

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

Fanrui Zhang^{1,2}, Qiang Zhang¹, Sizhuo Zhou^{1,2}, Jianwen Sun², Chuanhao Li³,
Jiaxin Ai², Yukang Feng², Yujie Zhang², Wenjie Li², Zizhen Li²,
Yifan Chang^{1,2}, Jiawei Liu¹, Kaipeng Zhang^{2,3}

¹University of Science and Technology of China, China

²Shanghai Innovation Institute ³Shanghai Artificial Intelligence Laboratory

zfr888@mail.ustc.edu.cn

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 generate, execute, and iteratively refine Python-based low-level tools around the detection objective, thereby achieving more flexible and interpretable forgery analysis. *ForenAgent* follows a two-stage training pipeline combining Cold Start and Reinforcement Fine-Tuning to enhance its tool interaction capability and reasoning adaptability progressively. Inspired by human reasoning, we design a dynamic reasoning loop comprising global perception, local focusing, iterative probing, and holistic adjudication, and instantiate it as both a data-sampling strategy and a task-aligned process reward. For systematic training and evaluation, we construct *FABench*, a heterogeneous, high-quality agent-forensics dataset comprising 100k images and approximately 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 will be released after the review process is completed.

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

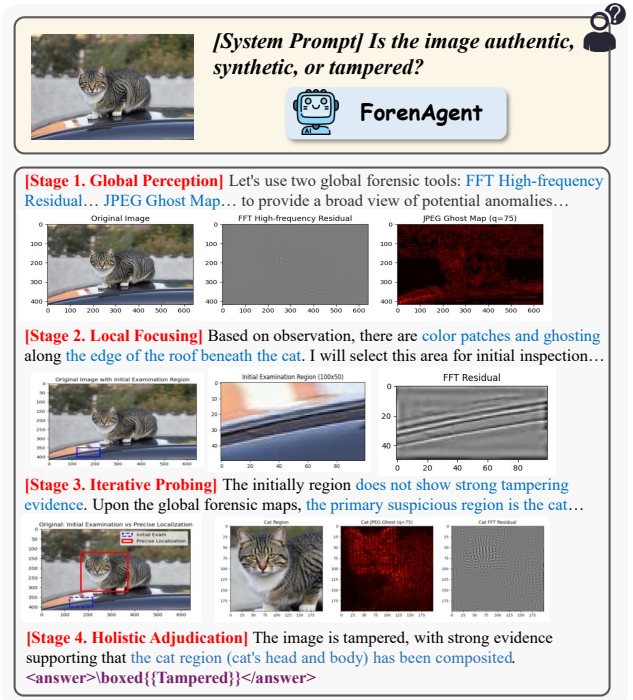


Figure 1. ForenAgent autonomously composes a global-to-local Python toolchain, accurately delivers a tampered verdict with precise localization of the forged region, and further demonstrates reflective self-correction by carefully revising an initially mislocalized crop to the appropriate region of interest.

expression but also enables the malicious fabrication of multimedia content. [31–34, 54]. 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 [46]. Methods such as FakeShield [45] and SIDA [11] 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.

Recent progress in MLLMs has shown that they are increasingly capable of complex reasoning and interaction with external tools [13, 49, 50]. 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 present ForenAgent, a novel interactive multi-turn framework that empowers MLLM to autonomously generate, execute, and iteratively refine Python-based low-level tools for IFD. To achieve this, we first abstract and generalize several commonly used low-level Python tools for IFD, such as frequency residual, noise residual, and high-pass filtering, and then integrate them into a comprehensive toolbox consisting of 12 candidate utilities for future community extension. 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 [13], Nano-Banana [7], and Midjourney-v7 [24]). 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 [27] 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:

(1) We propose ForenAgent, a novel interactive, multi-turn framework that enables an MLLM to autonomously generate, 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

