level precision.

### 4.4 Ablation Study

As shown in Table 1, ForenAgent substantially outperforms the trained Qwen2.5-VL-7B baseline, confirming that its Python-based low-level toolchain leads to more accurate and robust solutions for IFD. Table 3 further presents ablation studies evaluating the contribution of each training stage. Removing the Cold-Start stage (*w/o Cold Start*) noticeably degrades reasoning quality, while removing RFT (*w/o RFT*), which uses our verifiable reward under GRPO, significantly harms final prediction accuracy. In addition, we verify the effectiveness of the tool reward. Removing it during RFT (*w/o Tool Reward*) weakens the model's incentive to utilize tools properly and leads to performance degradation. Overall, these findings highlight the critical role of our staged training and reward design in progressively enhancing reasoning capability.

We also investigate the impact of specific reward components and backbone architectures. Systematic exclusion of the four reward functions: Global Forensic Priority ($R_{\text{global}}$), Tool Logic ($R_{\text{logic}}$), Crop Sensitivity ($R_{\text{crop}}$), and Reasoning Coherence ($R_{\text{coh}}$). Notably, the *w/o logic* variant exhibits the most significant drop in both accuracy and F1-score, suggesting that maintaining syntactically and logically correct tool usage is the foundation of agent efficacy. The degradation in *w/o global* and *w/o coh* further validates our "locate-then-investigate" reasoning chain. Moreover, replacing the default architecture with InternVL (*with InternVL3-8B*) or LLaVA (*with LLaVA-7B*) results in substantial performance loss, which confirms that our selected backbone is better suited for the IFD task. Collectively, these findings underscore the critical synergy between our staged training, structured reward design, and backbone selection.

### 4.5 Visualization

Figure 5 illustrates a successful synthetic-image case where ForenAgent constructs a coherent evidence chain mirroring human reasoning. The process progresses through global perception, local focusing, iterative probing, and holistic adjudication to deliver accurate detection and a convincing explanation. It first applies the Bayar filter for global scanning to identify initial suspicion based on global texture uniformity and lack of photographic noise. The agent then conducts local focusing on a representative region and iteratively probes with DCT high-pass filtering and PRNU residual analysis. Finally, the agent synthesizes multi-scale forensic cues, concluding that the hyper-realistic smooth rendering and lack of random noise point to a fully synthetic origin.

### 4.6 Tool Usage Analysis

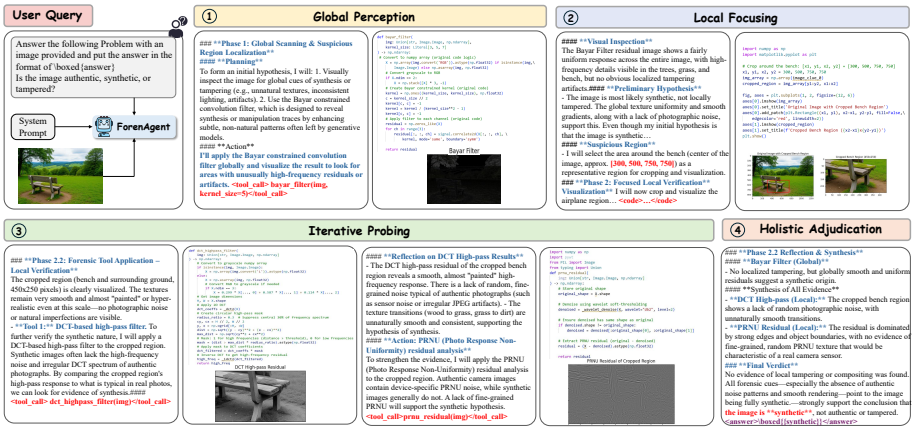


**Fig. 5:** The evidence chain by which ForenAgent correctly identifies a synthetic image.

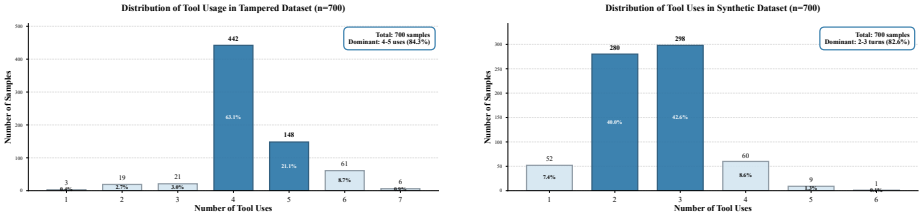


**Fig. 6:** Distribution of tool invocation frequencies for ForenAgent across FABench.

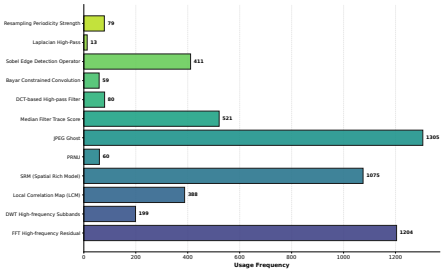


**Fig. 7:** Quantitative distribution of low-level forensic tool usage frequencies across the comprehensive FABench dataset.

