



# ForgerySleuth: Empowering Multimodal Large Language Models for Image Manipulation Detection

Zhihao Sun<sup>1,2</sup>, Haoran Jiang<sup>1,2</sup>, Haoran Chen<sup>1,2</sup>, Yixin Cao<sup>1,2</sup>, Xipeng Qiu<sup>1,2</sup>,  
Zuxuan Wu<sup>1,2†</sup>, Yu-Gang Jiang<sup>1,2</sup>

<sup>1</sup>Shanghai Key Lab of Intell. Info. Processing, School of CS, Fudan University

<sup>2</sup>Shanghai Collaborative Innovation Center of Intelligent Visual Computing

<https://github.com/sunzhihao18/ForgerySleuth>

## Abstract

Multimodal large language models have unlocked new possibilities for various multimodal tasks. However, their potential in image manipulation detection remains unexplored. When directly applied to the IMD task, M-LLMs often produce reasoning texts that suffer from hallucinations and overthinking. To address this, we propose ForgerySleuth, which leverages M-LLMs to perform comprehensive clue fusion and generate segmentation outputs indicating specific regions that are tampered with. Moreover, we construct the ForgeryAnalysis dataset through the Chain-of-Clues prompt, which includes analysis and reasoning text to upgrade the image manipulation detection task. A data engine is also introduced to build a larger-scale dataset for the pre-training phase. Our extensive experiments demonstrate the effectiveness of ForgeryAnalysis and show that ForgerySleuth significantly outperforms existing methods in generalization, robustness, and explainability.

## 1 Introduction

Recent advancements in multimodal large language models (M-LLMs) [1] have unlocked new possibilities across a variety of tasks, propelling advancements in numerous traditional research fields [2, 3, 4]. Despite these advancements, few attempts have been made to enhance the ability of multimodal large language models on the image manipulation detection task (IMD). In fact, with the rise of advanced image generation and editing tools, both creative and malicious alterations to visual content have become increasingly difficult to detect. As a result, identifying manipulated images has become essential for maintaining the integrity of digital media [5, 6, 7].

While appealing, directly applying M-LLMs for IMD produces results in text, which falls short even compared with traditional IMD methods that are able to generate segmentation masks to highlight tampered regions. In addition, M-LLMs are often prone to hallucinations and overthinking, making them unreliable for precise manipulation detection. For instance, as shown in Figure 1 (a), M-LLMs may incorrectly identify non-tampered regions, such as the “fish” area, as altered. This highlights the need to enhance the reasoning capabilities of M-LLMs for tampered region detection. Furthermore, M-LLMs, pre-trained on vast datasets, excel in recognizing authentic image patterns and world knowledge, yet lack the ability to combine low-level statistical features to pinpoint manipulation evidence, which has proven crucial for IMD [8, 9, 10, 11].

In this paper, we explore how to unleash the power of M-LLMs for IMD tasks. We introduce carefully designed modules that empower an M-LLM as an IMD expert, named ForgerySleuth.

†: corresponding author.

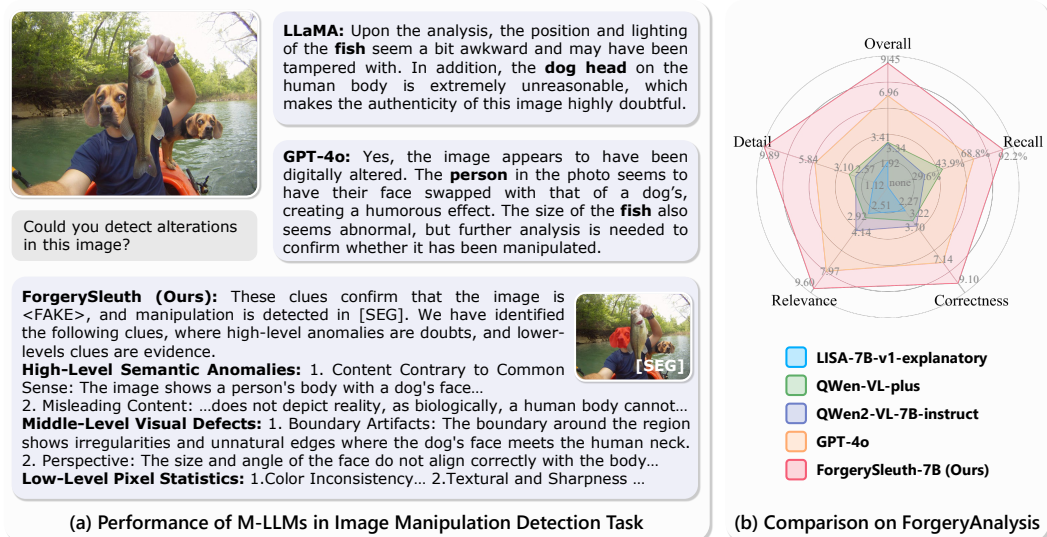


Figure 1: Performance and comparison of existing M-LLMs on the image manipulation detection task. Our ForgerySleuth assistant provides explanatory analysis with Chain-of-Clues and demonstrates the best forgery analysis capabilities.

is expected to provide a textual explanation of detected clues with the reasoning process, along with a segmentation mask to highlight tampered regions. Inspired by the fact that existing IMD methods rely on low-level features, such as noise patterns, to localize precise tampered regions, we aim to capture similar low-level features through ForgerySleuth. To do so, ForgerySleuth integrates M-LLMs with a trace encoder, enabling the model to leverage world knowledge to detect high-level semantic anomalies while still capturing low-level forgery traces. Additionally, inspired by LISA [12], we introduce a vision decoder with a fusion mechanism that uses attention to combine high-level anomalies in the LLM tokens and low-level traces in the trace embedding, ensuring the generation of accurate segmentation masks.

To further enhance the reasoning capability of M-LLMs on manipulation detection, we propose a supervised fine-tuning (SFT) dataset ForgeryAnalysis specifically tailored for the IMD task. Each entry in ForgeryAnalysis is initially generated by GPT-4o using a novel Chain-of-Clues prompt. Specifically, we ask GPT-4o to produce a detailed thought process and provide reasoning for why a particular region is tampered with, including high-level semantic anomalies (e.g., content that contradicts common sense), mid-level visual defects (such as lighting inconsistencies), and low-level pixel information (such as color, texture, etc.). The generated entries are then reviewed and refined by experts. Additionally, we build a data engine based on this dataset to automate forgery analysis, enabling us to create a larger-scale ForgeryAnalysis-PT dataset for pre-training.

Extensive experiments on popular benchmarks demonstrate the success of ForgerySleuth in pixel-level manipulation localization and text-based forgery analysis. Specifically, our approach outperforms the current SoTA method by up to 24.7% in pixel-level localization tasks. Moreover, in the ForgeryAnalysis-Eval comprehensive scoring, our method surpasses the best available model, GPT-4o, achieving an improvement of 35.8%. In summary, our main contributions include:

- **Novel Exploration.** We explored the role of M-LLMs in image manipulation detection, upgrading the manipulation detection task by incorporating clues analysis and reasoning.
- **Valuable Dataset.** We constructed ForgeryAnalysis dataset, providing instructions for analysis and reasoning through Chain-of-Clues prompting. Additionally, we developed a data engine to automate forgery analysis, enabling the creation of a large-scale dataset.
- **Practical Framework.** We introduced ForgerySleuth assistant framework, which integrates M-LLMs with a trace encoder to leverage multi-level clues. The vision decoder with a fusion mechanism enables comprehensive clues fusion and segmentation outputs.