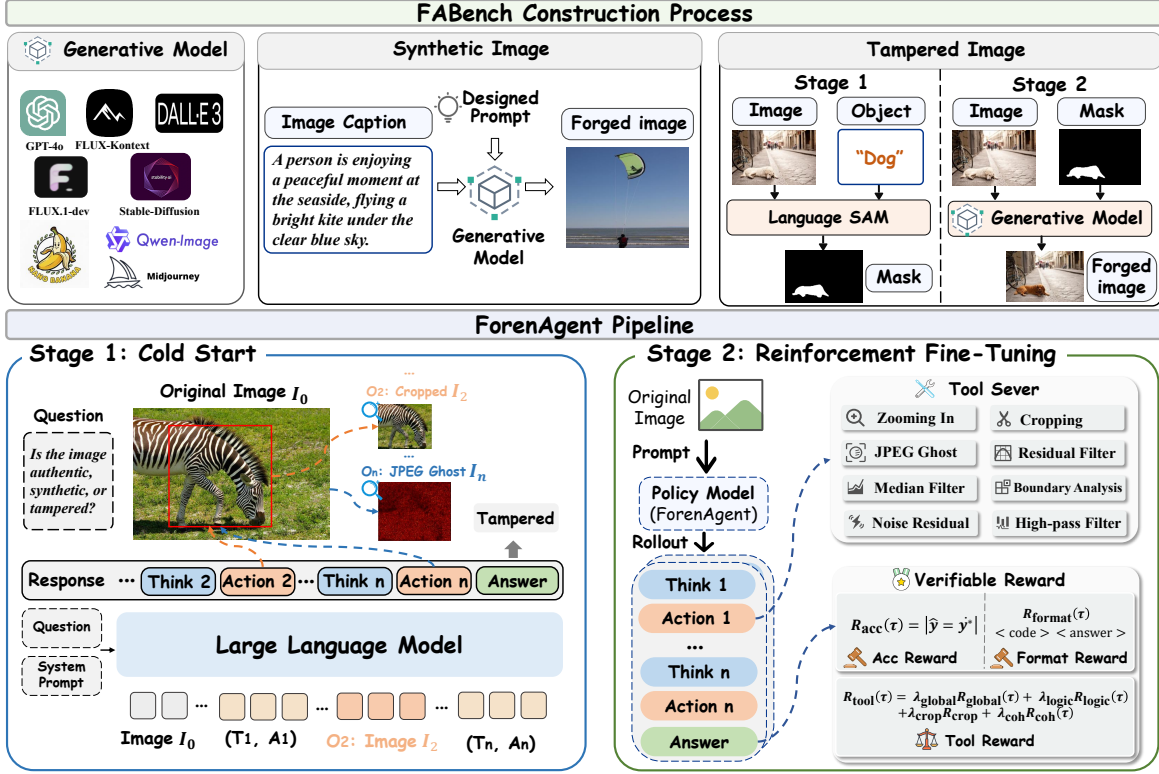


Figure 2. The overall architecture of the ForenAgent is illustrated, with the upper part showing the FABench construction process and the lower part presenting the training pipeline of ForenAgent.

reward that foster flexible, tool-adaptive evidence reasoning and enable robust, interpretable decisions.

2. Related Work

2.1. Image Forgery Detection

Early IFD research largely centered on low-level feature extractors that capture non-semantic inconsistencies between manipulated and authentic regions. Representative methods include frequency- and residual-based cues: FreqNet applies FFT to learn high-frequency patterns and improves cross-dataset generalization [38]; RGB-N employs Steganalysis Rich Model (SRM) filters to expose local noise inconsistencies [23]; SAFE leverages Discrete Wavelet Transform (DWT)/Discrete Cosine Transform (DCT) with crop-based preprocessing to preserve local artifacts [23]; PRNU-based approaches exploit camera fingerprints for multi-scale trace analysis with end-to-end fusion [47]; JPEG ghost detection reveals regions with inconsistent compression [30]; ObjectFormer localizes forgeries via DCT high-pass cues [41]; MVSS-Net integrates Bayar convolution and Sobel operators for fine-grained boundaries [6], while HiFi-Net strengthens multi-level detection with high-pass filtering [9]. Additional work further enriches low-level forensic cues and robustness [3, 5, 19]. However, these

methods often fail to capture higher-level forensic semantics, particularly those driven by language and knowledge, reducing their effectiveness against complex manipulations.