Figure 6 analyzes tool usage frequencies on the FABench test sets. For synthetic images, ForenAgent typically converges within about 3 tool calls, whereas tampered images require around 4 calls due to higher structural complexity and the need to localize manipulated regions. This suggests that ForenAgent dynamically adapts its reasoning workflow to task difficulty, substantially improving efficiency. Figure 7 summarizes the usage distribution across low-level forensic tools: SRM, FFT, and JPEG Ghost are used most frequently, while Laplacian High-Pass appears less often. This pattern suggests that ForenAgent learns an adaptive tool-selection policy conditioned on image characteristics, rather than relying on mechanical tool enumeration. Notably, many classical low-level forensic tools are revived and integrated into our pipeline, offering a new perspective on combining traditional

image forgery detection techniques with modern MLLMs and potentially inspiring future hybrid-agent designs. Detailed evaluations of computational efficiency and robustness are discussed in the *Supplementary Materials*.

## 5 Conclusion

In this paper, we introduced ForenAgent, an interactive multi-round framework that enables MLLMs to autonomously invoke and iteratively refine Python-based low-level tools for IFD. We abstract and generalize key low-level tools in IFD, forming a 12-tool forensic toolbox for future community extension. Through a two-stage training pipeline of Cold Start and Reinforcement Fine-Tuning, ForenAgent learns a dynamic reasoning process from global perception to holistic adjudication. Experimental results on FABench and SIDA-Test demonstrate its superior interpretability, robustness, and reflective tool-use capability across diverse forgery scenarios. Our work marks an important first step toward building intelligent agent systems for image forensics.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62476260, the Fundamental Research Funds for the Central Universities under Grant WK2100000057.

## References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
3. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
4. Bayar, B., Stamm, M.C.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: Proceedings of the 4th ACM workshop on information hiding and multimedia security. pp. 5–10 (2016)
5. Bi, X., Liu, B., Yang, F., Xiao, B., Li, W., Huang, G., Cosman, P.C.: Detecting generated images by real images only. arXiv preprint arXiv:2311.00962 (2023)

6. Chen, L., Zhang, Y., Song, Y., Liu, L., Wang, J.: Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18710–18719 (2022)
7. Chen, R., Xi, J., Yan, Z., Zhang, K.Y., Wu, S., Xie, J., Chen, X., Xu, L., Guan, I., Yao, T., et al.: Dual data alignment makes ai-generated image detector easier generalizable. arXiv preprint arXiv:2505.14359 (2025)
8. Chen, X., Dong, C., Ji, J., Cao, J., Li, X.: Image manipulation detection by multi-view multi-scale supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14185–14193 (2021)
9. Google: Nanobanana. `https://developers.googleblog.com/en/introducing-gemini-2-5-flash-image/` (2025), accessed: 2026-06-29
10. Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)
11. Guo, X., Liu, X., Ren, Z., Grosz, S., Masi, I., Liu, X.: Hierarchical fine-grained image forgery detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3155–3165 (June 2023)
12. Gupta, T., Kembhavi, A.: Visual programming: Compositional visual reasoning without training. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14953–14962 (2023)
13. Hong, J., Zhao, C., Zhu, C., Lu, W., Xu, G., Yu, X.: Deepeyesv2: Toward agentic multimodal model. arXiv preprint arXiv:2511.05271 (2025)
14. Huang, Z., Hu, J., Li, X., He, Y., Zhao, X., Peng, B., Wu, B., Huang, X., Cheng, G.: Sida: Social media image deepfake detection, localization and explanation with large multimodal model. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 28831–28841 (2025)
15. Huang, Z., Li, T., Li, X., Wen, H., He, Y., Zhang, J., Fei, H., Yang, X., Huang, X., Peng, B., et al.: So-fake: Benchmarking and explaining social media image forgery detection. arXiv preprint arXiv:2505.18660 (2025)
16. Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al.: Openai o1 system card. arXiv preprint arXiv:2412.16720 (2024)
17. Kang, H., Wen, S., Wen, Z., Ye, J., Li, W., Feng, P., Zhou, B., Wang, B., Lin, D., Zhang, L., et al.: Legion: Learning to ground and explain for synthetic image detection. arXiv preprint arXiv:2503.15264 (2025)
18. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
19. Labs, B.F.: Flux. `https://github.com/black-forest-labs/flux` (2024), accessed: 2026-06-29
20. Li, O., Cai, J., Hao, Y., Jiang, X., Hu, Y., Feng, F.: Improving synthetic image detection towards generalization: An image transformation perspective. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1. pp. 2405–2414 (2025)
21. Li, Y., Tian, Y., Huang, Y., Lu, W., Wang, S., Lin, W., Rocha, A.: Fakescope: Large multimodal expert model for transparent ai-generated image forensics. arXiv preprint arXiv:2503.24267 (2025)
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)