## 2 Related Work

### 2.1 LLMs and Multimodal LLMs

The success of large language models in various natural language processing tasks has led researchers to explore their integration with vision modalities, resulting in the development of M-LLMs. BLIP-2 [13] introduces a visual encoder to process image features. LLaVA [14] aligns image and text features to achieve comprehensive visual and language understanding. Researchers also utilize prompt engineering to connect independent vision and language modules via API calls without end-to-end training [15, 16, 17]. However, while these approaches enable M-LLMs to perceive, the intersection with vision-centric tasks, such as segmentation, remains underexplored. Additionally, VisionLLM [18] and LISA [12] effectively integrate segmentation capabilities into M-LLMs, making them support vision-centric tasks, such as segmentation.

With advancements in fundamental reasoning and multimodal information processing, M-LLMs have demonstrated impressive proficiency across a diverse range of tasks, including image captioning and video understanding [1]. Moreover, M-LLMs have been developed to address more complex real-world tasks in robotics, such as embodied agents [19, 20] and autonomous driving [4]. However, integrating M-LLMs into the field of image manipulation detection remains unexplored. While M-LLMs possess valuable world knowledge and can potentially detect high-level anomalies, M-LLMs are often prone to hallucinations and overthinking, making them unreliable for precise manipulation detection. Furthermore, there is no existing IMD dataset with analysis instructions for supervised fine-tuning, which further restricts their capabilities.

### 2.2 Image Manipulation Detection

Image manipulation detection is a critical task in digital image forensics. The task has evolved beyond merely determining whether an image is authentic [21, 22]; it involves localizing tampered regions and providing segmentation masks [23, 24], which leads to more intuitive results. Early attempts [25, 26, 27, 28, 29] identified anomalies and designed corresponding hand-engineered features. These efforts systematically use various tampering clues, laying a solid foundation for the field. However, such hand-engineered features are specific to certain tampering types, which limits their applicability in real-world scenarios.

Recent approaches have shifted to a more general capability of identifying complex and unknown manipulations. Semantic-agnostic features, less dependent on specific content, are thought to provide better generalization [10, 11]. Common strategies include incorporating filters or extractors to capture low-level noise features [8, 9, 30, 31] and high-frequency features [32], and using content features extracted from the image view as a supplement to detect manipulation traces [10, 30, 31]. However, many of these features are learned implicitly by the network, which limits their explainability. Studies also [11, 32] detect anomalies by comparing patch-level or object-level features. However, capturing high-level semantic anomalies, such as content that conflicts with common sense or physical laws, is still challenging. In this work, we extend the task by presenting a reasoning process with multiple levels of clues expressed in natural language, making the detection results more comprehensible. Our proposed ForgerySleuth framework leverages M-LLMs to address this challenge, effectively leveraging world knowledge to detect high-level semantic anomalies while still capturing low-level forgery traces using a trace encoder. FakeShield [33] is a concurrent work that proposes a multimodal large model for image manipulation detection.

## 3 ForgeryAnalysis Dataset

Our goal is to leverage existing M-LLMs to construct a high-quality dataset for IMD. We first describe how M-LLMs are utilized to generate initial clue analyses from various types of manipulated image sources in Section 3.1. These analyses are then meticulously refined by experts to create a high-quality dataset of 2,370 samples, which are used for the supervised fine-tuning (SFT) phase and the evaluation of M-LLMs. Section 3.2 describes our proposed data engine, which expands the 2k high-quality analysis instructions from Section 3.1 to 50k. This expansion supports the pre-training phase while maintaining quality standards. Detailed statistics of our dataset, along with examples of tampering analysis instructions, are provided in Section A.

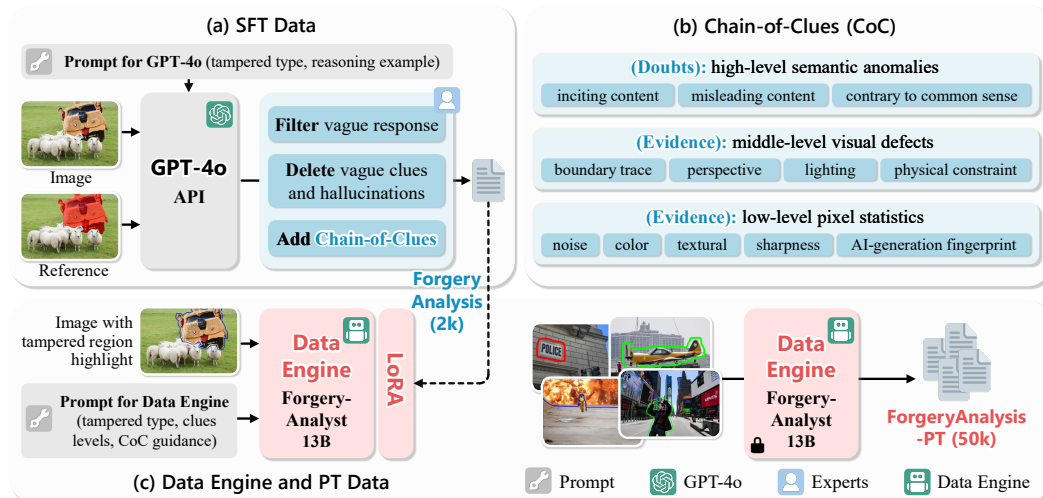


Figure 2: **ForgeryAnalysis Dataset Construction Pipeline.** Our pipeline begins with (a) GPT-4o generating initial analyses for manipulated images with annotated regions, followed by human expert review. The refined analyses are organized into (b) the Chain-of-Clues format. This human-curated ForgeryAnalysis (2k) dataset is used to train a data engine. Finally, (c) this data engine generates ForgeryAnalysis-PT, a larger-scale dataset for model pre-training.

### 3.1 ForgeryAnalysis Data

Due to the absence of instruction-following datasets that support clue analysis, we introduce the ForgeryAnalysis dataset to support the supervised fine-tuning (SFT) and evaluation of M-LLMs. As illustrated in Figure 2 (a), for each image selected from various data sources, we utilize the advanced M-LLM, GPT-4o, to generate initial clue analyses. Experts then carefully revise these analyses to eliminate hallucinations and ensure high quality. Furthermore, we have integrated a Chain-of-Clues structure to enhance reasoning capabilities.

**Data Source.** To ensure a diverse dataset that covers various types of manipulation, we collect 4,000 tampered images from existing IMD datasets, including MIML [34], CASIA2 [35], DEFACTO [36], and AutoSplice [37]. Each image is accompanied by annotations indicating the tampered regions. These sources include common real-world manipulation techniques such as splicing, copy-move, object removal, AI generation, and Photoshop edits.

**Instruction Design.** We aim to provide highly specific and content-driven descriptions of clues for each image, rather than vague or general analysis. We refer to the clues and evidence widely used in digital image forensics and manipulation detection [38, 39] to design a more reliable and effective instruction structure. The key clues include but are not limited to 1) low-level pixel statistics (*e.g.*, noise, color, texture, sharpness, and AI-generation fingerprints), 2) middle-level visual defects (*e.g.*, traces of tampered boundaries, lighting inconsistencies, and perspective relationships), and 3) high-level semantic anomalies (*e.g.*, content that contradicts common sense, incites, or misleads).

The prompt instructs GPT-4o to assume it has detected manipulation in the highlighted region of the reference image and to analyze its detection based on clues from various levels and aspects. We provide the tampering type to help GPT-4o focus on relevant clues, along with a reasoning example to guide the content and format of the output. We also instruct GPT-4o to incorporate corresponding world knowledge, such as well-known individuals or landmarks.

