EVOGUARD: AN EXTENSIBLE AGENTIC RL-BASED FRAMEWORK FOR PRACTICAL AND EVOLVING AI-GENERATED IMAGE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

The rapid proliferation of AI-Generated Images (AIGIs) poses severe misinformation risks, making AIGI detection critical yet challenging. Traditional detection paradigms mainly rely on low-level features, whereas recent research increasingly focuses on leveraging the general understanding ability of Multimodal Large Language Models (MLLMs) to achieve better generalization, yet it still suffers from limited extensibility and expensive data annotations. Instead of building yet another detector, we recast AIGI detection as learned, reasoning-based evidence synthesis over a pool of heterogeneous off-the-shelf detectors, realized through EvoGuard, a novel agentic framework. A capability-aware selection mechanism profiles each detector and gathers complementary evidence per sample; a dynamic orchestration mechanism then reasons over heterogeneous outputs across multiple rounds, cross-validating conflicting or low-confidence signals before concluding. This design exploits the complementary strengths among heterogeneous detectors, transcending the limits of any single model. Furthermore, optimized by a GRPO-based Agentic Reinforcement Learning algorithm using only low-cost binary labels, it eliminates the reliance on fine-grained annotations. Extensive experiments demonstrate that this learned reasoning paradigm outperforms single-detector and static ensembling, achieving SOTA accuracy while mitigating the bias between positive and negative samples. More importantly, it allows the plug-and-play integration of new detectors to boost overall performance in a train-free manner, offering a highly practical, long-term solution to ever-evolving AIGI threats. Source code is included in the supplementary materials and will be publicly available upon acceptance.

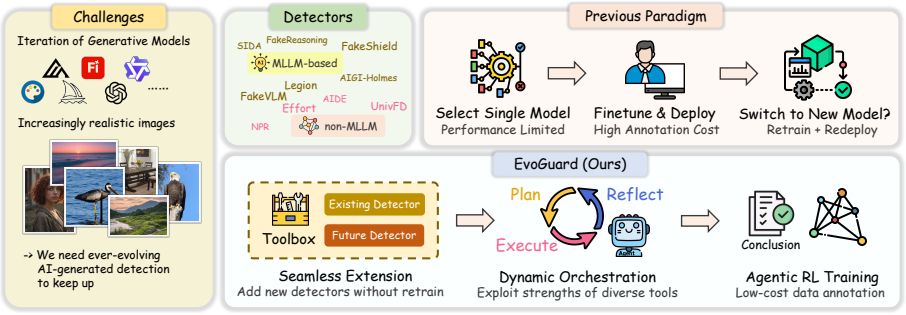


Figure 1: Motivation of EvoGuard. As generative models rapidly evolve, AIGI detection must evolve accordingly. Prior work focuses on building ever-stronger single detectors; we instead let an MLLM agent reason over a pool of off-the-shelf detectors, cross-validating their outputs rather than routing to one or fusing all by fixed rules. This reasoning-based synthesis exploits complementary strengths from heterogeneous detectors, enables training-free extensibility, and removes the need for fine-grained annotations.

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

1 INTRODUCTION

From GANs (Goodfellow et al., 2014), Diffusion models (Rombach et al., 2022) to autoregressive models (Fan et al., 2024; Liu et al., 2025a), the rapid evolution of AI generative models has spurred unprecedented growth in the creative industry. However, the proliferation of hyper-realistic AI-Generated Images (AIGIs) has introduced severe risks of misinformation (Ha et al., 2024), making AIGI detection a critical research area. Existing methods include capturing generator artifacts in spatial and frequency domains (Wang et al., 2020; Tan et al., 2024), leveraging pre-trained vision encoders (Yan et al., 2025; Ojha et al., 2023), and exploiting diffusion reconstruction errors (Wang et al., 2023). However, due to the rapid iteration of generative models, image degradations on social media (Li et al., 2025d), and sophisticated adversarial attacks (Tailanián et al., 2024), existing detectors struggle in complex and ever-changing real-world environments (Tariang et al., 2024). The problem of detecting AI-generated images is far from being solved (Yan et al., 2024).

The recent emergence of Multimodal Large Language Models (MLLMs) offers a novel perspective. Unlike traditional paradigms that rely heavily on low-level features (Zhong et al., 2023), MLLMs comprehend images through high-level semantics, identifying artifacts requiring global reasoning, such as logical inconsistencies or physical anomalies (Kang et al., 2025). These semantic representations are robust to image transformations, thereby improving generalization (Wang et al., 2025b), while their ability to generate explanations provides a rationale for detection. However, MLLMs are insensitive to low-level features, and often fail to capture fine-grained, pixel-level artifacts (Jia et al., 2024). Moreover, developing effective explanatory and reasoning capabilities (Wen et al., 2025; Huang et al., 2025e), or aligning auxiliary vision encoder features with the LLM's semantic space (Gao et al., 2025; Zhou et al., 2025), typically requires large amounts of carefully annotated data, making it difficult to balance data quality against acquisition costs (Zhang et al., 2025b).

We argue that an effective AIGI detection system must simultaneously address three fundamental challenges: (1) Accuracy and Generalization. The method must generalize effectively across diverse generative models and remain robust to real-world image perturbations. (2) Extensibility. As generative models continuously advance, detectors must be capable of swift adaptation to keep pace. (3) Practicality. The required training annotations should be easy to obtain.

To alleviate the aforementioned issues, we further explore MLLMs for AIGI detection, motivated by the strong decision-making and planning capabilities exhibited by modern LLMs. While prior studies largely focus on developing powerful monolithic MLLM-based detectors, the idea of tackling this task via autonomous planning within an agentic framework remains largely unexplored. An agent is an intelligent entity capable of perception, reasoning, planning, and execution, which can leverage diverse external tools to solve complex problems (Wang et al., 2024). Agentic frameworks have been widely adopted across various domains (Abou Ali et al., 2025), demonstrating remarkable adaptability, generalization, and immense potential for flexibility and extensibility. These intrinsic attributes of agents perfectly align with the core requirements of AIGI detection.

Guided by the agentic paradigm, we propose EvoGuard, which reframes AIGI detection into reasoning-based evidence synthesis through an agentic framework: First, we encapsulate diverse off-the-shelf detectors (both MLLM-based and non-MLLM-based) as executable tools and propose a tool profile for each. Notably, these profiles are designed to use ambiguous, linguistic descriptors rather than exact metrics, enabling the agent to adapt better to distribution shifts. We then devise a capability-aware dynamic orchestration mechanism, empowering the agent to dynamically select tools based on the input image and the tool profile, and at each round, to reason over their heterogeneous outputs and cross-validate conflicting or low-confidence predictions before deciding whether to invoke additional tools or synthesize a final verdict. In this way, it performs multi-round, reasoning-based evidence synthesis over heterogeneous detectors, going beyond traditional single-detector and fixed-ensemble methods. We optimize the agent using GRPO (Shao et al., 2024), a prevalent Agentic Reinforcement Learning (Agentic RL) algorithm, with a specially designed reward to elicit its reasoning and execution ability from low-cost binary labels alone. Our main contributions are summarized as follows:

- **A novel agentic paradigm for AIGI detection.** We pioneer formulating AIGI detection as reasoning-based evidence synthesis via an agentic framework. Instead of routing to one detector or fusing outputs by fixed rules, the agent synthesizes complementary evidence from heterogeneous detectors, transcending the limits of single detectors and static ensembles.

Table 1: Comparison with other AIGI detection methods. "Could be Train-Free" implies that it is theoretically possible to be train-free, though this has not yet been empirically validated. * indicates available checkpoint and code for AIGI detection; these methods serve as baselines in our experiments. † indicates methods included in the EvoGuard toolbox.

Method	Paradigm	Detector Composition	Any-Detector Compatibility	Extensibility to New Detectors
Effort*† (ICML '25)	Single Detector	non-MLLM Detector (CLIP)	–	–
AIDE*† (ICLR '25)	Single Detector	non-MLLM Detector (CLIP + CNN)	–	–
MIRROR*† (arXiv '26)	Single Detector	non-MLLM Detector (DINO)	–	–
FakeReasoning* (TIP '26)	Single Detector	MLLM-based Detector	–	–
LEGION (ICCV '25)	Single Detector	MLLM-based Detector	–	–
FakeVLM*† (NeurIPS '25)	Single Detector	MLLM-based Detector	–	–
SIDA* (CVPR '25)	Single Detector	MLLM-based Detector + Extra Module	–	–
FakeShield* (ICLR '25)	Single Detector	MLLM-based Detector + Extra Module	–	–
AIGI-Holmes (ICCV '25)	Single Detector	MLLM-based Detector + Extra Module	–	–
Forensic-MOE* (ICCV '25)	Ensemble (MLP-based Fusion)	Multiple non-MLLM Detectors	NO	Training Required
E3 (CVPRW '24)	Ensemble (Transformer-based Fusion)	Multiple non-MLLM Detectors	NO	Training Required
AlignGemini (arXiv '26)	Ensemble (Fusion by OR Operation)	MLLM-based + non-MLLM Detectors	YES	Train-Free
X2-DFD (NeurIPS '25)	Pipeline (Non-MLLM outputs → MLLM)	MLLM-based + non-MLLM Detectors	YES	Could be Train-Free
EvoGuard (Ours)	**Agentic Framework (Dynamic)**	**Any Composition of Detectors**	**YES**	**Train-Free**

- **SOTA accuracy and mitigated bias.** We propose EvoGuard, which leverages agent's perception, reasoning, and planning capabilities to exploit complementary strengths of diverse tools across multiple action rounds. Extensive experiments show that EvoGuard achieves SOTA accuracy while balancing performance between positive and negative samples.
- **Seamless extensibility and efficient training strategy.** Experiments show that EvoGuard can improve performance without retraining by plug-and-play integration of off-the-shelf new tools, offering a promising long-term solution to evolving AIGI threats. Moreover, we introduce a GRPO-based Agentic RL algorithm that trains using only a small set of low-cost binary labels, overcoming the reliance on expensive fine-grained annotations.

2 Related Work

2.1 AI-Generated Image Detection

Early AIGI detection largely relied on identifying synthetic artifacts (Chai et al., 2020), e.g., generator fingerprints in spatial/frequency features (Qian et al., 2020; Wang et al., 2020; Frank et al., 2020) or upsampling-induced textures (Tan et al., 2024). These cues often overfit to specific generators and are fragile to real-world degradations (Liu et al., 2025b). More fundamentally, this paradigm treats authenticity as a sink class and ties the decision boundary to particular forgery domains (Ojha et al., 2023), exposing significant generalization deficiencies as generative models rapidly advance.