To address these challenges, recent work integrates large language models (LLMs) with vision-language reasoning for image forensics. For tampered images, Zhang et al. introduce DD-VQA [48], a dataset of 2,968 FaceForensics++ samples used to fine-tune BLIP with contrastive loss, improving both accuracy and explanatory quality. FakeShield [45] combines an LLM with visual understanding via dual modules, while ForgeryGPT [20] tailors an LLM to capture higher-order forensic cues across diverse feature spaces for explainable generation and interactive dialogue. For synthetic images, FakeScope [17] and LEGION [14] leverage MLLMs to deliver strong explainability and detection performance. Building on these advances, SIDA [11] and So-Fake-R1 [12] unify tampered and synthetic detection in a multi-class setting to evaluate MLLM capabilities. Collectively, these efforts mark a shift in IFD research toward multimodal reasoning, yielding more accurate and interpretable forgery analysis.

2.2. Thinking with Images

The “Thinking with Images” paradigm is pushing MLLMs beyond passive description toward interactive, iterative



Figure 3. Examples of tampered and synthetic images from diverse FABench generators.

agents [53]. Early work typically relied on predefined CoT formats and static toolsets (e.g., VisProg [10] and ViperGPT [36]) to prompt models to call fixed tools for specific vision tasks, but this design limits flexibility and generalization. To address this, METATOOL [43] introduces meta-task augmentation to improve tool mastery and transfer, while PyVision [49] enables dynamic Python code generation to invoke complex tools for more versatile reasoning. Pushing autonomy further, recent studies optimize tool use via reinforcement learning (RL) [22]. DeepEyes [50] performs end-to-end reinforcement learning with tool-oriented data selection and reward design; V-TOOLRL [35] directly maximizes task success from tool-interaction feedback; and ReVPT [51] adopts a two-stage scheme with multiple visualization tools to lift general capability.

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 [15]) mainly contain low-fidelity generations that are significantly easier than modern photorealistic outputs (e.g., GPT-4o-image [13], Midjourney-v7 [24]). (ii) Fixed tampering pipelines. Many datasets rely on a narrow set of inpainting models (e.g., Stable Diffusion [29]) and rarely explore newer pipelines (e.g., FLUX-Kontext [16], Qwen-image [44]), 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:

- Authentic (40k): COCO [18] 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 [18], 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 pipeline is provided in the *Supplementary Material*.

We adopt a multi-stage pipeline consisting of quality validation (resolution bounds, 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 [26], FLUX-1-dev, FLUX-Kontext, Stable Diffusion, Qwen-Image, and Nano Banana [7]. 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. The composition of the training set is provided in the *Supplementary Material*. Figure 3 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

We present ForenAgent, an novel interactive multi-turn framework that enables the MLLMs to autonomously gen-

erate, execute, and iteratively refine Python-based low-level tools for IFD. This approach provides more flexible and interpretable solutions compared to traditional methods. ForenAgent empowers MLLMs to dynamically generate and execute low-level Python code during the reasoning process. In each session, the MLLM receives input, generates Python code as a response, and executes it within an isolated Python runtime environment. The generated outputs, whether textual, visual, or both, are fed back into the MLLM’s context to iteratively refine its reasoning over multiple turns until a final answer is produced.

3.2.1. Tool Boxes

ForenAgent provides Python as the fundamental building block for tool construction. We identify two major categories of tools in the following:

(1) Basic Image Processing: These 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 applies contrast adjustments and other enhancements to make latent structures 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:

- **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.
- **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.
- **Edge & Boundary Analysis:** Methods to pinpoint inconsistent edges or gradients. (8) **Sobel Edge Detector:** Identifies splicing boundaries or anomalous edge patterns. (9)

General High-Pass Filters: Extracts high-frequency components to detect tampering artifacts.

- **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:** Statistically measures artifacts and suspicious smoothing patterns.
- **Statistical Analysis:** (12) **Local Correlation Map:** Quantifies enhanced correlations within pixel neighborhoods, often indicative of manipulation.

3.2.2. Cold Start

The training process for ForenAgent consists of two sequential stages, designed to progressively equip the MLLM with the capabilities to handle complex IFD tasks.

System Prompt Design: To steer the MLLM’s reasoning and code generation, ForenAgent uses a carefully engineered system prompt in addition to user queries. The prompt specifies how to access inputs, structure code, and return the final answer: (i) encourage executable code over free-form text; (ii) preload images/frames as `image.clue_i` (with resolution) so the model can reference them directly (e.g., cropping); (iii) standardize outputs via `print(...)` (text) and `plt.show()` (visuals); (iv) wrap every code block in `<code>...</code>` for reliable parsing; (v) place the final class token inside `<answer>...</answer>` for consistent evaluation. This design yields parsable, executable code with minimal runtime errors. The full system prompt appears in the *Supplementary material*.