**Chain-of-Clues.** The responses generated by GPT-4o are subsequently revised by experts to ensure quality. Responses lacking detailed content analysis are filtered out, retaining only high-quality responses as draft data. The experts carefully review vague and incorrect statements that may arise from hallucinations, removing irrelevant clues and evidence. Inspired by works such as [40, 41] that introduce Chain-of-Thought (CoT) prompting and demonstrate its effectiveness in enhancing the step-by-step reasoning capabilities of LLMs, we propose a Chain-of-Clues (CoC) prompting approach, illustrated in Figure 2 (b). The reasoning process begins with “unveiling doubts” using

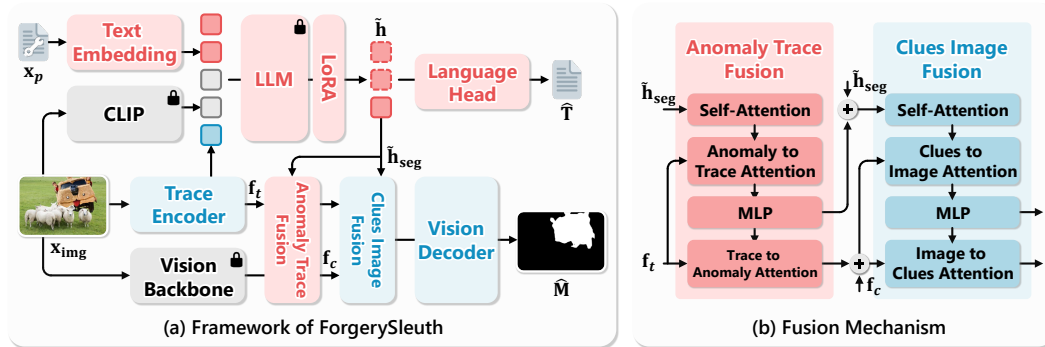


Figure 3: **Framework of ForgerySleuth.** Given an image  $x_{img}$  and a prompt query  $x_p$ , the M-LLM  $\mathcal{F}_m$  detects high-level semantic anomalies and generates a textual output  $\hat{T}$ . The trace encoder  $\mathcal{F}_t$  captures low-level, semantic-agnostic features. The vision decoder  $\mathcal{D}$  fuses vision embeddings with the prompt embedding corresponding to [SEG] token to generate the segmentation mask  $\hat{M}$ . LoRA is utilized in trainable modules for fine-tuning.

high-level clues, followed by “pinpointing evidence” through middle-level and low-level features. We organize the clues according to this structure, creating a coherent chain of clues, which results in 2,370 manually revised samples. To ensure accuracy, we conduct additional cross-validation with more than two experts, selecting 618 samples for evaluation. The remaining data are used for supervised fine-tuning.

### 3.2 ForgeryAnalyst Engine

The pre-training phase requires a larger-scale dataset compared to the supervised fine-tuning phase. However, the high cost of GPT-4o and the need for expert revisions make expanding the dataset a challenge. To overcome this limitation and create a pre-training dataset, we introduce an additional data engine [42, 43, 44, 45] that scales up the training set. Specifically, we fine-tune LLaVA-v1.5-13B using LoRA [46] on the ForgeryAnalysis-SFT dataset, aiming to replicate the ability of GPT-4o to analyze clues while having experts eliminate any hallucinations. The resulting model is used as our data engine that is able to annotate data automatically. In particular, we select 50k images from existing public datasets. The data engine receives input that includes explicit information about the tampered region, aiming to generate more precise and comprehensive clue analyses. It outputs manipulation analyses organized in the CoC format, as illustrated in Figure 2 (c). For more detailed information on the data engine, including the specific prompts used, please refer to Section A.2, and Figure 17 for output examples. The entire analysis generation process takes approximately 16 A800 GPU days. We refer to this dataset as ForgeryAnalysis-PT.

## 4 ForgerySleuth

Given an input image for detection and a prompt specifying the detection request, our goal is to output a binary segmentation mask  $\hat{M}$  of the tampered regions, as well as text  $\hat{T}$  that includes reasoning and evidence. This poses a challenge as the model needs to bridge vision and language modalities and capture tampering features across multiple levels. To address this, ForgerySleuth uses M-LLMs to connect vision and language modalities, enabling the description of detected clues and evidence in textual form. Additionally, we incorporate a vision decoder into the multimodal large language models to perform dense prediction, generating the tampered region mask for this specific task. The pipeline of our framework is illustrated in Figure 3 (a).

More formally, we extend the original LLM vocabulary with new tokens, including [SEG], which requests segmentation output, and <REAL> and <FAKE>, which indicate the image classification results. Given the input image  $x_{img}$  to be detected along with the prompt  $x_p$ , we first feed them into the M-LLM  $\mathcal{F}_m$ , which outputs the hidden embedding  $\tilde{h}$  at the last layer of LLM. We then extract the embedding  $\tilde{h}_{seg}$  corresponding to the [SEG] token from the hidden embedding  $\tilde{h}$  and apply an

MLP projection layer  $\gamma$  to obtain  $\mathbf{h}_{\text{seg}}$ . The process can be formulated as

$$\tilde{\mathbf{h}} = \mathcal{F}'_m(\mathbf{x}_{\text{img}}, \mathbf{x}_p), \quad (1)$$

$$\mathbf{h}_{\text{seg}} = \gamma(\tilde{\mathbf{h}}_{\text{seg}}). \quad (2)$$

$\mathcal{F}'_m$  denotes M-LLM without the language head. This embedding represents the suspicious anomalous regions the M-LLM detects and serves as the input to guide the subsequent vision decoder.

Inspired by existing IMD methods, we propose an independent trace encoder  $\mathcal{F}_t$  to focus on low-level features, which complements the relatively high-level vision and semantic features discovered by the M-LLM and provides more reliable tampering evidence, helping to minimize hallucinations from the LLM. Specifically, constrained convolutions [47] are employed with residual connections [48] at the front part of the encoder to suppress the image content and learn manipulation features adaptively. The input image  $\mathbf{x}_{\text{img}}$  is also fed into encoder  $\mathcal{F}_t$ , producing dense manipulation trace features  $\mathbf{f}_t$ , which can be expressed by

$$\mathbf{f}_t = \mathcal{F}_t(\mathbf{x}_{\text{img}}). \quad (3)$$

The vision backbone  $\mathcal{F}_v$  simultaneously extracts dense vision content feature  $\mathbf{f}_c$  to support more precise segmentation. Finally,  $\mathbf{f}_c$ ,  $\mathbf{f}_t$  and  $\mathbf{h}_{\text{seg}}$  are fed to the vision decoder  $\mathcal{D}$  to generate the segmentation mask  $\hat{\mathbf{M}}$  which indicates the tampered region. The language modeling head  $\mathcal{H}$  processes  $\tilde{\mathbf{h}}$  and outputs  $\hat{\mathbf{y}}_{\text{txt}}$ , containing an analysis of the reasoning and evidence. It can be formulated as

$$\mathbf{f}_c = \mathcal{F}_v(\mathbf{x}_{\text{img}}), \quad (4)$$

$$\hat{\mathbf{M}} = \mathcal{D}(\mathbf{f}_c, \mathbf{f}_t, \mathbf{h}_{\text{seg}}), \quad \hat{\mathbf{T}} = \mathcal{H}(\tilde{\mathbf{h}}). \quad (5)$$

**Fusion Mechanism.** To integrate the image content embedding  $\mathbf{f}_c$ , trace embedding  $\mathbf{f}_t$ , and the LLM output tokens  $\mathbf{h}_{\text{seg}}$  obtained from the M-LLM, we take inspiration from Transformer segmentation models [49, 50, 42] and design a vision decoder with a fusion attention mechanism, as illustrated in Figure 3 (b). Here, the first layer of the module computes attention between anomalies in the LLM output tokens and traces in the trace embeddings, facilitating the organization and pinpointing of clues. The subsequent layers focus on attention between refined clues in the upgraded tokens and content in the image embeddings, enabling more precise segmentation of the tampered regions.