Recent studies explored leveraging pre-trained visual encoders (Ojha et al., 2023; Yan et al., 2025; 2024; Peng et al., 2025a; Costanzino et al., 2025) such as CLIP (Radford et al., 2021), SAM (Kirillov et al., 2023), and DINO (Oquab et al., 2023) to incorporate high-level semantics and capture artifacts requiring global reasoning (Kang et al., 2025). Others compare images with authentic visual priors (Liu et al., 2026; Li et al., 2025b), exploit diffusion-reconstruction errors (Wang et al., 2023; Wu et al., 2025a), or address bias from generation-agnostic artifacts (Guillaro et al., 2025; Chen et al., 2025b; Rajan et al., 2024; Chen et al., 2024a; Cai et al., 2025). These approaches improve individual detectors, while combining their complementary strengths remains a separate challenge.

The rapid evolution of generative models has rendered single detectors increasingly inadequate. A classic remedy is Mixtures of Experts (MoE), either by training fusion networks (Fang et al., 2025; Azizpour et al., 2024; Liu et al., 2024b; Kong et al., 2025) or directly combining detector outputs (Chen et al., 2025a). Another line of work adapts detectors to new generators through additional training data (Epstein et al., 2023; Park & Owens, 2025). Compared to these methods, EvoGuard proposes the capability-aware dynamic orchestration mechanism to dynamically invoke heterogeneous detectors and exploit their complementary strengths, enabling retraining-free, plug-and-play extensibility in response to complex, ever-changing real-world environments.

2.2 MLLM in Visual Forensics

MLLMs contribute semantic understanding and explanatory reasoning to visual forensics (Ji et al., 2025a). Their reliance on high-level semantics benefits robustness and generalization (Wang et al.,

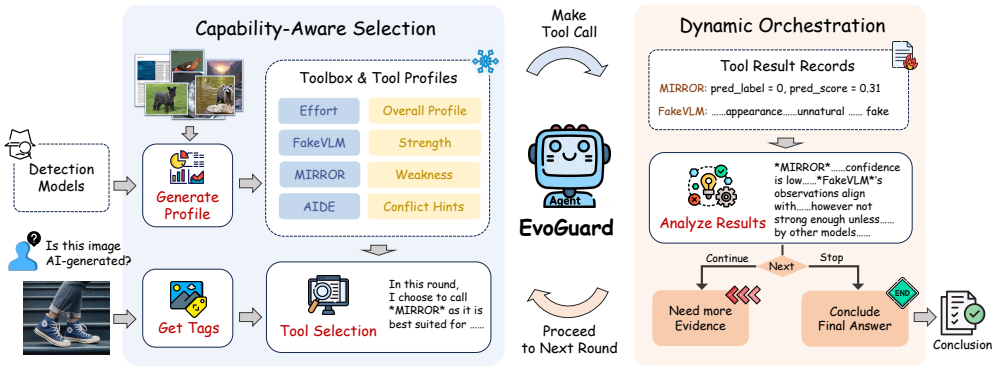


Figure 2: Overview of EvoGuard. We wrap heterogeneous AIGI detectors as tools with capability profiles. Given a query image, the agent first performs **Capability-Aware Selection** by matching image tags with tool profiles to select suitable tools, then performs **Dynamic Orchestration** by iteratively analyzing tool outputs to decide whether to call additional tools for validation or to conclude.

2025b), but the lack of low-level visual backbones limits their ability to capture pixel-level artifacts (Zhang et al., 2025b; He et al., 2025), hindering forensic performance (Li et al., 2025c; Zhang et al., 2024b; Jia et al., 2024).

Prior work addresses this limitation by improving artifact-oriented reasoning (Wen et al., 2025; Tan et al., 2025a; Huang et al., 2025c; Zhang et al., 2024a; Tan et al., 2025b; Qin et al., 2025), adding low-level visual features to MLLMs (Xu et al., 2025b; Zhou et al., 2025; Gao et al., 2025; Sun et al., 2025a; Wang et al., 2025a; Yu et al., 2025; Liu et al., 2024a; Sun et al., 2024; Nguyen et al., 2025), or passing specialist detector outputs as text or images (Chen et al., 2024b; Peng et al., 2025b; Sun et al., 2025b). Other methods use task-specific tokens for classification or localization (Huang et al., 2025d; Xu et al., 2025a; Li et al., 2025a; Zhang et al., 2025b; Huang et al., 2025b; 2024), or build MLLM-based routers to select a suitable detector (Huang et al., 2025a). Adding visual features typically requires further alignment training; existing methods rely on costly reasoning data obtained through GPT-4o (Huang et al., 2025e; Xu et al., 2024b), human annotation (Ji et al., 2025b; Kang et al., 2025), or multiple MLLMs (Tan et al., 2025b; Zhou et al., 2025).

Unlike approaches that build a single MLLM-based detector, EvoGuard uses an MLLM agent to invoke heterogeneous detectors as tools, capturing comprehensive visual cues beyond the limits of the MLLM or any single detector. Its tailored capability-aware dynamic orchestration mechanism enables dynamic tool scheduling and plug-and-play extensibility, offering a more intelligent and flexible alternative to conventional single-model or ensemble paradigms. Moreover, EvoGuard is trained purely with RL to elicit autonomous planning and reflection for AIGI detection without expensive fine-grained annotations. Table 1 compares EvoGuard with other recent methods.

3 Methodology

3.1 Overview

Practical AIGI detection must simultaneously achieve strong generalization across diverse and evolving generative models, remain extensible, and avoid reliance on costly fine-grained annotations. To address these challenges, our agentic paradigm follows three key principles: (1) encapsulate heterogeneous detectors as tools and let an MLLM-based agent autonomously plan their use through action and reflection; (2) leverage the agent's general understanding ability to handle heterogeneous tool outputs and support seamless integration of future tools; and (3) train the agent via Agentic RL using only binary labels.

Following these principles, we present EvoGuard. As shown in fig. 2, it consists of two key components: Capability-Aware Selection and Dynamic Orchestration. Capability-Aware Selection encapsulates existing detectors as tools and constructs tool profiles to select suitable tools for each input image (section 3.2). Dynamic Orchestration enables the agent to iteratively analyze previous tool outputs and decide whether to invoke additional tools or produce the final prediction (section 3.3). We train EvoGuard with GRPO using specially designed rewards and only image-level binary labels

(section 3.4). The resulting framework also supports plug-and-play extension to future detectors without retraining (section 3.5).

3.2 Capability-Aware Selection

The first key to the design of EvoGuard lies in the intelligent invocation of appropriate tools to obtain effective results while mitigating the negative influence of suboptimal outputs. To this end, it is essential to recognize the disparities among different tools and establish a paradigm for matching suitable tools. We therefore design a profile for each tool to characterize its capabilities, thereby representing the differences. Subsequently, we encourage the agent to select tools by integrating the input image with the tool profiles.

Tool Profile Design. The profile comprises two layers: *structure*, which specifies the dimensions used to characterize a tool, and *content*, which provides concrete descriptors within each dimension. We define four *structure* dimensions: *Overall Profile*, *Strengths*, *Weaknesses*, and *Conflict Hints*. The *Overall Profile* summarizes behavioral tendencies (e.g., models trained to detect generative artifacts tend to be conservative, often overlooking unseen artifacts (Liu et al., 2026)). *Strengths* and *Weaknesses* indicate data types where the tool performs particularly well or poorly. *Conflict Hints* support decision-making under disagreement by emphasizing a tool's relative advantage on the given sample type rather than its overall performance. Together, these dimensions capture both holistic and fine-grained characteristics, covering absolute and relative capabilities, thereby establishing a comprehensive framework for tool description.

The *content* layer selects criteria to distinguish images so that capability differences are exposed; it should be neither too coarse nor too granular. We use three tag dimensions—*Subject*, *Quality*, and *Style*—and constrain the candidate tag set to facilitate profile generation and tool selection.

Tool Profile Generation. We generate tool profiles based on two principles. 1) *Conciseness*. We only describe cases where metric values (accuracy, false negative rate, and false positive rate) are notably high or low on specific data types, omitting ordinary cases, as excessive context length degrades performance (Du et al., 2025) and increases inference costs. 2) *Ambiguity*. Rather than specifying exact numerical values, we use linguistic expressions to convey the magnitude of metrics. This is because the training set distribution inevitably differs from real-world applications, and LLM-driven agents possess inherent reasoning capabilities; thus, maintaining ambiguity enables dynamic decision-making based on actual sample characteristics.

In practice, we generate tool profiles using the training set. We employ the same underlying MLLM of EvoGuard to assign tags to each image, then compute each tool's metrics under each tag. Based on these metrics, we use LLM to produce an initial tool profile and then manually refine it. Given a tool set $\mathbb{T} = \{t_1, t_2, \cdots, t_M\}$, we obtain the tool profile set $\mathbb{P} = \{p_1, p_2, \cdots, p_M\}$. We then use this $\mathbb{P}$ for training and inference. More details can be found in the appendix.

Tool Selection. To determine the tools for a given image x, EvoGuard uses its backbone MLLM to obtain its tag $g(x)$. Then, at dialogue round n, EvoGuard selects a tool set $\mathbb{T}(n)$ based on the dialogue context c_{n-1} available before this round, the image x and its tag $g(x)$, and the tool profile set $\mathbb{P}$:

$$\mathbb{T}(n) = \text{Select}(x, g(x), c_{n-1}, \mathbb{P}). \tag{1}$$

3.3 Dynamic Orchestration

Capability-Aware Selection decides which tool(s) to invoke at each round. We do not constrain the interaction to a single round, as tools may produce conflicting conclusions or low-confidence predictions. We therefore introduce a Dynamic Orchestration mechanism to better leverage the agent's ability to autonomously plan actions. After each tool invocation, the agent reflects on the current evidence and decides whether to call more tools or to conclude. When tool outputs conflict or a prediction is of low confidence, the agent consults tool profiles to judge which tool is more credible on the current sample, and if still inconclusive, tends to invoke a complementary tool to gather additional evidence.