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 [27] to obtain an authenticity judgment and a multi-step chain. We retain a response only if: (1) the predicted label is correct; (2) it contains executable Python wrapped in `<code>...</code>`; (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.

3.2.3. Reinforcement Fine-Tuning

Following DeepEyes [50] and related work, we study how MLLMs acquire tool-calling and reasoning without supervised labels, using pure RL for 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 [8], we sample a small candidate set per input, score them within-group, and update the policy without a critic using clipped importance weights plus a KL penalty to a reference, stabilizing training and leveraging preferences from model- or human-generated 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>` and `<answer>` tags, we introduce a tool usage reward R_{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}_{low} . The reward $R_{\text{tool}}(\tau)$ integrates four components to assess the logical and context-aware 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:

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

The global priority reward is:

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

(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

$$\mathbb{I}_{\text{crop}} = \mathbb{K}\{\exists t \in \{1, \dots, T\} : a_t = \text{Crop}\}, \quad (3)$$

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

(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

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

The tool usage reward aggregates the four sub-rewards:

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

Finally, the overall reward function R is defined as:

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

By incorporating these verifiable reward components into the reinforcement learning process, ForenAgent achieves more interpretable and systematic reasoning for image forensic detection, 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 against 18 competitive baselines in three groups: (1) Closed-source MLLMs: Gemini2.5-flash [39], Gemini2.5-Pro [39], GPT-4o [1], GPT-o3-mini [13], GPT-4.1 [27], GPT-5 [28]. (2) Large-scale MLLMs (zero-shot, no finetuning): InternVL3-78B [52], Qwen2.5-VL-72B [2], QVQ-72B-preview [40], InternVL2.5-78B-MPO [42], Qwen3-VL-30B [2]. (3) Trained baselines (same image training set as ForenAgent): we train Qwen2.5-VL-7B [2] and Qwen3-VL-8B [2] as MLLM comparison models. We also include advanced detectors Gram-Net [21], UnivFD [25], LGrad [37], LNP [4], and SIDA (We use the 13B-parameter variant.) [11].

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

Evaluation Metrics. Following prior work [11], we evaluate detection at the image level using Accuracy and F1. More details are provided in the *Supplementary Material*.

4.2. Detection Evaluation

As shown in Table 1, we report performance across all test splits, and ForenAgent achieves the highest overall accuracy and F1-score, while also outperforming all baselines on both the synthetic and tampered categories. Notably, zero-shot performance from both closed-source and large-scale open-source MLLMs remains weak: these models consistently misclassify tampered and synthetic images as authentic, highlighting the insufficiency of IFD-relevant knowledge in current MLLM pretraining corpora. After supervised training, Qwen2.5-VL-7B outperforms Qwen3-VL-8B, supporting our choice of Qwen2.5-VL as the backbone.

Table 1. Comparison of ForenAgent with other state-of-the-art methods on the FABench dataset.

Methods	Authentic				Synthetic				Tampered				Overall	
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	F1
Gemini2.5-flash [39]	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 [39]	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 [13]	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 [27]	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 [28]	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 [52]	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 [2]	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 [40]	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 [42]	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
Qwen2.5-VL-7B [2]	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	79.7
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 [21]	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 [11]	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
LGrad [37]	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 [4]	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
UnivFD [25]	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	93.3	89.3	89.4	89.4	91.3	86.2	88.0	87.1	92.1	89.0	87.0	88.0	88.1	88.2

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 [2]	72.9	69.9
Qwen3-VL-8B [2]	68.7	65.5
Gram-Net [21]	53.4	55.0
SIDA [11]	77.2	77.1
LGrad [37]	64.5	64.5
LNP [4]	53.3	53.2
UnivFD [25]	61.1	60.9
ForenAgent	80.6	80.4

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 Gram-Net, UnivFD, LGrad, LNP, and SIDA. ForenAgent achieves the highest scores, demonstrating its strong adaptive capacity. We further discuss in the *Supplementary Material* the limitations of current MLLMs in using bounding boxes for forgery localization and outline the optimal pipeline for adapting

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

Methods	Accuracy	F1-score
w/o Cold Start	79.8	78.7
w/o RFT	81.9	81.7
w/o Tool Reward	83.6	83.5
ForenAgent	88.1	88.2

ForenAgent to pixel-level localization.