**Learning Objective.** Our framework is trained end-to-end using both the reasoning text loss  $\mathcal{L}_{\text{txt}}$  and the tampered region mask loss  $\mathcal{L}_{\text{mask}}$ . The final learning objective  $\mathcal{L}$  is formulated as a weighted sum of these losses, with weight  $\lambda_{\text{txt}}$  and  $\lambda_{\text{mask}}$ , as

$$\mathcal{L} = \lambda_{\text{txt}}\mathcal{L}_{\text{txt}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}}. \quad (6)$$

Specifically,  $\mathcal{L}_{\text{txt}}$  is the auto-regressive cross-entropy (CE) loss for reasoning text generation, guiding the model to collect multi-level clues, while  $\mathcal{L}_{\text{mask}}$  is the mask loss, which encourages the model to produce precise segmentation results.  $\mathcal{L}_{\text{mask}}$  is implemented by the combination of binary cross-entropy (BCE) loss and DICE loss, with respective weights  $\lambda_{\text{bce}}$  and  $\lambda_{\text{dice}}$ . Given the ground-truth targets  $\mathbf{T}$  and  $\mathbf{M}$ , the final loss functions are defined as

$$\mathcal{L}_{\text{txt}} = \text{CE}(\hat{\mathbf{T}}, \mathbf{T}), \quad (7)$$

$$\mathcal{L}_{\text{mask}} = \lambda_{\text{bce}}\text{BCE}(\hat{\mathbf{M}}, \mathbf{M}) + \lambda_{\text{dice}}\text{DICE}(\hat{\mathbf{M}}, \mathbf{M}). \quad (8)$$

It is noteworthy that no separate loss is required for the classification result, as the supervision for <REAL> and <FAKE> tokens are integrated into the text loss  $\mathcal{L}_{\text{txt}}$ .

#### 4.1 Training Strategy

**Trainable Parameter.** To preserve the world knowledge and the normal patterns of authentic images learned by the pre-trained M-LLM  $\mathcal{F}_m$ , we adopt LoRA [46] for efficient fine-tuning. The vision backbone  $\mathcal{F}_v$  is entirely frozen to retain the capacity for modeling image content features, which are crucial for accurate segmentation. Both the trace encoder  $\mathcal{F}_t$  and the vision decoder  $\mathcal{D}$  are fully trainable and fine-tuned. Additionally, the token embeddings of the LLM, the language modeling head  $\mathcal{H}$ , and the projection layer  $\gamma$  are updated during training. Despite the large model scale, only 5.47% of the parameters are trainable, making end-to-end training more efficient.

Table 1: Manipulation localization results comparing ForgerySleuth with SoTA methods.

| Method               | Optimal Threshold F1 |              |              |              |              | Fixed Threshold F1 (0.5) |              |              |              |              |
|----------------------|----------------------|--------------|--------------|--------------|--------------|--------------------------|--------------|--------------|--------------|--------------|
|                      | Columbia             | Coverage     | CASIA1       | NIST16       | COCOglide    | Columbia                 | Coverage     | CASIA1       | NIST16       | COCOglide    |
| Mantra-Net [8]       | 0.650                | 0.486        | 0.320        | 0.225        | 0.673        | 0.508                    | 0.317        | 0.180        | 0.172        | 0.516        |
| SPAN [9]             | 0.873                | 0.428        | 0.169        | 0.363        | 0.350        | 0.759                    | 0.235        | 0.112        | 0.228        | 0.298        |
| MVSS-Net [10]        | 0.781                | 0.659        | 0.650        | 0.372        | 0.642        | 0.729                    | 0.514        | 0.528        | 0.320        | 0.486        |
| PSCC-Net [53]        | 0.760                | 0.615        | 0.670        | 0.210        | 0.685        | 0.604                    | 0.473        | 0.520        | 0.113        | 0.515        |
| CAT-Net2 [54]        | 0.923                | 0.582        | 0.852        | 0.417        | 0.603        | 0.859                    | 0.381        | 0.752        | 0.308        | 0.434        |
| TruFor [30]          | 0.914                | 0.735        | 0.822        | 0.470        | 0.720        | 0.859                    | 0.600        | 0.737        | 0.399        | 0.523        |
| UnionFor. [31]       | 0.925                | 0.720        | 0.863        | 0.489        | 0.742        | 0.861                    | 0.592        | 0.760        | 0.413        | 0.536        |
| <b>ForgerySleuth</b> | <b>0.931</b>         | <b>0.792</b> | <b>0.870</b> | <b>0.610</b> | <b>0.751</b> | <b>0.925</b>             | <b>0.684</b> | <b>0.804</b> | <b>0.518</b> | <b>0.562</b> |

**Data Formulation.** The training process involves two phases: **1) Pre-training Phase:** (a) In the first stage, we aim to align the framework modules and ensure that the model can perform basic segmentation and visual reasoning tasks. For foundational segmentation abilities, we use semantic segmentation datasets such as ADE20k [51] and COCO-Stuff [52], transforming these datasets into visual question-answer pairs using class names as questions. Additionally, we incorporate ReasonSeg [12] to strengthen visual reasoning capabilities. (b) The ForgeryAnalysis-PT dataset focuses on the IMD task, including analysis instructions that enable the model to recognize tampering traces and identify clues. We also utilize public IMD datasets, including MIML [34] and CASIA2 [35], with prompts randomly selected from simple or vague responses. **2) Supervised Fine-tuning Phase:** The ForgeryAnalysis-SFT dataset, meticulously revised by experts to ensure the accuracy of reasoning and analysis, is used for final supervised fine-tuning to standardize the analysis and output.

## 5 Experiment

### 5.1 Experimental Setting

**Testing Dataset.** We utilize six publicly accessible test datasets, which are Columbia [55], Coverage [56], CASIA1 [35], NIST16 [57], IMD20 [58], and COCOglide [30], to evaluate and compare our method with state-of-the-art methods thoroughly. To effectively evaluate the model’s generalization capability, these datasets are ensured to be disjoint from the training data. Additionally, we use our ForgeryAnalysis-Eval dataset to assess the reasoning and analysis capabilities of the methods.

**Evaluation Metrics.** Localizing the tampered regions at the pixel level is crucial in image manipulation detection. We follow established practices [31] by using optimal threshold and fixed threshold F1 scores and the threshold-independent Area Under the Curve (AUC) metric. To ensure fairness and precision in our comparative analysis, we refer to some results for other methods from [31, 32].

Evaluating the novel reasoning analysis outputs presents a unique challenge, as it involves assessing the comprehension, reasoning, and correctness in generating text explanations. To measure the similarity between the generated text and the ground-truth text, thereby reflecting its accuracy, we incorporate Semantic Textual Similarity (STS) metric. We follow SBERT [59] and use STS to measure the similarity in our ForgeryAnalysis-Eval, where various models are utilized to calculate the similarity. Additionally, inspired by previous work [60, 14], we use GPT-4 as an automated evaluator to assess the reasoning performance of different models, which enables a more holistic evaluation of the generated analysis text, Figure 18 and Section D.3 provide the prompt format and the evaluation criteria. We conduct evaluations twice and report the average performance.