Specifically, at round n, after invoking $\mathbb{T}(n)$, EvoGuard obtains the tool outputs O_n. Given c_{n-1} and O_n, EvoGuard analyzes the current state and produces an analysis h_n and an action $a_n \in$

$\{\texttt{stop}, \texttt{continue}\}$. The context c_n is also updated by incorporating O_n and h_n:

$$(h_n, a_n) = \text{Analyze}(x, g(x), c_{n-1}, O_n, \mathbb{P}), \qquad c_n = \{c_{n-1}, O_n, h_n\}. \tag{2}$$

Afterwards, if $a_n = \texttt{continue}$, we proceed to round $n+1$, where Eq. (1) performs tool selection using the updated context c_n; otherwise, the agent synthesizes the evidence accumulated across rounds, reasoning in light of the tool profiles (especially the Conflict Hints) and the image content, to resolve the conflicting predictions and produce the final prediction $\text{Answer}(x)$:

$$\text{Answer}(x) = \text{Conclude}(x, g(x), c_n, \mathbb{P})\,. \tag{3}$$

Enabling reflection and additional actions essentially improves performance on complex cases via test-time scaling (Snell et al., 2024; Xia et al., 2025). Compared with selecting a single tool (Huang et al., 2025a) or ensembling all tools (Fang et al., 2025), our approach preserves the benefits of multi-tool collaboration while mitigating interference and adaptively controlling cost based on sample-specific runtime behavior. Notably, the MLLM's general understanding capability enables the fusion of heterogeneous outputs from different detectors, including numerical scores, text, and images, which is crucial for extending to future tools. Meanwhile, such auxiliary signals beyond binary labels further help the MLLM make more informed decisions.

3.4 Agentic RL Training

Under our paradigm, training focuses on learning effective tool use rather than improving a single detector. This leads us to Agentic Reinforcement Learning (Agentic RL), which reframes LLMs from passive sequence generators into autonomous, decision-making agents embedded in complex, dynamic worlds (Zhang et al., 2025a). Since reasoning and tool-use trajectories are highly diverse and fine-grained SFT trajectories are costly to obtain, we adopt GRPO (Shao et al., 2024), a widely used policy optimization algorithm in Agentic RL (Zhang et al., 2025a), to encourage autonomous exploration. Such a paradigm aligns closely with our task setting, and reinforcement learning has also been shown to provide stronger generalization than SFT (Chu et al., 2025).

For reward design, in addition to the commonly used outcome-correctness $\text{Reward}_{\text{acc}}$ and format-compliance $\text{Reward}_{\text{fmt}}$ rewards (Guo et al., 2025), we introduce an explicit thinking reward $\text{Reward}_{\text{analysis}}$. Motivated by prior findings that moderate reasoning can improve performance (Jin et al., 2024; Wu et al., 2025b) and our observation that insufficient deliberation degrades performance, we design $\text{Reward}_{\text{analysis}}$ to penalize insufficient analysis before tool use and the final conclusion, while leaving the reasoning form unconstrained. Accordingly, our reward function $\text{R}(\cdot)$ is formulated as:

$$\text{R}(\text{traj}(x)) = \text{R}_{\text{acc}}\big(\text{gt}(x), \text{Answer}(x)\big) + \text{R}_{\text{fmt}}\big(\text{traj}(x)\big) + \text{R}_{\text{analysis}}\big(\text{traj}(x)\big). \tag{4}$$

where $\text{gt}(x)$ is the ground-truth label and $\text{traj}(x)$ denotes the trajectory produced by the agent for input x. More details about the reward are provided in the appendix.

3.5 Extending to Future Tools

EvoGuard is inherently designed to extend to additional detectors. First, we encapsulate heterogeneous detectors as tools under a unified schema, so adding or removing tools requires only a prompt update, enabling true plug-and-play integration. Second, we leverage an MLLM to interpret tool outputs, which naturally supports diverse return formats.

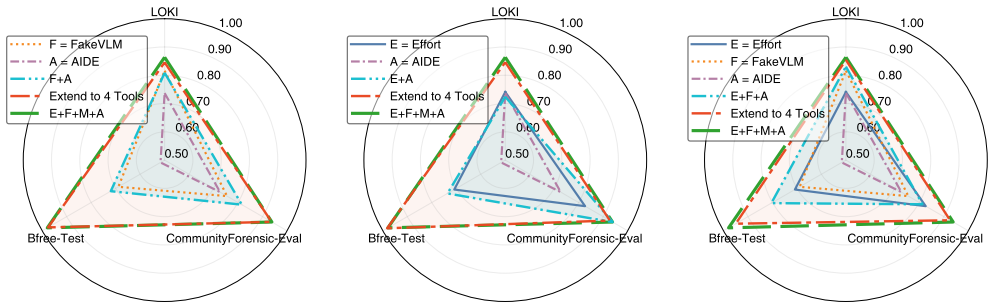
For newly added tools, our experiments show that a lightweight profile describing only overall behavior (drafted from prior knowledge) is already capable of improving performance (section 4.3). While a comprehensive profile would evaluate the tool on a given dataset, this still avoids retraining and redeploying the agent, making the extension substantially more convenient.

4 Experiments

To validate the effectiveness and advantages of the proposed method, we design our experiments to answer three questions: (1) Can EvoGuard more effectively detect AIGI than existing SOTA methods (section 4.2)? (2) Can EvoGuard be easily extended to future tools? This extensibility is crucial for rapidly responding to newly emerging real-world threats (section 4.3). (3) What are the

Table 2: Performance comparison of various methods on LOKI, CommunityForensic-Eval, and Bfree-Test datasets. Method categories are indicated by distinct markers with shaded row colors: ▲ baselines used as tools, ● other baselines, and ■ Pretrained MLLMs. EvoGuard uses Effort, FakeVLM, and AIDE as its primary tool set ("3 tools"), with MIRROR added as a fourth tool in the extended setting. Each EvoGuard variant is trained with its corresponding tool set. The best and second-best results are highlighted in **bold** and underlined, respectively. B-Acc: balanced accuracy, R-Acc: real accuracy, F-Acc: fake accuracy, R/F Acc Gap: average absolute difference between R-Acc and F-Acc across datasets. Based on Qwen3-VL-4B-Instruct, EvoGuard improves upon its constituent tools in performance while maintaining a better real/fake balance, and achieves strong overall performance among all compared methods.

Methods	LOKI			CommunityForensic-Eval			Bfree-Test			Average		
	B-Acc	F1	R-Acc / F-Acc	B-Acc	F1	R-Acc / F-Acc	B-Acc	F1	R-Acc / F-Acc	B-Acc	F1	R/F Acc Gap
▲ Effort (ICML'25)	0.7409	0.7832	0.711 / 0.771	0.8260	0.8205	0.802 / 0.850	0.7083	0.6235	0.954 / 0.463	0.7584	0.7424	0.200
▲ FakeVLM (NeurIPS'25)	0.8162	0.8613	0.743 / 0.889	0.7548	0.7763	0.569 / 0.941	0.6889	0.8586	0.417 / 0.961	0.7533	0.8321	0.354
▲ AIDE (ICLR'25)	0.7370	0.7198	0.857 / 0.617	0.7264	0.6902	0.819 / 0.634	0.5168	0.1558	0.947 / 0.087	0.6601	0.5219	0.428
▲ MIRROR (arXiv'26)	0.8523	0.8643	0.882 / 0.822	0.9332	0.9288	0.975 / 0.891	0.9736	0.9776	0.983 / 0.964	0.9197	0.9236	0.054
● FakeShield (ICLR'25)	0.6155	0.7148	0.476 / 0.756	0.7777	0.7942	0.607 / 0.949	0.6331	0.7423	0.550 / 0.717	0.6754	0.7504	0.263
● SIDA (CVPR'25)	0.6192	0.5999	0.731 / 0.507	0.5336	0.3839	0.768 / 0.299	0.6718	0.5633	0.940 / 0.403	0.6082	0.5157	0.410
● FakeReasoning (TIP'26)	0.6478	0.6901	0.639 / 0.657	0.8038	0.7892	0.836 / 0.772	0.6684	0.8015	0.513 / 0.824	0.7067	0.7603	0.131
● Forensic-MOE (ICCV'25)	0.7492	0.7705	0.776 / 0.723	0.6867	0.6813	0.660 / 0.713	0.4940	0.0155	0.980 / 0.008	0.6433	0.4891	0.359
■ Qwen3-VL-4B	0.534	0.338	0.843 / 0.226	0.606	0.555	0.698 / 0.514	0.694	0.683	0.828 / 0.561	0.6113	0.5253	0.356
■ Qwen3-VL-8B	0.522	0.091	0.997 / 0.048	0.595	0.346	0.974 / 0.215	0.660	0.516	0.967 / 0.353	0.5923	0.3177	0.774
■ Qwen3-VL-30B-A3B	0.627	0.534	0.853 / 0.401	0.770	0.738	0.868 / 0.672	0.928	0.925	0.993 / 0.863	0.7750	0.7323	0.259
■ DeepSeek-V4.1-Flash (552B)	0.692	0.680	0.799 / 0.585	0.806	0.795	0.815 / 0.796	0.888	0.928	0.848 / 0.928	0.7953	0.8010	0.104
EvoGuard (Ours, 4B, 3 tools)	0.8035	0.8465	0.747 / 0.860	0.8125	0.8202	0.680 / 0.945	0.8044	0.8367	0.832 / 0.777	0.8068	0.8345	0.144
EvoGuard (Ours, 4B, 4 tools)	**0.8638**	**0.8824**	0.867 / 0.861	**0.9380**	**0.9344**	0.943 / 0.933	**0.9792**	**0.9843**	0.980 / 0.978	**0.9270**	**0.9337**	**0.006**



(a) FakeVLM + AIDE (b) Effort + AIDE (c) FakeVLM + Effort + AIDE

Figure 3: Results of EvoGuard trained on different tool subsets then evaluated on the same tool subsets (F+A/E+A/F+E+A) and on full tool set (Extend to 4 Tools), and EvoGuard trained and evaluated on full tool set (E+F+M+A). The metric is Balanced Accuracy. Every group achieves a performance improvement through train-free extension.

individual contributions of the components in EvoGuard (section 4.4)? We first introduce the setup of experiments, and then present results and analysis. More details on the experimental settings, computation costs, tool-call distributions, robustness evaluations, and additional results can be found in the appendix.