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.

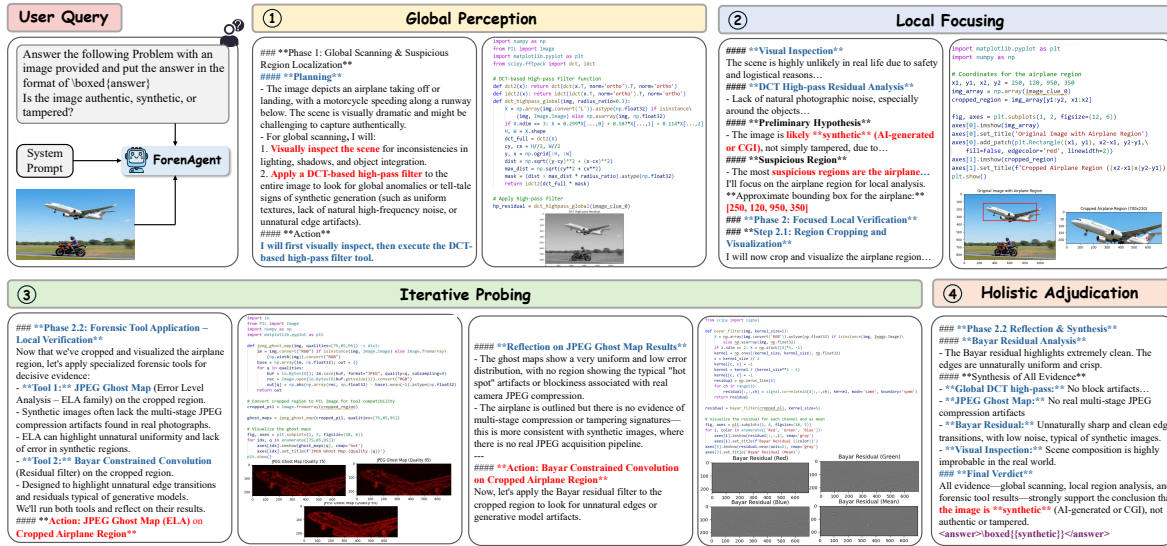


Figure 4. The complete evidence chain by which ForenAgent correctly identifies a synthetic image.

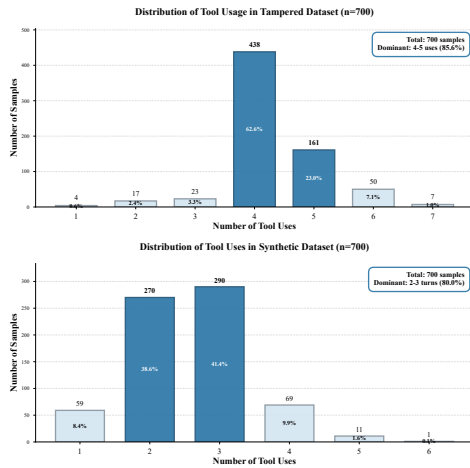


Figure 5. Distribution of tool usage frequencies.

4.5. Visualization

Figure 4 illustrates a successful synthetic-image case where ForenAgent constructs a coherent evidence chain that mirrors human reasoning, progressing through global perception, local focusing, iterative probing, and holistic adjudication to deliver both accurate detection and a convincing explanation. It first applies DCT-based global screening and flags initial suspicion, then conducts local focusing and iteratively probes with JPEG Ghost and Bayar Constrained Convolution, and finally integrates all cues into a well-founded decision. This example shows that ForenAgent not only produces correct labels but also assembles a logically sound, tool-driven forensic rationale, charting a promising path toward general-purpose IFD.

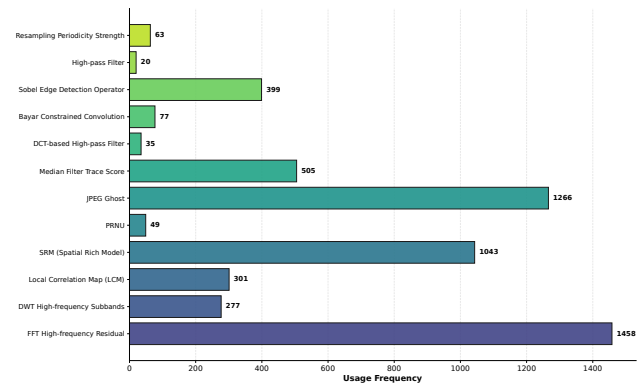


Figure 6. Low-level forensic tool usage frequency distribution across the FABench dataset.