### 5.2 Manipulation Detection Results

The results in Table 1 and Table 2 demonstrate the performance of our ForgerySleuth and comparisons with SoTA methods for image manipulation localization, using pixel-level F1 scores and AUC metrics, respectively. Our method consistently achieves the highest or second-highest AUC. Regarding F1 scores, our approach surpasses other methods across all datasets, showcasing its reliability under both fixed and optimal thresholds. On challenging datasets like NIST16 and IMD20, ForgerySleuth outperforms UnionFormer by margins of 0.105 and 0.121 for fixed and optimal thresholds, respectively, which we believe is significant given its challenging nature. Furthermore, on the COCOglide dataset, which features novel diffusion-based manipulations, our model also exceeds the UnionFormer. These significant improvements can be attributed to the capability to effectively capture both low-level

Table 2: Manipulation localization results of ForgerySleuth and SoTA methods, using pixel-level AUC as the evaluation metric.

| Method               | Columbia     | Coverage     | CASIA1       | NIST16       | IMD20        |
|----------------------|--------------|--------------|--------------|--------------|--------------|
| Mantra-Net [8]       | 0.824        | 0.819        | 0.817        | 0.795        | 0.748        |
| SPAN [9]             | 0.936        | 0.922        | 0.797        | 0.840        | 0.750        |
| PSCC-Net [53]        | 0.982        | 0.847        | 0.829        | 0.855        | 0.806        |
| ObjectFormer [32]    | 0.955        | 0.928        | 0.843        | 0.872        | 0.821        |
| TruFor [30]          | 0.947        | 0.925        | 0.957        | 0.877        | -            |
| UnionFormer [31]     | 0.989        | 0.945        | <b>0.972</b> | 0.881        | 0.860        |
| <b>ForgerySleuth</b> | <b>0.992</b> | <b>0.962</b> | 0.969        | <b>0.898</b> | <b>0.911</b> |

Table 3: Robust evaluation results of ForgerySleuth and existing methods using pixel-level AUC.

| Distortion              | SPAN   | PSCC-Net | ObjectFor. | UnionFor. | Ours          |
|-------------------------|--------|----------|------------|-----------|---------------|
| <i>w/o</i> distortion   | 0.8395 | 0.8547   | 0.8718     | 0.8813    | <b>0.8982</b> |
| Resize (0.78 $\times$ ) | 0.8324 | 0.8529   | 0.8717     | 0.8726    | <b>0.8962</b> |
| Resize (0.25 $\times$ ) | 0.8032 | 0.8501   | 0.8633     | 0.8719    | <b>0.8792</b> |
| GSBr ( $k = 3$ )        | 0.8310 | 0.8538   | 0.8597     | 0.8651    | <b>0.8863</b> |
| GSBr ( $k = 15$ )       | 0.7915 | 0.7993   | 0.8026     | 0.8430    | <b>0.8658</b> |
| GSN ( $\sigma = 3$ )    | 0.7517 | 0.7842   | 0.7958     | 0.8285    | <b>0.8452</b> |
| GSN ( $\sigma = 15$ )   | 0.6728 | 0.7665   | 0.7815     | 0.8057    | <b>0.8139</b> |
| JPEG ( $q = 100$ )      | 0.8359 | 0.8540   | 0.8637     | 0.8802    | <b>0.8974</b> |
| JPEG ( $q = 50$ )       | 0.8068 | 0.8537   | 0.8624     | 0.8797    | <b>0.8839</b> |

trace features and high-level semantic inconsistencies, enabling it to detect even subtle generative manipulations. Overall, the results emphasize the generalization ability of ForgerySleuth.

**Robustness Evaluation.** We apply several image distortions to the NIST16 dataset following [31, 32] to evaluate the robustness of our method and compare the results with other methods. The distortions included 1) resizing images to different scales, 2) applying Gaussian blur with different kernel sizes  $k$ , 3) adding Gaussian noise with various standard deviation  $\sigma$ , and 4) compressing images using JPEG with different quality factors  $q$ . The results in Table 3 show that our model consistently outperforms other methods across all types of distortions. This improvement in robustness stems from the ability to identify high-level semantic anomalies rather than relying solely on low-level statistical features that are more susceptible to distortions.

### 5.3 Forgery Analysis Results

We compare our forgery analysis results with several M-LLMs, including GPT-4o, Qwen2-VL [61], and LISA [12]. Additionally, we perform LoRA fine-tuning on the LISA using the ForgeryAnalysis-SFT to provide a more comprehensive comparison. We incorporate STS to measure the similarity between the generated text and the ground-truth text in ForgeryAnalysis-Eval. The results, shown in Table 4, confirm that our ForgerySleuth consistently outperforms other methods.

We also leverage GPT-4 as an evaluator based on the ForgeryAnalysis-Eval dataset to assess the quality of text analysis and reasoning. We collect answers from each M-LLM, and GPT-4 assigns a score from 1 to 10 for each response. Beyond scoring, GPT-4 provides detailed explanations for its ratings. We also report the recall rate to directly reflect the ability to identify tampered images.

Figure 1 (b) presents the scores of different models without additional fine-tuning across various evaluation dimensions. The existing models exhibit low recall rates, and the overall evaluation suggests that they struggle to identify manipulations and provide accurate analyses. Figure 5 shows the scores of LISA and ForgerySleuth after SFT, along with the versions without SFT. ForgerySleuth shows an improvement of 5.05 in the overall score compared to LISA, further demonstrating the effectiveness of our design targeted specifically for the IMD task. Furthermore, using the SFT dataset results in performance gains for both methods, indicating the quality of the ForgeryAnalysis dataset.

### 5.4 Ablation Study

We conduct an extensive ablation study to analyze the effect of our ForgeryAnalysis dataset and each component and setting within ForgerySleuth framework. We report the pixel-level localization

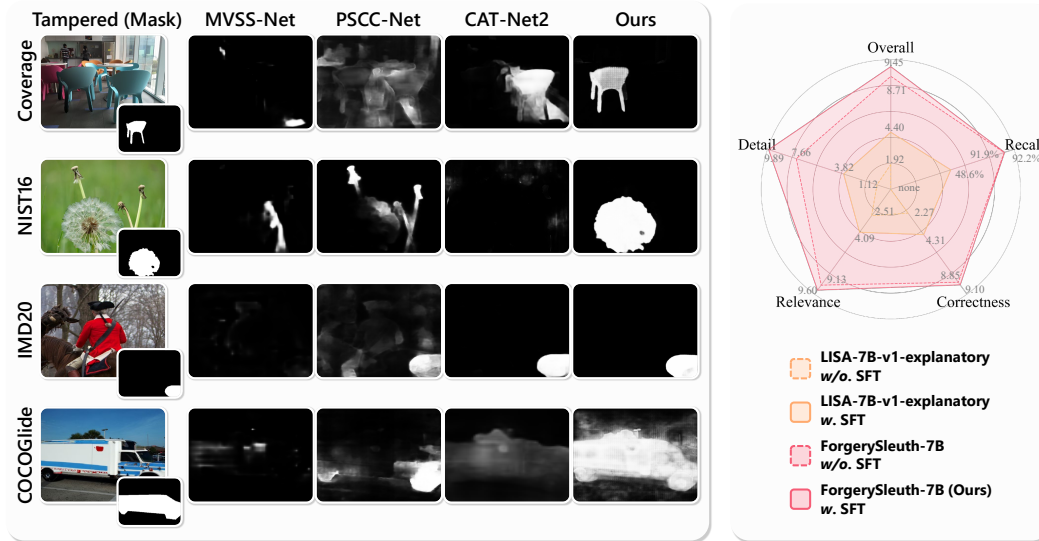


Figure 4: Visualization results comparing ForgerySleuth with exist- Figure 5: Forgery analysis results of the ablation study.