4.1 Experiment Setup

Tool Set. We use three recently published SOTA methods with complementary architectures and visual cues as the primary tool set: Effort (Yan et al., 2025), a CLIP-based detector that freezes the principal components of the feature space to prevent overfitting; FakeVLM (Wen et al., 2025), a VLM-based detector trained with fine-grained artifact-explanation supervision; and AIDE (Yan et al., 2024), which combines DCT-based frequency and CLIP-based semantic features. We further introduce MIRROR (Liu et al., 2026), which measures the perplexity and residual of DINO-based (Oquab et al., 2023) manifold-consistent references.

Datasets. To simulate in-the-wild tool use while avoiding the training data of the selected tools, we construct an 8,000-image mixed training set. Using seed 42, we shuffle each source and select 1,000 real and 1,000 fake images from each of the MMFR training set, MMFR test set (Gao et al., 2025), and DDA-COCO (Chen et al., 2025b), together with 2,000 fake images from SynthScars (Kang et al., 2025).

Table 3: Ablation results on LOKI, CommunityForensic-Eval, and Bfree-test. Markers and shaded rows distinguish the ablation of Capability-Aware Selection (▲), training (◆), and Dynamic Orchestration (●), as well as comparisons with other MoE methods (■). The best performance is in **bold**, while the second-best is underlined. B-Acc: balanced accuracy. ⊕ denotes logical OR.

Method	LOKI		CommunityForensic-Eval		Bfree-Test		Average	
	B-Acc	F1	B-Acc	F1	B-Acc	F1	B-Acc	F1
EvoGuard (4 tools)	**0.8638**	**0.8824**	**0.9380**	**0.9344**	**0.9792**	**0.9843**	**0.9270**	**0.9337**
▲ EvoGuard (w/o Tool Profile)	0.7900	0.8392	0.8590	0.8606	0.9523	0.9696	0.8671	0.8898
▲ Match Best Tools	0.8150	0.8173	0.9330	0.9302	0.9736	0.9776	0.9072	0.9084
◆ EvoGuard (w/o RL Training)	0.7067	0.8132	0.8201	0.8283	0.7592	0.8959	0.7620	0.8458
◆ EvoGuard (w/o Analysis Reward)	0.8505	0.8684	0.8933	0.8883	0.9643	0.9782	0.9027	0.9116
● EvoGuard (select one tool)	0.8513	0.8639	0.9332	0.9287	0.9755	0.9786	0.9200	0.9237
● EvoGuard (invoke all tools)	0.8417	0.8603	0.9138	0.9098	0.9408	0.9634	0.8988	0.9112
● EvoGuard (one round)	0.8316	0.8509	0.9181	0.9139	0.9704	0.9780	0.9067	0.9143
■ MoE by Gate Network	0.7419	0.8229	0.9113	0.9072	0.9660	0.9763	0.8731	0.9021
■ Weighted Majority Vote	0.8496	0.8598	0.9227	0.9183	0.9601	0.9645	0.9108	0.9142
■ Confidence Averaging	0.8462	0.8563	0.8828	0.8805	0.9586	0.9628	0.8959	0.8999
■ Calibrated Score Average	0.7893	0.8567	0.8016	0.8175	0.8854	0.9433	0.8254	0.8725
■ Calibrated Weighted Vote	0.7988	0.8588	0.8144	0.8268	0.9158	0.9588	0.8430	0.8815
■ FakeVLM ⊕ MIRROR	0.8003	0.8671	0.7635	0.7883	0.7047	0.8744	0.7562	0.8433

For evaluation, we use three in-the-wild benchmarks: LOKI (Ye et al., 2025), which is a recent comprehensive benchmark covering multiple image sources with broad content diversity; Bfree-Test (Guillaro et al., 2025), which contains different versions of the same images collected from social networks, making it suitable for evaluating robustness; and CommunityForensic (Park & Owens, 2025), which includes images from thousands of generators. Due to its scale, we sample a subset of it. Together, these three datasets form a strong benchmark for effectively assessing the performance of EvoGuard and baseline methods in complex and diverse in-the-wild settings.

Implementation Details. EvoGuard uses Qwen3-VL-4B-Instruct (Bai et al., 2025) as the base model. We conduct GRPO training with multi-round tool calls using the AgentLoop module in the VeRL framework (Sheng et al., 2025), with the learning rate set to 1×10^{-6} and the KL loss coefficient to 0.001. Only the agent is trained, while all tools remain frozen.

Comparative Methods. In order to ensure a comprehensive comparison, the evaluated methods include four existing tools, other SOTA detection methods including FakeShield (Xu et al., 2024b), SIDA (Huang et al., 2025d), FakeReasoning (Gao et al., 2025), and Forensic-MOE (Fang et al., 2025), and pretrained MLLMs of different size, forming a diverse benchmark for comparison. For all competing methods, we use the officially released checkpoints from their authors.

Metrics. We employ commonly used evaluation metrics Balanced Accuracy (B-Acc) and F1 score, and accuracy on real and fake images (R-Acc / F-Acc).

4.2 Performance Comparison

Generalization on In-the-Wild Benchmarks. Table 2 presents the performance comparison between EvoGuard and comparative methods. With its primary three tools, EvoGuard improves average B-Acc by nearly 5 points over the best individual tool and exceeds every other baseline except MIRROR, including much larger pretrained MLLMs. Although MIRROR already achieves 91.97% average B-Acc, EvoGuard further improves performance across all three benchmarks with MIRROR in its tool set. These results demonstrate that EvoGuard effectively integrates different detectors and leverages their complementary strengths to achieve gains beyond any individual tool.

Mitigating Detector Bias. EvoGuard consistently reduces the average R/F Acc Gap of its constituent tools in both settings. While many existing detectors exhibit severe class bias, EvoGuard narrows the real/fake accuracy gap by 28–66% in the three-tool setting, and by 9–70× after adding MIRROR (table 2). This shows that capability-aware dynamic orchestration can exploit complementary evidence to mitigate detector bias, with the degree of mitigation depending on the available tools. Such biases may arise from spurious features (Chen et al., 2025b; Guillaro et al., 2025) or limitations of existing detection paradigms (Liu et al., 2026) .

Fusion of Heterogeneous Detectors. Among the 4 selected tools, FakeVLM returns only textual outputs, while the other three return confidence scores. Leveraging the general reasoning ability of MLLMs, EvoGuard can reason over outputs in different formats, which is also crucial for future extensibility. Detailed examples are provided in the appendix.

4.3 Extensibility to Future Tools

The rapid evolution of generative models and forgery methods requires detectors to continuously evolve as well. A detector that can improve efficiently and continuously would be highly valuable. An important design goal of EvoGuard is to scale to more tools without training. In this way, when new detection tools emerge in the future, EvoGuard can incorporate them directly, thereby improving its ability to handle novel forgeries.

To investigate the effectiveness of this design, we train EvoGuard on a subset of tools and evaluate whether introducing held-out tools only at inference time improves performance. For newly introduced tools, we supply only the *Overall Profile*, mimicking a realistic setting where only coarse information about a new tool is available. Since the number of possible subsets is large, we construct representative 3 tool combinations according to two principles: (1) selecting heterogeneous detectors to exploit complementarity, and (2) simulating the introduction of newer, stronger detectors. The results are shown in fig. 3.

Train-free Extensibility. As shown, agents trained on each tool subset benefit from the addition of previously unseen tools at test time. This train-free tool extension yields substantial gains, bringing performance close to that of an agent trained with the full tool set. These results suggest that EvoGuard effectively learns to leverage diverse tools during training and can generalize this capability to newly introduced tools.

4.4 Ablation & Mechanism Analysis

We further analyze the core design of EvoGuard through multi-dimensional ablations with the full tool set and comparisons with routing-based and other ensemble methods, as summarized in table 3.

Capability-Aware Selection and Training. Removing the Tool Profile reduces performance across all three benchmarks, showing the value of explicit guidance about tool capabilities. We also test using an image's three-dimensional tags to match three best-performing tools and taking a majority vote, which seems more precise, yet underperforms EvoGuard, likely due to train–test data shift. This suggests that the deliberate ambiguity in tool profile helps it generalize. Then, without RL training or removing the analysis reward, performance also drops even though the base model and prompts remain the same. These results show that capability guidance and training both contribute to effective tool use.

Dynamic Orchestration and Other MoE Methods. Restricting EvoGuard to selecting one tool, invoking all tools, or using only one round reduces performance in every case, supporting the benefit of adapting both tool selection and the inference process to the image. Meanwhile, EvoGuard outperforms all compared methods on both B-Acc and F1 across the three benchmarks, including learned gating, voting, score averaging, and logical OR. The results demonstrate the advantage of reasoning over tool outputs and coordinating their use according to the input. Implementation details of the compared methods and illustrative cases are provided in the appendix.

5 Discussion and Conclusion

Conclusion. We present EvoGuard, an agentic framework that recasts AIGI detection as reasoning-based evidence synthesis over a pool of heterogeneous detectors. Going beyond single detectors and conventional ensembles, EvoGuard reasons over their heterogeneous outputs, cross-validating conflicting or low-confidence signals before concluding. Trained via Agentic RL with only low-cost binary labels, it achieves SOTA accuracy across diverse benchmarks while markedly mitigating the real/fake class bias. Most importantly, it supports plug-and-play integration of new tools without retraining, enabling continual evolution against emerging threats.

Discussion. While EvoGuard offers a practical and sustainable solution, it also introduces additional MLLM overhead and remains bounded by its underlying toolset: if all tools fail, its reasoning may also be misled.

Future Work. Future directions include (1) developing task-specific small MLLMs via distillation (Xu et al., 2024a), (2) continually updating tools and profiles with real-world samples, and (3) extending EvoGuard to broader visual forensics tasks to explore the potential of this paradigm.

AI USE STATEMENT

In this work, we used generative AI tools to improve the readability, conciseness, grammar, and translation quality of the manuscript while preserving its original meaning, and to assist with parts of the experimental code. We did not employ generative AI tools to create synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims or contribute to proofs, propose or refine research hypotheses, design or provide feedback on the research methodology or experiments, implement the proposed method, clean or reformat datasets, conduct qualitative or thematic data analysis, or interpret experimental results. The authors reviewed and revised all AI-assisted changes to the manuscript and manually inspected and tested all AI-assisted code to verify its correctness. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide the source code for the training and evaluation pipelines in the supplementary material, together with detailed instructions for reproducing our experiments. The frameworks and datasets used in our experiments are all obtained from publicly available open-source repositories, and their sources are documented in the accompanying code to facilitate access and reproduction. For all external models, we use the official checkpoints released by the original authors. We commit to publicly releasing our code upon acceptance of the paper.