4.6. Tool Usage Analysis

Figure 5 analyzes tool usage frequencies on the FABench test sets. For synthetic images, ForenAgent typically converges around 3 tool calls, whereas tampered images require around 4 calls due to their higher structural complexity and the need to localize manipulated regions. Figure 6 summarizes the usage distribution across low-level forensic tools: SRM, FFT, and JPEG Ghost are used most frequently, while PRNU and High-pass Filter appear 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 IFD techniques with modern MLLMs and potentially inspiring future hybrid-agent designs.

5. Conclusion

In this paper, we introduced ForenAgent, an interactive multi-round framework that enables MLLMs to autonomously construct and iteratively refine Python-based low-level tools for image forgery detection. We abstract and generalize key low-level Python 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.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6, 7
- [2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 6, 7
- [3] Belhassen Bayar and Matthew C Stamm. 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*, pages 5–10, 2016. 3
- [4] Xiuli Bi, Bo Liu, Fan Yang, Bin Xiao, Weisheng Li, Gao Huang, and Pamela C. Cosman. Detecting generated images by real images only. *Arxiv*, 2023. 6, 7
- [5] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. 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*, pages 18710–18719, 2022. 3
- [6] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14185–14193, 2021. 3
- [7] Google. Nanobanana. <https://aistudio.google.com/models/gemini-2-5-flash-image>, 2025b. 2, 4
- [8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 6
- [9] Xiao Guo, Xiaohong Liu, Zhiyuan Ren, Steven Grosz, Iacopo Masi, and Xiaoming Liu. Hierarchical fine-grained image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3155–3165, 2023. 3
- [10] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14953–14962, 2023. 4
- [11] Zhenglin Huang, Jinwei Hu, Xiangtai Li, Yiwei He, Xingyu Zhao, Bei Peng, Baoyuan Wu, Xiaowei Huang, and Guangliang Cheng. Sida: Social media image deepfake detection, localization and explanation with large multimodal model, 2025. 2, 3, 6, 7
- [12] Zhenglin Huang, Tianxiao Li, Xiangtai Li, Haiquan Wen, Yiwei He, Jiangning Zhang, Hao Fei, Xi Yang, Xiaowei Huang, Bei Peng, et al. So-fake: Benchmarking and explaining social media image forgery detection. *arXiv preprint arXiv:2505.18660*, 2025. 3
- [13] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. 2, 4, 6, 7
- [14] Hengrui Kang, Siwei Wen, Zichen Wen, Junyan Ye, Weijia Li, Peilin Feng, Baichuan Zhou, Bin Wang, Dahua Lin, Linfeng Zhang, et al. Legion: Learning to ground and explain for synthetic image detection. *arXiv preprint arXiv:2503.15264*, 2025. 3
- [15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 4
- [16] Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024. 4
- [17] Yixuan Li, Yu Tian, Yipo Huang, Wei Lu, Shiqi Wang, Weisi Lin, and Anderson Rocha. Fakescope: Large multimodal expert model for transparent ai-generated image forensics. *arXiv preprint arXiv:2503.24267*, 2025. 3
- [18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4
- [19] Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10770–10780, 2024. 3
- [20] Jiawei Liu, Fanrui Zhang, Jiaying Zhu, Esther Sun, Qiang Zhang, and Zheng-Jun Zha. Forgerygpt: Multimodal large language model for explainable image forgery detection and localization. *arXiv preprint arXiv:2410.10238*, 2024. 3
- [21] Zhengzhe Liu, Xiaojuan Qi, and Philip H. S. Torr. Global texture enhancement for fake face detection in the wild. In *CVPR*, 2020. 6, 7
- [22] Ziyu Liu, Yuhang Zang, Yushan Zou, Zijian Liang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Ji-