Table 4: Forgery analysis results of ForgerySleuth and SoTA methods, evaluated using STS.

| Model         | Dim | Ours         | GPT-4o | QWen2 | QWen  | LISA  |
|---------------|-----|--------------|--------|-------|-------|-------|
| MiniLM-L6-v2  | 384 | <b>0.926</b> | 0.725  | 0.595 | 0.551 | 0.313 |
| MiniLM-L12-v2 | 384 | <b>0.919</b> | 0.645  | 0.505 | 0.475 | 0.350 |
| mpnet-base-v2 | 768 | <b>0.961</b> | 0.724  | 0.635 | 0.546 | 0.401 |

Table 5: Ablation study on different parts of ForgeryAnalysis.

| Data             | NIST16       | IMD20        | COCOGlide    | ForgeryA.   |
|------------------|--------------|--------------|--------------|-------------|
| ForgeryA.-SFT    | 0.191        | 0.094        | 0.152        | 5.09        |
| ForgeryA.-PT     | 0.516        | 0.705        | <b>0.571</b> | 8.71        |
| ForgeryA.-PT&SFT | <b>0.518</b> | <b>0.710</b> | 0.562        | <b>9.45</b> |

performance on IMD datasets using the F1 score. We use ForgeryAnalysis-Eval to evaluate the quality of the reasoning text, following the scoring criteria described in Section 5.3.

**Contribution of ForgeryAnalysis.** The experimental results in Table 5 show the performance drop when ForgeryAnalysis-PT data is excluded during the pre-training phase, demonstrating the importance of large-scale data. Using ForgeryAnalysis-PT, the model achieves strong performance on the IMD task, even without the final stage of supervised fine-tuning with ForgeryAnalysis-SFT data. However, this final fine-tuning further enhances the quality of the tampering analysis text.

**Effect of Designed Components.** As shown in Table 6, removing the trace encoder  $\mathcal{F}_t$  significantly degrades performance across all datasets, confirming its critical contribution in capturing low-level trace features and mitigating hallucinations in the M-LLM. At the same time, the performance drop observed when the fusion mechanism is excluded highlights the effectiveness of our fusion strategy in integrating multimodal and trace-based information.

**Impact of Training Strategies.** In our comparison of various training strategies for the visual encoder  $\mathcal{F}_v$ , including trainable and LoRA fine-tuned, we discover that the best performance is achieved by keeping the encoder completely frozen. This could be attributed to the fact that trainable and LoRA fine-tuned strategies slightly diminish the generalization ability of the original SAM.

## 5.5 Qualitative Results

Figure 4 showcases pixel-level localization results across different datasets, comparing our framework with other SoTA methods. The masks are displayed without binarization to provide a more detailed view of the localization capability. Our method consistently delivers more precise tampered region detection with higher confidence across various types of manipulation. More detailed examples of clues and analysis provided by ForgerySleuth can be found in Section A.4. Across different types of manipulation, relevant clues and high-quality analysis demonstrate the effectiveness of M-LLM in capturing high-level semantic anomalies.

Table 6: Ablation study on different modules and settings, using pixel-level F1 with fixed threshold 0.5 as the evaluation metric.

| Setting                            | CASIA1       | NIST16       | IMD20        | COCOGlide    |
|------------------------------------|--------------|--------------|--------------|--------------|
| 1. w/o. Trace Enc. $\mathcal{F}_t$ | 0.637        | 0.323        | 0.622        | 0.395        |
| 2. w/o. Fusion Mechanism           | 0.628        | 0.463        | 0.649        | 0.513        |
| 3. w. $\mathcal{F}_v$ Trainable    | 0.755        | 0.451        | 0.716        | 0.526        |
| 4. w. $\mathcal{F}_v$ LoRA-ft.     | 0.766        | 0.493        | <b>0.731</b> | 0.547        |
| <b>ForgerySleuth</b>               | <b>0.804</b> | <b>0.518</b> | 0.710        | <b>0.562</b> |

## 6 Conclusion

In this work, we explored the potential of multimodal large language models in the image manipulation detection task. The proposed ForgerySleuth integrates M-LLMs with a trace encoder, allowing the model to utilize world knowledge to detect high-level semantic anomalies while effectively capturing low-level forgery traces. Additionally, we introduced a vision decoder with a fusion mechanism to integrate different features, ultimately producing precise segmentation masks. We also proposed a supervised fine-tuning dataset, ForgeryAnalysis, specifically designed for the IMD task. Each entry was initially generated by GPT-4o using a novel Chain-of-Clues prompt and then reviewed and refined by experts. Furthermore, we developed a data engine based on this dataset to automate forgery analysis, facilitating the creation of a larger-scale ForgeryAnalysis-PT dataset for pre-training purposes. A discussion of limitations can be found in Section E. We have already made the resources publicly available, including the data, code, and weights, to provide resources for advancing the field.

**Acknowledgment.** This work was supported in part by the National Natural Science Foundation of China (Grant 62472098). We would like to express our gratitude to Ruirui Tu and Xu Han for their valuable assistance with data annotation, which significantly contributed to the development of the ForgeryAnalysis dataset.

## References

- [1] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, “A survey on multimodal large language models,” *arXiv preprint arXiv:2306.13549*, 2023.
- [2] M. Moor, Q. Huang, S. Wu, M. Yasunaga, Y. Dalmia, J. Leskovec, C. Zakka, E. P. Reis, and P. Rajpurkar, “Med-flamingo: a multimodal medical few-shot learner,” in *Machine Learning for Health*, 2023, pp. 353–367.
- [3] A. Hu, Y. Shi, H. Xu, J. Ye, Q. Ye, M. Yan, C. Li, Q. Qian, J. Zhang, and F. Huang, “mplug-paperowl: Scientific diagram analysis with the multimodal large language model,” in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 6929–6938.
- [4] C. Cui, Y. Ma, X. Cao, W. Ye, Y. Zhou, K. Liang, J. Chen, J. Lu, Z. Yang, K.-D. Liao *et al.*, “A survey on multimodal large language models for autonomous driving,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 958–979.
- [5] V. Schetinger, M. M. Oliveira, R. da Silva, and T. J. Carvalho, “Humans are easily fooled by digital images,” *Computers & Graphics*, vol. 68, pp. 142–151, 2017.
- [6] S. J. Nightingale, K. A. Wade, and D. G. Watson, “Can people identify original and manipulated photos of real-world scenes?” *Cognitive Research: Principles and Implications*, vol. 2, pp. 1–21, 2017.
- [7] Z. Lu, D. Huang, L. Bai, J. Qu, C. Wu, X. Liu, and W. Ouyang, “Seeing is not always believing: benchmarking human and model perception of ai-generated images,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [8] Y. Wu, W. AbdAlmageed, and P. Natarajan, “Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9543–9552.