23. Liu, H., Tan, Z., Tan, C., Wei, Y., Wang, J., Zhao, Y.: Forgery-aware adaptive transformer for generalizable synthetic image detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10770–10780 (2024)
24. Liu, J., Zhang, F., Zhu, J., Sun, E., Zhang, Q., Zha, Z.J.: Forgerygpt: Multimodal large language model for explainable image forgery detection and localization. arXiv preprint arXiv:2410.10238 (2024)
25. Liu, Z., Qi, X., Torr, P.H.S.: Global texture enhancement for fake face detection in the wild. In: CVPR (2020)
26. Liu, Z., Zang, Y., Zou, Y., Liang, Z., Dong, X., Cao, Y., Duan, H., Lin, D., Wang, J.: Visual agentic reinforcement fine-tuning. arXiv preprint arXiv:2505.14246 (2025)
27. Luo, Y., Zhang, Y., Yan, J., Liu, W.: Generalizing face forgery detection with high-frequency features. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16317–16326 (2021)
28. Midjourney: Midjourney (version 7). `https://www.midjourney.com/` (2025), accessed: 2026-06-29
29. Ojha, U., Li, Y., Lee, Y.J.: Towards universal fake image detectors that generalize across generative models. In: CVPR (2023)
30. OpenAI: DALL·E 3. `https://openai.com/dall-e` (2024), accessed: 2026-06-29
31. OpenAI: Gpt-5 and the new era of work. `https://openai.com/index/gpt-5-new-era-of-work/` (aug 2025), accessed: 2026-02-28
32. OpenAI: Introducing gpt-4.1. `https://openai.com/index/gpt-4-1/` (2025), accessed: 2025-11-13
33. OpenAI: OpenAI o3-mini system card (2025), `https://cdn.openai.com/o3-mini-system-card-feb10.pdf`, published January 31, 2025. Accessed: 2026-06-29
34. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: improving latent diffusion models for high-resolution image synthesis. In: ICLR. OpenReview.net (2024)
35. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on signal processing **53**(2), 758–767 (2005)
36. Qi, S., Zhang, Y., Wang, C., Zhou, J., Cao, X.: A principled design of image representation: Towards forensic tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(5), 5337–5354 (2022)
37. Qiao, T., Xie, S., Chen, Y., Retraint, F., Luo, X.: Fully unsupervised deepfake video detection via enhanced contrastive learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
38. Rao, Y., Ni, J., Zhang, W., Huang, J.: Towards jpeg-resistant image forgery detection and localization via self-supervised domain adaptation. IEEE transactions on pattern analysis and machine intelligence **47**(5), 3285–3297 (2022)
39. Shao, R., Wu, T., Wu, J., Nie, L., Liu, Z.: Detecting and grounding multi-modal media manipulation and beyond. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
40. Su, Z., Li, L., Song, M., Hao, Y., Yang, Z., Zhang, J., Chen, G., Gu, J., Li, J., Qu, X., et al.: Openthinkimg: Learning to think with images via visual tool reinforcement learning. arXiv preprint arXiv:2505.08617 (2025)
41. Surís, D., Menon, S., Vondrick, C.: Vipergpt: Visual inference via python execution for reasoning. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11888–11898 (2023)
42. Tan, C., Zhao, Y., Wei, S., Gu, G., Liu, P., Wei, Y.: Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning.