REFERENCES

Mohamad Abou Ali, Fadi Dornaika, and Jinan Charafeddine. Agentic ai: a comprehensive survey of architectures, applications, and future directions. *Artificial Intelligence Review*, 59(1):11, 2025. 2

Aref Azizpour, Tai D. Nguyen, Manil Shrestha, Kaidi Xu, Edward Kim, and Matthew C. Stamm. E3: Ensemble of expert embedders for adapting synthetic image detectors to new generators using limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4334–4344, June 2024. 3

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. 8

Yu Cai, Shan Jia, Jiahe Tian, Jiao Dai, Jizhong Han, and Siwei Lyu. Cataid: Category-guided ai-generated image detection via vision-language model adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1553–1563, 2025. 3

Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European conference on computer vision*, pp. 103–120. Springer, 2020. 3

Baoying Chen, Jishen Zeng, Jianquan Yang, and Rui Yang. Drct: Diffusion reconstruction contrastive training towards universal detection of diffusion generated images. In *Forty-first International Conference on Machine Learning*, 2024a. 3

Ruoxin Chen, Jiahui Gao, Kaiqing Lin, Keyue Zhang, Yandan Zhao, Isabel Guan, Taiping Yao, and Shouhong Ding. Task-model alignment: A simple path to generalizable ai-generated image detection. *arXiv preprint arXiv:2512.06746*, 2025a. 3

Ruoxin Chen, Junwei Xi, Zhiyuan Yan, Ke-Yue Zhang, Shuang Wu, Jingyi Xie, Xu Chen, Lei Xu, Isabel Guan, Taiping Yao, et al. Dual data alignment makes ai-generated image detector easier generalizable. *arXiv preprint arXiv:2505.14359*, 2025b. 3, 7, 8

Yize Chen, Zhiyuan Yan, Guangliang Cheng, Kangran Zhao, Siwei Lyu, and Baoyuan Wu. X2-dfd: A framework for explainable and extendable deepfake detection. *arXiv preprint arXiv:2410.06126*, 2024b. 4

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025. 6

Alex Costanzino, Woody Bayliss, Juil Sock, Marc Gorriz Blanch, Danijela Horak, Ivan Laptev, Philip Torr, and Fabio Pizzati. Towards reliable identification of diffusion-based image manipulations. *arXiv preprint arXiv:2506.05466*, 2025. 3

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A Huerta, and Hao Peng. Context length alone hurts llm performance despite perfect retrieval. *arXiv preprint arXiv:2510.05381*, 2025. 5

David C Epstein, Ishan Jain, Oliver Wang, and Richard Zhang. Online detection of ai-generated images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 382–392, 2023. 3

Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. *arXiv preprint arXiv:2410.13863*, 2024. 2

Mingqi Fang, Ziguang Li, Lingyun Yu, Quanwei Yang, Hongtao Xie, and Yongdong Zhang. Forensic-moe: Exploring comprehensive synthetic image detection traces with mixture of experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17772–17782, 2025. 3, 6, 8, 18

Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pp. 3247–3258. PMLR, 2020. 3

Yueying Gao, Dongliang Chang, Bingyao Yu, Haotian Qin, Lei Chen, Kongming Liang, and Zhanyu Ma. Fakereasoning: Towards generalizable forgery detection and reasoning. *arXiv preprint arXiv:2503.21210*, 2025. 2, 4, 7, 8

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

Fabrizio Guillaro, Giada Zingarini, Ben Usman, Avneesh Sud, Davide Cozzolino, and Luisa Verdoliva. A bias-free training paradigm for more general ai-generated image detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18685–18694, 2025. 3, 8

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 6

Anna Yoo Jeong Ha, Josephine Passananti, Ronik Bhaskar, Shawn Shan, Reid Southen, Haitao Zheng, and Ben Y Zhao. Organic or diffused: Can we distinguish human art from ai-generated images? In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 4822–4836, 2024. 2

Xinan He, Yue Zhou, Bing Fan, Bin Li, Guopu Zhu, and Feng Ding. Vlforgery face triad: Detection, localization and attribution via multimodal large language models. *arXiv preprint arXiv:2503.06142*, 2025. 4

Qing Huang, Zhipei Xu, Xuanyu Zhang, and Jian Zhang. Unishield: An adaptive multi-agent framework for unified forgery image detection and localization. *arXiv preprint arXiv:2510.03161*, 2025a. 4, 6

Qing Huang, Xiangyu Yu, and Zhipei Xu. Mm-iml: Multi-modal image forgery detection and localization. In *2025 IEEE International Conference on Image Processing (ICIP)*, pp. 1588–1593. IEEE, 2025b. 4

Tai-Ming Huang, Wei-Tung Lin, Kai-Lung Hua, Wen-Huang Cheng, Junichi Yamagishi, and Jun-Cheng Chen. Thinkfake: Reasoning in multimodal large language models for ai-generated image detection. *arXiv preprint arXiv:2509.19841*, 2025c. 4

Zhengchao Huang, Bin Xia, Zicheng Lin, Zhun Mou, Wenming Yang, and Jiaya Jia. Ffaa: Multimodal large language model based explainable open-world face forgery analysis assistant. *arXiv preprint arXiv:2408.10072*, 2024. 4

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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28831–28841, 2025d. 4, 8

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*, 2025e. 2, 4

Yikun Ji, Yan Hong, Jiahui Zhan, Haoxing Chen, Jun Lan, Huijia Zhu, Weiqiang Wang, Liqing Zhang, and Jianfu Zhang. Towards explainable fake image detection with multi-modal large language models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 4398–4407, 2025a. 3

Yikun Ji, Hong Yan, Jun Lan, Huijia Zhu, Weiqiang Wang, Qi Fan, Liqing Zhang, and Jianfu Zhang. Interpretable and reliable detection of ai-generated images via grounded reasoning in mllms. *arXiv preprint arXiv:2506.07045*, 2025b. 4

Shan Jia, Reilin Lyu, Kangran Zhao, Yize Chen, Zhiyuan Yan, Yan Ju, Chuanbo Hu, Xin Li, Baoyuan Wu, and Siwei Lyu. Can chatgpt detect deepfakes? a study of using multimodal large language models for media forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4324–4333, 2024. 2, 4

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1830–1842, 2024. 6

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18937–18947, 2025. 2, 3, 4, 7

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023. 3

Chenqi Kong, Anwei Luo, Peijun Bao, Yi Yu, Haoliang Li, Zengwei Zheng, Shiqi Wang, and Alex C Kot. Moe-ffd: Mixture of experts for generalized and parameter-efficient face forgery detection. *IEEE Transactions on Dependable and Secure Computing*, 2025. 3

Songlin Li, Zhiqing Guo, Yuanman Li, Zeyu Li, Yunfeng Diao, Gaobo Yang, and Liejun Wang. Bridging semantic logic gaps: A cognition inspired multimodal boundary preserving network for image manipulation localization. *arXiv preprint arXiv:2508.07216*, 2025a. 4

Tianxiao Li, Zhenglin Huang, Haiquan Wen, Yiwei He, Shuchang Lyu, Baoyuan Wu, and Guangliang Cheng. Raidx: A retrieval-augmented generation and grpo reinforcement learning framework for explainable deepfake detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 11746–11755, 2025b. 3

Yixuan Li, Xuelin Liu, Xiaoyang Wang, Bu Sung Lee, Shiqi Wang, Anderson Rocha, and Weisi Lin. Fakebench: Probing explainable fake image detection via large multimodal models. *IEEE Transactions on Information Forensics and Security*, 2025c. 4

Ziqiang Li, Jiazhen Yan, Ziwen He, Kai Zeng, Weiwei Jiang, Lizhi Xiong, and Zhangjie Fu. Is artificial intelligence generated image detection a solved problem? *arXiv preprint arXiv:2505.12335*, 2025d. 2

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*, 2024a. 4

Jinlai Liu, Jian Han, Bin Yan, Hui Wu, Fengda Zhu, Xing Wang, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Infinitystar: Unified spacetime autoregressive modeling for visual generation. *arXiv preprint arXiv:2511.04675*, 2025a. 2

Ruiqi Liu, Yi Han, Zhengbo Zhang, Liwei Yao, Zhiyuan Yan, Jialiang Shen, ZhiJin Chen, Boyi Sun, Lubin Weng, Jing Dong, et al. Beyond artifacts: Real-centric envelope modeling for reliable ai-generated image detection. *arXiv preprint arXiv:2512.20937*, 2025b. 3

Ruiqi Liu, Manni Cui, Ziheng Qin, Zhiyuan Yan, Ruoxin Chen, Yi Han, Zhiheng Li, Junkai Chen, ZhiJin Chen, Kaiqing Lin, et al. Mirror: Manifold ideal reference reconstructor for generalizable ai-generated image detection. *arXiv preprint arXiv:2602.02222*, 2026. 3, 5, 7, 8

Zihan Liu, Hanyi Wang, Yaoyu Kang, and Shilin Wang. Mixture of low-rank experts for transferable ai-generated image detection. *arXiv preprint arXiv:2404.04883*, 2024b. 3

Tuan Nguyen, Naseem Khan, Khang Tran, NhatHai Phan, and Issa Khalil. Prpo: Paragraph-level policy optimization for vision-language deepfake detection. *arXiv preprint arXiv:2509.26272*, 2025. 4

Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24480–24489, 2023. 2, 3

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 3, 7

Jeongsoo Park and Andrew Owens. Community forensics: Using thousands of generators to train fake image detectors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8245–8257, 2025. 3, 8

Rongxuan Peng, Shunquan Tan, Chenqi Kong, Anwei Luo, Alex C Kot, and Jiwu Huang. Forensic-ssam: Toward robust and unified image forgery detection and localization resisting to adversarial attack. *arXiv preprint arXiv:2508.07402*, 2025a. 3

Siran Peng, Zipei Wang, Li Gao, Xiangyu Zhu, Tianshuo Zhang, Ajian Liu, Haoyuan Zhang, and Zhen Lei. Mllm-enhanced face forgery detection: A vision-language fusion solution. *arXiv preprint arXiv:2505.02013*, 2025b. 4

Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision*, pp. 86–103. Springer, 2020. 3

Lixiong Qin, Yang Zhang, Mei Wang, Jiani Hu, Weihong Deng, and Weiran Xu. Fake-in-facext: Towards fine-grained explainable deepfake analysis. *arXiv preprint arXiv:2510.20531*, 2025. 4

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021. 3

Anirudh Sundara Rajan, Utkarsh Ojha, Jedidiah Schloesser, and Yong Jae Lee. Aligned datasets improve detection of latent diffusion-generated images. *arXiv preprint arXiv:2410.11835*, 2024. 3

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022. 2

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 2, 6

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025. 8

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. 6

Haoran Sun, Chen Cai, Huiping Zhuang, Kong Aik Lee, Lap-Pui Chau, and Yi Wang. Edvd-llama: Explainable deepfake video detection via multimodal large language model reasoning. *arXiv preprint arXiv:2510.16442*, 2025a. 4

Ke Sun, Shen Chen, Taiping Yao, Ziyin Zhou, Jiayi Ji, Xiaoshuai Sun, Chia-Wen Lin, and Rongrong Ji. Towards general visual-linguistic face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19576–19586, 2025b. 4

Zhihao Sun, Haoran Jiang, Haoran Chen, Yixin Cao, Xipeng Qiu, Zuxuan Wu, and Yu-Gang Jiang. Forgerysleuth: Empowering multimodal large language models for image manipulation detection. *arXiv preprint arXiv:2411.19466*, 2024. 4

Matías Tailanián, Marina Gardella, Alvaro Pardo, and Pablo Musé. Diffusion models meet image counter-forensics. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3925–3935, 2024. 2

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 28130–28139, 2024. 2, 3

Chuangchuang Tan, Xiang Ming, Jinglu Wang, Renshuai Tao, Bin Li, Yunchao Wei, Yao Zhao, and Yan Lu. Semantic visual anomaly detection and reasoning in ai-generated images. *arXiv preprint arXiv:2510.10231*, 2025a. 4

Hao Tan, Jun Lan, Zichang Tan, Ajian Liu, Chuanbiao Song, Senyuan Shi, Huijia Zhu, Weiqiang Wang, Jun Wan, and Zhen Lei. Veritas: Generalizable deepfake detection via pattern-aware reasoning. *arXiv preprint arXiv:2508.21048*, 2025b. 4

Diangarti Tariang, Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. Synthetic image verification in the era of generative artificial intelligence: What works and what isn't there yet. *IEEE Security & Privacy*, 22(3):37–49, 2024. 2

Hongyang Wang, Yichen Shi, Zhuofu Tao, Yuhao Gao, Liepiao Zhang, Xun Lin, Jun Feng, Xiaochen Yuan, Zitong Yu, and Xiaochun Cao. Faceshield: Explainable face anti-spoofing with multimodal large language models. *arXiv preprint arXiv:2505.09415*, 2025a. 4

Jiarui Wang, Huiyu Duan, Juntong Wang, Ziheng Jia, Woo Yi Yang, Xiaorong Zhu, Yu Zhao, Jiaying Qian, Yuke Xing, Guangtao Zhai, et al. Dfbench: Benchmarking deepfake image detection capability of large multimodal models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 12666–12673, 2025b. 2, 3

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024. 2

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8695–8704, 2020. 2, 3

Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22445–22455, 2023. 2, 3

Siwei Wen, Junyan Ye, Peilin Feng, Hengrui Kang, Zichen Wen, Yize Chen, Jiang Wu, Wenjun Wu, Conghui He, and Weijia Li. Spot the fake: Large multimodal model-based synthetic image detection with artifact explanation. *arXiv preprint arXiv:2503.14905*, 2025. 2, 4, 7

Yixin Wu, Feiran Zhang, Tianyuan Shi, Ruicheng Yin, Zhenghua Wang, Zhenliang Gan, Xiaohua Wang, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. Explainable synthetic image detection through diffusion timestep ensembling. *arXiv preprint arXiv:2503.06201*, 2025a. 3

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025b. 6

Cheng Xia, Manxi Lin, Jiexiang Tan, Xiaoxiong Du, Yang Qiu, Junjun Zheng, Xiangheng Kong, Yuning Jiang, and Bo Zheng. Mirage: Towards ai-generated image detection in the wild. *arXiv preprint arXiv:2508.13223*, 2025. 6

Guitao Xu, Ziqi Yi, Peirong Zhang, Jiahuan Cao, Shihang Wu, and Lianwen Jin. From pixels to semantics: A novel mllm-driven approach for explainable tampered text detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 757–766, 2025a. 4

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*, 2024a. 9

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. *arXiv preprint arXiv:2410.02761*, 2024b. 4, 8

Zhipei Xu, Xuanyu Zhang, Qing Huang, Xing Zhou, and Jian Zhang. Avatarshield: Visual reinforcement learning for human-centric synthetic video detection. *arXiv preprint arXiv:2505.15173*, 2025b. 4

Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for ai-generated image detection. *arXiv preprint arXiv:2406.19435*, 2024. 2, 3, 7

Zhiyuan Yan, Jiangming Wang, Peng Jin, Ke-Yue Zhang, Chengchun Liu, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Orthogonal subspace decomposition for generalizable ai-generated image detection. In *International Conference on Machine Learning*, pp. 70268–70288. PMLR, 2025. 2, 3, 7

Junyan Ye, Baichuan Zhou, Zilong Huang, Junan Zhang, Tianyi Bai, Hengrui Kang, Jun He, Honglin Lin, Zihao Wang, Tong Wu, et al. Loki: A comprehensive synthetic data detection benchmark using large multimodal models. *ICLR*, 2025. 8

Peipeng Yu, Jianwei Fei, Hui Gao, Xuan Feng, Zhihua Xia, and Chip Hong Chang. Unlocking the capabilities of large vision-language models for generalizable and explainable deepfake detection. *arXiv preprint arXiv:2503.14853*, 2025. 4

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025a. 6

Keyang Zhang, Chenqi Kong, Hui Liu, Bo Ding, Xinghao Jiang, and Haoliang Li. Propose and rectify: A forensics-driven mllm framework for image manipulation localization. *arXiv preprint arXiv:2508.17976*, 2025b. 2, 4

Yue Zhang, Ben Colman, Xiao Guo, Ali Shahriyari, and Gaurav Bharaj. Common sense reasoning for deepfake detection. In *European conference on computer vision*, pp. 399–415. Springer, 2024a. 4

Zicheng Zhang, Haoning Wu, Chunyi Li, Yingjie Zhou, Wei Sun, Xiongkuo Min, Zijian Chen, Xiaohong Liu, Weisi Lin, and Guangtao Zhai. A-bench: Are lmms masters at evaluating ai-generated images? *arXiv preprint arXiv:2406.03070*, 2024b. 4

Nan Zhong, Yiran Xu, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. Patchcraft: Exploring texture patch for efficient ai-generated image detection. *arXiv preprint arXiv:2311.12397*, 2023. 2

Ziyin Zhou, Yunpeng Luo, Yuanchen Wu, Ke Sun, Jiayi Ji, Ke Yan, Shouhong Ding, Xiaoshuai Sun, Yunsheng Wu, and Rongrong Ji. Aigi-holmes: Towards explainable and generalizable ai-generated image detection via multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18746–18758, 2025. 2, 4

A Appendix

A.1 Reward Function

In GRPO, the training objective is ultimately determined by the rewards assigned to sampled responses. For a given question $q(x)$, GRPO samples a group of N response trajectories from the old policy $\pi_{\theta_{\text{old}}}$:

$$\{traj(x)_1, traj(x)_2, \ldots, traj(x)_N\} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q(x)), \tag{5}$$

and assigns a scalar reward to each sampled response trajectory:

$$R_i = R(q(x), traj(x)_i), \quad i = 1, 2, \ldots, N. \tag{6}$$

The rewards are then normalized within the group to obtain the relative advantage of each response trajectory:

$$A_i = \frac{R_i - \bar{R}}{\sigma_R}, \tag{7}$$

where

$$\bar{R} = \frac{1}{N}\sum_{j=1}^{N} R_j, \sigma_R = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(R_j - \bar{R})^2 + \epsilon}. \tag{8}$$

$\epsilon > 0$ is a small constant for numerical stability. These relative advantages are subsequently used in the GRPO objective to optimize the policy under a standard clipped policy update scheme. Therefore, the reward function $R(\cdot)$ plays a central role, as it directly determines the relative preference among sampled responses and thus the optimization direction.

The reward function $R(\cdot)$ defined in eq. (4) consists of three components: R_{acc}, R_{fmt}, and $\text{R}_{\text{analysis}}$.

- R_{acc} measures whether the final prediction matches the ground-truth label ("Real" or "Fake"). We require the agent to place its final decision within the `<final answer></final answer>` tags. A reward of 1.0 is assigned only when the content exactly matches the ground-truth label.
- R_{fmt} is designed to enforce correct formatting in tool usage. Specifically, a reward of 0.2 is assigned for a correctly formatted final answer, and 0.1 is assigned if all tool calls in a trajectory are correctly formatted and enclosed within `<tool_call></tool_call>`. In contrast, a penalty of -0.5 is imposed for each distinct hallucinated tool name, and a penalty of -0.5 is also applied to trajectories that contain no tool calls.
- $\text{R}_{\text{analysis}}$ encourages the agent to reason explicitly before and after tool use. A penalty of -0.2 is applied if the number of characters generated before the first tool call is fewer than 10, and another penalty of -0.2 is applied if the number of characters between the last tool result and final answer is fewer than 20.

A.2 Dataset Details

In section 4 we describe how we construct the training set from multiple datasets. For the CommunityForensic-Eval dataset in the evaluation sets, we also curated a subset in the same manner by selecting the first 1,000 real images and the first 1,000 fake images with the same random seed (42). Since the fake subset contained duplicate entries, deduplication reduced its size to 893 images. The full list of selected image filenames is available in the code we provide.