- [9] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia, "Span: Spatial pyramid attention network for image manipulation localization," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 312–328.
- [10] X. Chen, C. Dong, J. Ji, J. Cao, and X. Li, "Image manipulation detection by multi-view multi-scale supervision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 185–14 193.
- [11] Z. Sun, H. Jiang, D. Wang, X. Li, and J. Cao, "Saf-net: Semantic-agnostic feature learning network with auxiliary plugins for image manipulation detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 424–22 433.
- [12] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia, "Lisa: Reasoning segmentation via large language model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9579–9589.
- [13] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*, 2023, pp. 19 730–19 742.
- [14] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, 2024.
- [15] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [16] C. Wu, S. Yin, W. Qi, X. Wang, Z. Tang, and N. Duan, "Visual chatgpt: Talking, drawing and editing with visual foundation models," *arXiv preprint arXiv:2303.04671*, 2023.
- [17] R. Yang, L. Song, Y. Li, S. Zhao, Y. Ge, X. Li, and Y. Shan, "Gpt4tools: Teaching large language model to use tools via self-instruction," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [18] W. Wang, Z. Chen, X. Chen, J. Wu, X. Zhu, G. Zeng, P. Luo, T. Lu, J. Zhou, Y. Qiao *et al.*, "Visionllm: Large language model is also an open-ended decoder for vision-centric tasks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [19] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang, "An embodied generalist agent in 3d world," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 20 413–20 451.
- [20] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, and F. Wei, "Kosmos-2: Grounding multimodal large language models to the world," *arXiv preprint arXiv:2306.14824*, 2023.
- [21] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1849–1853, 2015.
- [22] B. Bayar and M. 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*, 2016, pp. 5–10.
- [23] R. Salloum, Y. Ren, and C.-C. J. Kuo, "Image splicing localization using a multi-task fully convolutional network (mfcn)," *Journal of Visual Communication and Image Representation*, vol. 51, pp. 201–209, 2018.
- [24] J. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, S. Chandrasekaran, A. Roy-Chowdhury, J. H. Lawrence Peterson *et al.*, "Detection and localization of image forgeries using resampling features and deep learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 69–77.
- [25] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of cfa artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, 2012.
- [26] B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image and vision computing*, vol. 27, no. 10, pp. 1497–1503, 2009.
- [27] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proceedings of the 7th workshop on Multimedia and security*, 2005, pp. 1–10.

- [28] Q. Liu, X. Cao, C. Deng, and X. Guo, "Identifying image composites through shadow matte consistency," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1111–1122, 2011.
- [29] E. Wengrowski, Z. H. Sun, and A. Hoogs, "Reflection correspondence for exposing photograph manipulation," in *IEEE International Conference on Image Processing*, 2017, pp. 4317–4321.
- [30] F. Guillaro, D. Cozzolino, A. Sud, N. Dufour, and L. Verdoliva, "Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 606–20 615.
- [31] S. Li, W. Ma, J. Guo, S. Xu, B. Li, and X. Zhang, "Unionformer: Unified-learning transformer with multi-view representation for image manipulation detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 12 523–12 533.
- [32] J. Wang, Z. Wu, J. Chen, X. Han, A. Shrivastava, S.-N. Lim, and Y.-G. Jiang, "Objectformer for image manipulation detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2364–2373.
- [33] Z. Xu, X. Zhang, R. Li, Z. Tang, Q. Huang, and J. Zhang, "Fakeshield: Explainable image forgery detection and localization via multi-modal large language models," in *International Conference on Learning Representations*, 2025.
- [34] C. Qu, Y. Zhong, C. Liu, G. Xu, D. Peng, F. Guo, and L. Jin, "Towards modern image manipulation localization: A large-scale dataset and novel methods," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 781–10 790.
- [35] J. Dong, W. Wang, and T. Tan, "Caisa image tampering detection evaluation database," in *IEEE China Summit and International Conference on Signal and Information Processing*, 2013, pp. 422–426.
- [36] G. Mahfoudi, B. Tajini, F. Retraint, F. Morain-Nicolier, J. L. Dugelay, and P. Marc, "Defacto: Image and face manipulation dataset," in *27th European Signal Processing Conference*, 2019, pp. 1–5.
- [37] S. Jia, M. Huang, Z. Zhou, Y. Ju, J. Cai, and S. Lyu, "Autosplice: A text-prompt manipulated image dataset for media forensics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 893–903.
- [38] W. Wang, J. Dong, and T. Tan, "A survey of passive image tampering detection," in *Digital Watermarking: 8th International Workshop*, 2009, pp. 308–322.
- [39] L. Zheng, Y. Zhang, and V. L. Thing, "A survey on image tampering and its detection in real-world photos," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 380–399, 2019.
- [40] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [41] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [42] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.
- [43] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.
- [44] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 371–10 381.
- [45] L. Chen, J. Li, X. Dong, P. Zhang, C. He, J. Wang, F. Zhao, and D. Lin, "Sharegpt4v: Improving large multi-modal models with better captions," *arXiv preprint arXiv:2311.12793*, 2023.

- [46] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, “Lora: Low-rank adaptation of large language models,” in *International Conference on Learning Representations*, 2022.
- [47] B. Bayar and M. C. Stamm, “Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2691–2706, 2018.
- [48] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [49] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*, 2020, pp. 213–229.
- [50] B. Cheng, A. Schwing, and A. Kirillov, “Per-pixel classification is not all you need for semantic segmentation,” *Advances in neural information processing systems*, vol. 34, pp. 17 864–17 875, 2021.
- [51] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, “Semantic understanding of scenes through the ade20k dataset,” *International Journal of Computer Vision*, vol. 127, pp. 302–321, 2019.
- [52] H. Caesar, J. Uijlings, and V. Ferrari, “Coco-stuff: Thing and stuff classes in context,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1209–1218.
- [53] X. Liu, Y. Liu, J. Chen, and X. Liu, “Psc-net: Progressive spatio-channel correlation network for image manipulation detection and localization,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7505–7517, 2022.
- [54] M.-J. Kwon, S.-H. Nam, I.-J. Yu, H.-K. Lee, and C. Kim, “Learning jpeg compression artifacts for image manipulation detection and localization,” *International Journal of Computer Vision*, vol. 130, no. 8, pp. 1875–1895, 2022.
- [55] Y.-F. Hsu and S.-F. Chang, “Detecting image splicing using geometry invariants and camera characteristics consistency,” in *IEEE International Conference on Multimedia and Expo*, 2006, pp. 549–552.
- [56] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, “Coverage—a novel database for copy-move forgery detection,” in *IEEE international conference on image processing*, 2016, pp. 161–165.
- [57] H. Guan, M. Kozak, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kheyrkhan, J. Smith, and J. Fiscus, “Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, 2019, pp. 63–72.
- [58] A. Novozamsky, B. Mahdian, and S. Saic, “Imd2020: A large-scale annotated dataset tailored for detecting manipulated images,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, 2020, pp. 71–80.
- [59] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.
- [60] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, “Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality,” March 2023. [Online]. Available: <https://lmsys.org/blog/2023-03-30-vicuna/>
- [61] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge *et al.*, “Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution,” *arXiv preprint arXiv:2409.12191*, 2024.
- [62] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, “Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3505–3506.

- [63] I. Loshchilov, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.

## A ForgeryAnalysis Dataset

### A.1 ForgeryAnalysis Data

We utilize the advanced M-LLM, GPT-4o, to generate the initial clue analyses, carefully designing prompts to ensure GPT-4o provides accurate and detailed responses. First, we inform GPT-4o of its role as an assistant, clearly outlining the levels of clues, along with specific examples, and specifying the task it needs to complete. The detailed prompt is as follows:

[ROLE] You are a rigorous and responsible image tampering (altering) detection expert. You can detect whether an image has been tampered with, localize the exact tampered region, and analyze your detection decision according to tampering clues of different levels. These clues include but are not limited to low-level pixel statistics (such as noise, color, textural, sharpness, and AI-generation fingerprint), middle-level visual defects (such as traces of tampered region or boundary, lighting inconsistency, perspective relationships, and physical constraints), and high-level semantic anomalies (such as content contrary to common sense, inciting and misleading content), etc. Altering operations could be divided into types, including “splice”, “copy-move”, “remove”, and “AI-generate”, leaving different clues that you should consider.

[TASK] Now, your task is to provide analysis. Please note that in real detection scenarios, you cannot know in advance whether an image has been tampered with and the specific tampered region. However, now I will tell you this information to help you conduct a more rigorous and accurate analysis based on this. There is no need to include all aspects and views in your analysis, give some of your most confident points.