In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 5052–5060 (2024)
43. Tan, C., Zhao, Y., Wei, S., Gu, G., Wei, Y.: Learning on gradients: Generalized artifacts representation for gan-generated images detection. In: CVPR (2023)
44. Team, G., Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
45. Team, Q.: Qvq: To see the world with wisdom (December 2024), https://qwenlm.github.io/blog/qvq-72b-preview/, accessed: 2026-06-29
46. Wang, J., Wu, Z., Chen, J., Han, X., Shrivastava, A., Lim, S.N., Jiang, Y.G.: Objectformer for image manipulation detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2364–2373 (2022)
47. Wang, W., Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Zhu, J., Zhu, X., Lu, L., Qiao, Y., Dai, J.: Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. arXiv preprint arXiv:2411.10442 (2024)
48. Wang, X., Li, D., Zhao, Y., Wang, H., et al.: Metatool: Facilitating large language models to master tools with meta-task augmentation. arXiv preprint arXiv:2407.12871 (2024)
49. Wang, Y., Feng, Z., Wang, J., Lou, H., Zhou, B., Lei, J., Song, M., Bei, Y.: Spatial-temporal forgery trace based forgery image identification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17067–17076 (2025)
50. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
51. Xu, Z., Zhang, X., Li, R., Tang, Z., Huang, Q., Zhang, J.: Fakeshield: Explainable image forgery detection and localization via multi-modal large language models. In: International Conference on Learning Representations (2025)
52. Yan, S., Li, O., Cai, J., Hao, Y., Jiang, X., Hu, Y., Xie, W.: A sanity check for ai-generated image detection. arXiv preprint arXiv:2406.19435 (2024)
53. Yan, Z., Wang, J., Jin, P., Zhang, K.Y., Liu, C., Chen, S., Yao, T., Ding, S., Wu, B., Yuan, L.: Orthogonal subspace decomposition for generalizable ai-generated image detection. arXiv preprint arXiv:2411.15633 (2024)
54. Ye, J., Zhou, B., Huang, Z., Zhang, J., Bai, T., Kang, H., He, J., Lin, H., Wang, Z., Wu, T., et al.: Loki: A comprehensive synthetic data detection benchmark using large multimodal models. arXiv preprint arXiv:2410.09732 (2024)
55. Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., Chen, E.: A survey on multimodal large language models. CoRR **abs/2306.13549** (2023). https://doi.org/10.48550/ARXIV.2306.13549, https://doi.org/10.48550/arXiv.2306.13549
56. Zhang, Y.F., Lu, X., Yin, S., Fu, C., Chen, W., Hu, X., Wen, B., Jiang, K., Liu, C., Zhang, T., et al.: Thyme: Think beyond images. arXiv preprint arXiv:2508.11630 (2025)
57. Zhang, Y., Colman, B., Guo, X., Shahriyari, A., Bharaj, G.: Common sense reasoning for deepfake detection. In: European Conference on Computer Vision. pp. 399–415. Springer (2024)
58. Zhang, Y., Tan, Q., Qi, S., Xue, M.: Prnu-based image forgery localization with deep multi-scale fusion. ACM Transactions on Multimedia Computing, Communications and Applications **19**(2), 1–20 (2023)
59. Zhao, S., Zhang, H., Lin, S., Li, M., Wu, Q., Zhang, K., Wei, C.: Pyvision: Agentic vision with dynamic tooling. arXiv preprint arXiv:2507.07998 (2025)

60. Zheng, Z., Yang, M., Hong, J., Zhao, C., Xu, G., Yang, L., Shen, C., Yu, X.: Deepeyes: Incentivizing" thinking with images" via reinforcement learning. arXiv preprint arXiv:2505.14362 (2025)
61. Zhou, Z., Chen, D., Ma, Z., Hu, Z., Fu, M., Wang, S., Wan, Y., Zhao, Z., Krishna, R.: Reinforced visual perception with tools. arXiv preprint arXiv:2509.01656 (2025)
62. Zhou, Z., Luo, Y., Wu, Y., Sun, K., Ji, J., Yan, K., Ding, S., Sun, X., Wu, Y., Ji, R.: Aigi-holmes: Towards explainable and generalizable ai-generated image detection via multimodal large language models. arXiv preprint arXiv:2507.02664 (2025)
63. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., et al.: Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479 (2025)
64. Zhu, K., Gu, J., You, Z., Qiao, Y., Dong, C.: An intelligent agentic system for complex image restoration problems. arXiv preprint arXiv:2410.17809 (2024)
65. Zhu, X., Fei, H., Zhang, B., Zhang, T., Zhang, X., Li, S.Z., Lei, Z.: Face forgery detection by 3d decomposition and composition search. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(7), 8342–8357 (2023)