A.3 Implementation Details

Training and evaluation were conducted on a shared server equipped with multiple GPU models, including the A100 80GB, A100 40GB, and RTX 6000 Ada. We split the training data into training and validation sets with a ratio of 9:1, trained for one epoch, and selected the checkpoint with the best validation performance. To accelerate training, we precomputed all possible tool-call outputs and stored them in an SQLite database. In practice, tool calls were implemented as database queries. For EvoGuard, we resize all images proportionally so that the longest side is 512 pixels. For

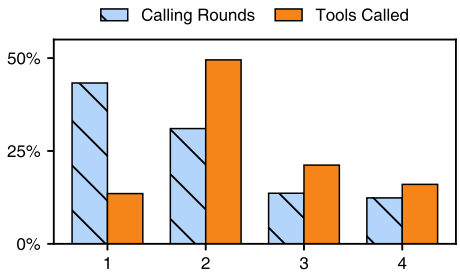


Figure 4: Distribution of interaction rounds and tool calls on the LOKI dataset of EvoGuard (4 tools). EvoGuard neither relies solely on the output of a single tool nor indiscriminately invokes all available tools.

the other baselines, we used the default image resizing settings provided in the respective official implementations. For GRPO training, we set the group size to 8 (i.e., 8 response trajectories were sampled for each question). The batch size and mini-batch size were set to 256 and 64, respectively. BF16 mixed-precision training was enabled to improve efficiency. Additional configuration details can be found in the code we provide.

A.4 Ablation Setting Details

Ablation settings. The w/o Tool Profile variant removes the tool capability descriptions provided to the agent. Match Best Tools matches an image's three-dimensional tags to the three tools with the highest corresponding accuracy and combines their predictions by majority vote. The w/o RL Training variant uses the same base model and prompts as EvoGuard without RL training. The w/o Analysis Reward variant omits the analysis reward during training. For the orchestration ablations, select one tool restricts the agent to a single tool in only one round, invoke all tools requires it to call all tools in only one round, and one round limits tool use to a single round.

Other MoE Methods. The MoE by Gate Network baseline uses a gating network whose architecture closely follows that of Fang et al. (2025), adapted to our four-tool setting. The remaining methods use the four detectors unless otherwise specified. Weighted Majority Vote combines binary predictions with weights $w_i = \log(a_i/(1-a_i))$, where a_i is each detector's accuracy on the training set. Confidence Averaging predicts fake when the mean of the four uncalibrated scores is at least 0.5. For the calibrated methods, we fit Platt calibration to the scores of detection models. Calibrated Score Average predicts fake when the mean calibrated score is at least 0.5. Calibrated Weighted Vote predicts fake when $\sum_i w_i(p_i - 0.5) \geq 0$, where p_i is the calibrated score and w_i is computed from the detector's accuracy after calibration on the training set. Finally, FakeVLM $\oplus$ MIRROR predicts fake if either detector predicts fake. All weights and calibration parameters are obtained from the training set and applied to the three test sets without test-set tuning. For score-based fusion, we encode FakeVLM's textual predictions as binary scores (Real = 0, Fake = 1).

A.5 Rounds and Tool Calls Distribution

Figure 4 shows the distribution of tool calls and interaction rounds made by EvoGuard on the LOKI dataset. Note that a single round may issue more than one tool call. Calling two tools is the most common case, followed by three tools. Approximately 75% of interactions terminate within two rounds. These patterns align with our design goal: EvoGuard neither relies solely on the output of a single tool nor indiscriminately invokes all available tools. Instead, it learns to orchestrate heterogeneous tools intelligently.

A.6 Computation Cost

A common concern about introducing MLLMs is the potential increase in computational cost. However, EvoGuard preserves the original MLLM architecture and does not require a complex training pipeline. This brings two key advantages. First, it allows us to fully leverage the latest training and

Table 4: Comparison of model size and training data scale across different MLLM-based baselines. Statistics for the baseline methods are collected from their original papers. EvoGuard is built on the smallest base model and uses the least amount of training data, highlighting that our proposed paradigm is effective without requiring large-scale training.

Method	Model Size	Training Data Scale
EvoGuard (ours)	4B	7.2k (Mixed Dataset), 1 epoch
FakeVLM (NeurIPS '25)	7B	100k (FakeClue Dataset), 2 epochs
FakeReasoning (TIP '26)	13B	312k (MMFR Dataset), 3 epochs
FakeShield (ICLR '25)	13B	34k (MMTD Dataset), 34 epochs
SIDA (CVPR '25)	7B/13B	210k (SID Dataset)

Table 5: Comparison of inference time on the LOKI dataset across different baselines. Reported time cost is GPU time on 8×A100 80G GPUs. We first adapted the original code for multi-GPU environments. Then, for all methods except those marked with *, we modified the implementation to allow batch-size tuning for efficient execution on the available hardware. Although EvoGuard invokes multiple tools to reach a decision, its cost remains practical and comparable to single MLLM-based detectors, and is lower than gated MoE baseline while achieving better detection performance.

Method	Paradigm	Time Cost
EvoGuard (ours, 4 tools)	Agentic Framework	17m35s (w/o tool execution: 4m29s)
MoE by Gated Network	Ensemble	24m9s
FakeVLM (NeurIPS '25)	MLLM-based Detector	10m36s
FakeReasoning (TIP '26)	MLLM-based Detector	8m31s
FakeShield (ICLR '25)	MLLM-based Detector	14m57s
FakeShield* (ICLR '25)	MLLM-based Detector	50m35s
SIDA* (CVPR '25)	MLLM-based Detector	2h58m
AIDE (ICLR '25)	non-MLLM-based Detector	8m57s
MIRROR (arXiv '26)	non-MLLM-based Detector	2m37s
Effort (ICML '25)	non-MLLM-based Detector	1m22s

inference acceleration techniques, such as SGLang, vLLM, and FlashAttention, thereby benefiting directly from ongoing advances in the LLM ecosystem—an important source of future scalability and efficiency gains. Second, it makes migration to newer and more capable MLLM backbones straightforward.

Training Cost. Table 4 summarizes the foundation model sizes and training data scales of different MLLM-based AIGI detection methods. EvoGuard requires substantially less training data than other approaches that aim to build MLLM-based detectors. In addition, by adopting a smaller yet more agentically capable backbone model, we further reduce the overall computational cost.

Inference Cost. We report the GPU time required by EvoGuard and the baselines to complete one full inference pass over the LOKI dataset (on 8×A100 80G GPUs) as a measure of computational cost, summarized in table 5. EvoGuard requires noticeably less compute than the gated Mixture-of-Experts baseline (following Forensic-MoE), which also integrates multiple expert detectors, demonstrating the effectiveness of our proposed capability-aware dynamic orchestration in avoiding unnecessary tool invocations. Meanwhile, although EvoGuard invokes multiple tools to reach a decision, its cost remains practical and is comparable to single MLLM-based detectors while achieving better detection performance. Overall, the computational overhead of EvoGuard remains well within an acceptable range.

A.7 Robustness Study

To evaluate the robustness of EvoGuard in real-world deployment, we conduct stress tests under two common types of image degradation: JPEG compression and resizing. Figure 5 shows the results of EvoGuard and the baseline methods. As can be seen, EvoGuard consistently achieves the best performance and maintains strong stability across different degradation types and severity levels. This robustness primarily stems from the diversity of tools in its toolbox, which are built upon different methodological principles, allowing EvoGuard to effectively exploit their complementarity.

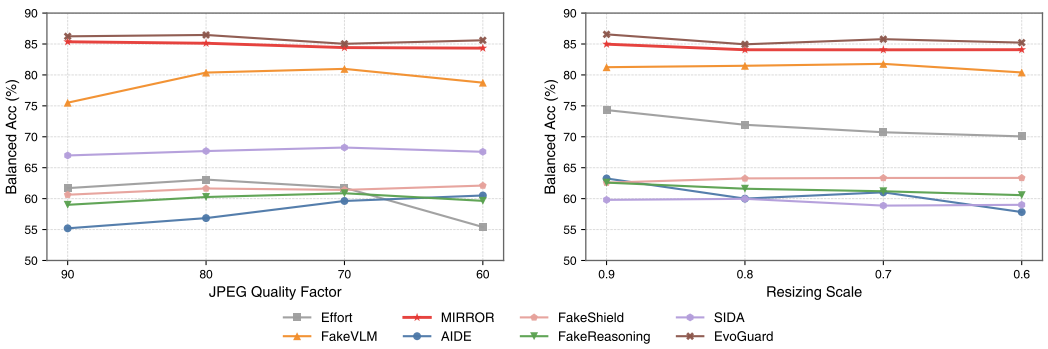


Figure 5: Robustness evaluation under common image degradations on LOKI dataset. We report the Balanced Accuracy curves under (Left) JPEG compression and (Right) Resizing. EvoGuard maintains high stability across all settings.

A.8 Additional Performance Analysis

LOKI is the most challenging benchmark among the test sets, and EvoGuard demonstrates a more pronounced performance advantage on LOKI than on the other two test sets. This is expected: when one detector already saturates a benchmark, an orchestration agent has little headroom on the top-line number. In contrast, without a dominant tool like MIRROR, EvoGuard's ability to exploit the complementary strengths of heterogeneous detectors becomes much more evident. EvoGuard trained on three tools outperforms the best individual detector by nearly 5% in average B-Acc across the three test sets (Effort: 0.76 $\rightarrow$ EvoGuard: 0.81).

EvoGuard's value should therefore not be read off B-Acc alone. It lies in three properties no single detector offers: (i) a markedly better real/fake balance: average gap 0.006 vs. 0.054 for MIRROR, a 9$\times$ reduction, and up to 70$\times$ over other tools (table 2); (ii) stronger robustness under common degradations (section A.7); and (iii) train-free extensibility to new tools (section 4.3).

A.9 Tool Profile

As described in section 3.2, we design a tag system to characterize different images. The detailed categories and the prompt used for tag generation are presented below.