- aqi Wang. Visual agentic reinforcement fine-tuning. *arXiv preprint arXiv:2505.14246*, 2025. 4
- [23] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16317–16326, 2021. 3
- [24] Midjourney. Midjourney. <https://www.midjourney.com/home>, 2028. 2, 4
- [25] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, 2023. 6, 7
- [26] OpenAI. DALL-E 3. <https://openai.com/dall-e>, 2024. 4
- [27] OpenAI. Introducing gpt-4.1. <https://openai.com/index/gpt-4-1/>, 2025. Accessed: 2025-11-13. 2, 5, 6, 7
- [28] OpenAI. Introducing gpt-5. <https://openai.com/introducing-gpt-5/>, 2025. Accessed: 2025-11-13. 6, 7
- [29] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. In *ICLR*. OpenReview.net, 2024. 4
- [30] Alin C Popescu and Hany Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on signal processing*, 53(2):758–767, 2005. 3
- [31] Shuren Qi, Yushu Zhang, Chao Wang, Jiantao Zhou, and Xiaochun Cao. A principled design of image representation: Towards forensic tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5337–5354, 2022. 1
- [32] Tong Qiao, Shichuang Xie, Yanli Chen, Florent Retraint, and Xiangyang Luo. Fully unsupervised deepfake video detection via enhanced contrastive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [33] Yuan Rao, Jiangqun Ni, Weizhe Zhang, and Jiwu Huang. Towards jpeg-resistant image forgery detection and localization via self-supervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [34] Rui Shao, Tianxing Wu, Jianlong Wu, Liqiang Nie, and Ziwei Liu. Detecting and grounding multi-modal media manipulation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1
- [35] Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guan jie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, et al. Openthinking: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025. 4
- [36] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11888–11898, 2023. 4
- [37] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *CVPR*, 2023. 6, 7
- [38] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5052–5060, 2024. 3
- [39] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 6, 7
- [40] Qwen Team. Qvq: To see the world with wisdom, 2024. 6, 7
- [41] Junke Wang, Zuxuan Wu, Jingjing Chen, Xintong Han, Abhinav Shrivastava, Ser-Nam Lim, and Yu-Gang Jiang. Objectformer for image manipulation detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2364–2373, 2022. 3
- [42] Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024. 6, 7
- [43] Xiaohan Wang, Dian Li, Yilin Zhao, Hui Wang, et al. Meta-tool: Facilitating large language models to master tools with meta-task augmentation. *arXiv preprint arXiv:2407.12871*, 2024. 4
- [44] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 4
- [45] Zhipei Xu, Xuanyu Zhang, Runyi Li, Zecheng Tang, Qing Huang, and Jian Zhang. Fakeshield: Explainable image forgery detection and localization via multi-modal large language models. In *International Conference on Learning Representations*, 2025. 2, 3
- [46] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *CoRR*, abs/2306.13549, 2023. 2
- [47] Yushu Zhang, Qing Tan, Shuren Qi, and Mingfu Xue. Pnu-based image forgery localization with deep multi-scale fusion. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(2):1–20, 2023. 3
- [48] Yue Zhang, Ben Colman, Xiao Guo, Ali Shahriyari, and Gaurav Bharaj. Common sense reasoning for deepfake detection. In *European Conference on Computer Vision*, pages 399–415. Springer, 2024. 3
- [49] Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Ming Li, Qilong Wu, Kaipeng Zhang, and Chen Wei. Pyvision: Agentic vision with dynamic tooling. *arXiv preprint arXiv:2507.07998*, 2025. 2, 4
- [50] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deep-eyes: Incentivizing” thinking with images” via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025. 2, 4, 5

- [51] Zetong Zhou, Dongping Chen, Zixian Ma, Zhihan Hu, Mingyang Fu, Sinan Wang, Yao Wan, Zhou Zhao, and Ranjay Krishna. Reinforced visual perception with tools. *arXiv preprint arXiv:2509.01656*, 2025. [4](#)
- [52] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internv13: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. [6](#), [7](#)
- [53] Kaiwen Zhu, Jinjin Gu, Zhiyuan You, Yu Qiao, and Chao Dong. An intelligent agentic system for complex image restoration problems. *arXiv preprint arXiv:2410.17809*, 2024. [4](#)
- [54] Xiangyu Zhu, Hongyan Fei, Bin Zhang, Tianshuo Zhang, Xiaoyu Zhang, Stan Z Li, and Zhen Lei. Face forgery detection by 3d decomposition and composition search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8342–8357, 2023. [1](#)