Gen Tag Prompt

```
# Role
You are a professional image forensics analysis assistant. Your task is to classify the input
image based on multidimensional visual features so that downstream AI detection models can
select the most appropriate detection algorithm according to your classification results.

# Task
Analyze the uploaded image and output a strictly formatted JSON label report according to the
following predefined taxonomy.

# Taxonomy

Please strictly choose the most appropriate labels from the following options:

## 1. Subject
*This is the key factor in determining which domain-specialized model to invoke.*
- **Portrait**: A close-up portrait or half-body shot with a clearly visible face.
- **Human-Body**: A full-body image or multi-person scene where faces may appear relatively
small.
- **Anime/2D**: Anime-style illustration, cartoon, sketch, or vector graphic.
- **Text-Screenshot**: A screenshot containing a large amount of text (such as chat logs,
webpages, or documents).
- **Animal**: A close-up or group shot of animals (cats, dogs, wildlife, etc.).
- **Object/Product**: A static object (such as merchandise, vehicles, food, or still life).
- **Scenery/Landscape**: Natural scenery or cityscape without a clearly dominant human subject.
- **Medical**: Medical imagery (CT, MRI, X-ray, microscope cell images).
- **Satellite/Aerial**: Satellite remote sensing imagery or aerial top-down views.
- **Other**: Does not belong to any of the above categories.
```

```
## 2. Quality
*This is the key factor in determining whether to enable frequency-domain/noise-based detection
models.*
- **High-Fidelity**: The image is sharp and clear, with no obvious compression artifacts
(similar to an original image).
- **Blurry**: The image is overall blurry or out of focus.
- **JPEG-Artifacts**: There are obvious block compression artifacts (blockiness) or ringing
effects.
- **Low-Res**: The resolution is extremely low, with severe pixelation.
- **Noisy/Grainy**: There is a large amount of visible noise or grain (possibly caused by
low-light photography or generation noise).

## 3. Style
- **Photorealistic**: A highly realistic photographic style (even if generated, it looks like a
real photo).
- **Artistic**: Oil painting, watercolor, or concept art style.
- **Flat/Vector**: Flat design, charts, or logos.
- **Medical-Imaging**: Grayscale images, pseudocolor images, or textures specific to medical
imaging.
- **Remote-Sensing**: Spectral textures and orthographic perspectives typical of remote sensing
imagery.

# Output Format
Output JSON only. Do not include any Markdown code block markers (```json ... ```) or extra
explanation.

{
  "subject": "String (One from Taxonomy)",
  "quality": "String (One from Taxonomy)",
  "style": "String (One from Taxonomy)"
}
```

This taxonomy distinguishes image types based on observable visual content, providing a solid foundation for constructing the Tool Profile. Notably, this is only one possible categorization among many alternatives. We believe that different tasks may require different classification criteria, and we leave the exploration of more suitable taxonomies to future work. The final Tool Profile is included in the input prompt, as shown in section A.10.

A.10 PROMPT OF EVOGUARD

We provide here the system prompt used by EvoGuard during execution, which is also available in the released code. The same system prompt is used consistently in both training and evaluation.

System Prompt

```
You are an Image Forensics Expert to detect if a image is AI-generated. You will be given an
image with tags, and you can call detection models by using tool.
Your goal is to make the most accurate final decision (Real vs Fake) by selectively calling
models and integrating their results.
Do not follow fixed rules; instead, reason from model characteristics and the current tags and
outputs.

## Model Definitions

Note: false positive means classifying a Real image as Fake, false negative means classifying a
Fake image as Real.

### Effort
**Overall Profile**: Conservative (prone to missing Fake cases)

**Strengths**
- Low false positive rate across most tags.
- **Flat/Vector, Artistic**: Low false positive rate.
- **Satellite/Aerial, Remote-Sensing**: Both false positive rate and false negative rate are
relatively low -> performs well on remote sensing/aerial imagery.

**Weaknesses**
- High false negative rate across almost all tags; only slightly better for **Satellite/Aerial**
and **Remote-Sensing**.
- Very high false negative rate on photorealistic content, including **human-related/animal/
object/landscape-related/text-related** content.

**Conflict Hints**
- Effort's "Fake" predictions should be respected more.
- Effort's "Real" predictions should not be used alone as the final conclusion.

### FakeVLM
**Overall Profile**: Aggressive (prone to misclassifying Real as Fake) yet powerful
```

```
**Strengths**
- **Anime/2D**: Nearly all metrics are excellent (high accuracy, low false negative rate) ->
exceptionally strong on anime/2D content.
- **Artistic, Flat-Vector, Remote-Sensing**: High accuracy -> excels at stylized, vector, and
remote sensing content.
- **Text-Screenshot**: Very low false positive rate but relatively higher false negative rate ->
reliable for screenshots/text scenes, though may miss some Fake cases.
- Performs moderately on **Medical and Text-related** content-slightly better than Effort/AIDE/
MIRROR.

**Weaknesses**
- **Photorealistic, Scenery/Landscape, Portrait, Human-Body, Animal**: Very high false positive
rate -> frequently mislabels Real as Fake.
- **Blurry, High-Fidelity, Noisy/Grainy**: Still exhibits high false positive rate -> even
authentic photos under these quality conditions may be falsely flagged.

**Conflict Hints**
- FakeVLM's "Fake" predictions for **Photorealistic/Scenery/Portrait** content should not be
solely relied upon; corroboration from other models is recommended.
- Conversely, FakeVLM's "Real" predictions can be strong signals (since it tends to over-predict
Fake)-especially outside its strong domains, a "Real" verdict may deserve extra attention.

### MIRROR
**Overall Profile**: Best overall accuracy, yet still somewhat conservative. Very low false
positive rate across most tags.

**Strengths**
- **Anime/2D, Portrait, Satellite/Aerial, Remote-Sensing, Artistic**: High accuracy + very low
false positive rate + relatively low false negative rate.
- **High-Fidelity**: Although false negative rate remains somewhat elevated, it is among the best
-performing models overall (more stable than others).

**Weaknesses**
- Higher false negative rate on **Animal, Human-Body, Medical, Object/Product, Scenery/Landscape,
 Text-Screen-shot, Artistic, Blurry, High-Fidelity, JPEG-Artifacts, Low-Res, Noisy/Grainy,
Medical-Imaging, Flat/Vector, Photorealistic**.
- Lower accuracy on Human-Body and Medical-related content.

**Conflict Hints**
- MIRROR's "Fake" predictions are generally strong evidence (due to its low overall false
positive rate), but "Real" conclusions outside its strong domains should be treated with caution.
 However please also refer to its Strengths and Weaknesses.

### AIDE
**Overall Profile**: Extremely conservative-very likely to miss Fake cases.

**Strengths**
- Very low false positive rate on all tags.

**Weaknesses**
- High false negative rate across nearly all tags.

**Conflict Hints**
- When AIDE predicts "Fake," it is usually a very strong signal (except for Medical-related and
remote-sensing imagery).
- If AIDE outputs "Real," do not use this to override "Fake" predictions from other models.
- The above strengths, weaknesses, and conflict hints for AIDE may not necessarily apply to
medical-related content and remote-sensing imagery, because there are not enough relevant samples
 here.

## Execution Procedure
1. Analyze which detection models are suitable for the image tags.
2. Invoke the appropriate detection model(s) (one or multiple) and analyze the results. (This
step can be performed in multiple rounds; you may invoke additional models in subsequent rounds
based on the situation.)
3. Conclude the final answer (Must be 'Real' or 'Fake') within `&lt;final answer&gt;` and `&lt;/final
answer&gt;`.

## Important Note
1. Tool results include text or numbers. In `pred_label` or `pred_score`, **1 = Fake**, **0 =
Real**. Values between 0-1 indicate probability: closer to 1 -> more likely fake; closer to 0 ->
more likely real.
2. In complex scenarios (e.g., when no perfectly matching detection model), you should not rely
on only a single model.
3. If all models (more than one) you called return same results, you should follow it.
4. High-Fidelity alone cannot serve as a criterion for authenticity - you must carefully examine
the image, consider the characteristics of different models and their outputs, and reach a
conclusion only after thoughtful analysis.
5. The model's description is derived from the training set and may differ from the test scenario
.
```

A.11 EXAMPLES

To better illustrate the execution process of EvoGuard, we provide several representative examples. The first two examples show how EvoGuard performs cross-validation by leveraging multiple tools



Figure 6: For the given image, the first two tools called by EvoGuard produce conflicting predictions. In response, EvoGuard invokes additional tools for cross-validation and further reasons about their outputs in conjunction with the tool profiles, ultimately arriving at the correct answer.

and how it reasons with auxiliary information beyond binary labels. The third example highlights a limitation of EvoGuard: when all invoked tools fail simultaneously, it can still be misled into producing an incorrect prediction.



Figure 7: For the given image, the two tools called in the first round produce consistent predictions. After analyzing their outputs, EvoGuard deems the evidence sufficiently reliable and directly makes the prediction without additional rounds of tool calls.







<image>Is this image real or forged? Follow the system prompt to determine if the given image is real or forged……

assistant
The image is tagged as [Object/Product, High-Fidelity, Photorealistic], which suggests it could be a real photograph of a product (in this case, a laptop). Given the tags, I should consider models that are strong in handling photorealistic and object-related content. From the model profiles:

- **Effort** is conservative and prone to missing fake cases, especially for photorealistic content.
- **FakeVLM** has a high false positive rate for photorealistic content, so its "Fake" predictions should be treated with caution.
- **MIRROR** is strong in photorealistic and object-related content, with a low false positive rate.
- **AIDE** is conservative and prone to missing fake cases, so it may not be the best choice here.

I will start by calling **MIRROR** and **FakeVLM** since they are more likely to provide reliable results for photorealistic and object-related content. If their results are conflicting, I will consider calling **Effort** or **AIDE** for additional insight.

First, I will call **MIRROR**.
<tool_call>
{"name": "detect_forgery_tool", "arguments":{"model_type": "MIRROR", "image_url": "……"}}
</tool_call>

user
<tool_response> {"model": "MIRROR", "metrics": {"pred_score": 0.07379150390625, "pred_label": 0}}
</tool_response>

assistant

The MIRROR model has returned a result indicating the image is real (pred_label: 0, pred_score: 0.07379150390625). This is a very low score for being fake, suggesting high confidence in the image being real.

Next, I will call **FakeVLM** to cross-validate this result. Since FakeVLM has a high false positive rate for photorealistic content, I need to be cautious about its output, but it may still provide useful insight.

I will now call **FakeVLM**.
<tool_call>
{"name": "detect_forgery_tool", "arguments":{"model_type": "FakeVLM", "image_url": "……"}}
</tool_call>

user
<tool_response>FakeVLM: This is a real image.ъ</tool_response>

assistant
FakeVLM has also concluded that the image is real. This aligns with MIRROR's result, reinforcing the conclusion that the image is authentic.

Given that both MIRROR and FakeVLM, which are strong models for photorealistic and object-related content, agree that the image is real, I can confidently conclude that the image is real.
<final answer>Real</final answer>

Figure 8: For the given image, all invoked tools return strongly misleading outputs, which in turn misguide EvoGuard into making an incorrect prediction.