The details of each conversation round are illustrated in Figure 12. In each round, we provide two images: the tampered image to be analyzed and a reference image with the tampered region highlighted. The prompt includes the <FAKE> token to indicate that the image is manipulated, specific tampering types to help the model focus on relevant clues, and a structured clue analysis format. Based on the response of GPT-4o, experts then conduct further revisions. The experts carefully review vague and incorrect statements that may arise from hallucinations, removing irrelevant clues and evidence. They also reorganize the clues into the Chain-of-Clues structure, which guides the reasoning process. This begins with “unveiling doubts” using high-level clues and continues with “pinpointing evidence” using middle-level and low-level features. The experts also check special tokens, such as <FAKE> and [SEG], to meet the requirements for subsequent model training.

### A.2 ForgeryAnalyst Engine

It is worth noting that the M-LLMs used in the data engine ForgeryAnalyst and the detection framework ForgerySleuth are different and independent. Although both M-LLMs are designed to analyze forgery and produce text-based clue analyses, their inputs and tasks are distinct. ForgeryAnalyst receives input that includes explicit information about the tampered region (highlighted to indicate tampering) with the goal of generating more precise and comprehensive clue analyses to construct pre-training data. In contrast, ForgerySleuth takes an image to be analyzed, aiming for the M-LLM to identify high-level semantic anomalies for detecting tampered regions. Furthermore, we design different prompts tailored to these two specific tasks. In our experiments, ForgeryAnalyst employs LLaVA-v1.5-13B, while the MLLM in ForgerySleuth uses LLaVA-7B-v1-1, balancing performance and efficiency.

We also designed a dedicated prompt for the data engine. In addition to indicating the image type as <FAKE> and specifying the tampering type [MANIPULATION-TYPE], we provide detailed instructions and examples of the Chain-of-Clues (CoC), as well as the required output data format. The specific prompt format is as follows:

You are a rigorous and responsible image tampering (altering) detection expert. You can localize the exact tampered region and analyze your detection decision according to tampering clues at different levels. Assuming that you have detected this is a <FAKE> image and the manipulation type is [MANIPULATION-TYPE], the exact tampered region boundary is highlighted with color in this image (and your detection IS correct).

Table 7: Statistic of our ForgeryAnalysis Dataset.

| Dataset            | Split                | Count | Sum   |
|--------------------|----------------------|-------|-------|
| ForgeryAnalysis    | ForgeryAnalysis-Eval | 618   | 2,370 |
|                    | ForgeryAnalysis-SFT  | 1752  |       |
| ForgeryAnalysis-PT | -                    | 50k   | 50k   |

Table 8: Overview of public dataset utilized in the construction of ForgeryAnalysis and for evaluation.

| Dataset         | Usage            | Authentic | Tampered | Manipulation Types            | Source      |
|-----------------|------------------|-----------|----------|-------------------------------|-------------|
| MIML [34]       | Train, Construct | 11,142    | 123,150  | Manual Editing                | PSBattles   |
| CASIA2 [35]     | Train, Construct | 7,491     | 5,063    | Splice, Copy-move             | Corel       |
| DEFACTO [36]    | Train, Construct | -         | 149,587  | Splice, Copy-move, AIGC (GAN) | MS COCO     |
| AutoSplice [37] | Train, Construct | 2,273     | 3,621    | AIGC                          | Visual News |
| Columbia [55]   | Eval             | 183       | 180      | Splice                        | -           |
| Coverage [56]   | Eval             | 100       | 100      | Copy-move                     | -           |
| CASIA1 [35]     | Eval             | 800       | 920      | Splice, Copy-move             | Corel       |
| NIST16 [57]     | Eval             | -         | 564      | Splice, Copy-move             | -           |
| IMD20 [58]      | Eval             | 414       | 2,010    | Manual Editing                | -           |
| COCOGlide [30]  | Eval             | 512       | 512      | AIGC (Diffusion)              | -           |

Please provide the chain-of-clues supporting your detection decision in the following style: # high-level semantic anomalies (such as content contrary to common sense, inciting and misleading content), # middle-level visual defects (such as traces of tampered region or boundary, lighting inconsistency, perspective relationships, and physical constraints) and # low-level pixel statistics (such as noise, color, textural, sharpness, and AI-generation fingerprint), where the high-level anomalies are significant doubts worth attention, and the middle-level and low-level findings are reliable evidence.

### A.3 Statistics

Table 7 presents the data statistics of the ForgeryAnalysis dataset. ForgeryAnalysis-Eval and ForgeryAnalysis-SFT are initially generated by GPT-4o and fully revised by experts. They are used for evaluating the quality of manipulation analysis generated by the M-LLMs and for the final supervised fine-tuning, respectively. ForgeryAnalysis-PT is automatically constructed by our proposed data engine, ForgeryAnalyst, maintaining consistency in data format with the other subsets. Table 8 provides detailed statistics on the scale, source, and manipulation types from each public dataset utilized in the construction of ForgeryAnalysis and for evaluation.

### A.4 Cases

Figures 13, 14, 15 and 16 present examples of analysis texts from ForgeryAnalysis-Eval for four different tampering types. These diverse cases highlight the variations in detectable clues across different tampering types and illustrate how varying levels of clues support manipulation detection. The analyses in ForgeryAnalysis-Eval are cross-checked by multiple experts to ensure comprehensive and accurate clues. Figure 17 shows analysis texts in ForgeryAnalysis-PT. Although the subset is automatically generated by the data engine, it also provides precise descriptions and analyses of the tampered regions.

## B ForgerySleuth

### B.1 Framework Details

**Trace Encoder.** Considering that the vision backbone (ViT-H SAM backbone) is pre-trained on tasks highly correlated with semantics and remains frozen during the fine-tuning of our framework, the semantic-agnostic features widely used in IMD tasks [8, 9, 10, 11] cannot be effectively leveraged. We propose an independent trace encoder  $\mathcal{F}_t$ , equipped with a noise enhancement module to focus on low-level features and provide more reliable tampering evidence. Specifically, the noise enhancement module, positioned at the front of the encoder, uses constrained convolutions [47] to compute local

Table 9: Manipulation localization results comparing ForgerySleuth with SoTA methods.

| Method                   | Optimal Threshold F1 |              |              |              |              | Fixed Threshold F1 (0.5) |              |              |              |              |
|--------------------------|----------------------|--------------|--------------|--------------|--------------|--------------------------|--------------|--------------|--------------|--------------|
|                          | Columbia             | Coverage     | CASIA1       | NIST16       | COCOglide    | Columbia                 | Coverage     | CASIA1       | NIST16       | COCOglide    |
| Mantra-Net [8]           | 0.650                | 0.486        | 0.320        | 0.225        | 0.673        | 0.508                    | 0.317        | 0.180        | 0.172        | 0.516        |
| SPAN [9]                 | 0.873                | 0.428        | 0.169        | 0.363        | 0.350        | 0.759                    | 0.235        | 0.112        | 0.228        | 0.298        |
| MVSS-Net [10]            | 0.781                | 0.659        | 0.650        | 0.372        | 0.642        | 0.729                    | 0.514        | 0.528        | 0.320        | 0.486        |
| PSCC-Net [53]            | 0.760                | 0.615        | 0.670        | 0.210        | 0.685        | 0.604                    | 0.473        | 0.520        | 0.113        | 0.515        |
| CAT-Net2 [54]            | 0.923                | 0.582        | 0.852        | 0.417        | 0.603        | 0.859                    | 0.381        | 0.752        | 0.308        | 0.434        |
| TruFor [30]              | 0.914                | 0.735        | 0.822        | 0.470        | 0.720        | 0.859                    | 0.600        | 0.737        | 0.399        | 0.523        |
| UnionFor. [31]           | 0.925                | 0.720        | 0.863        | 0.489        | 0.742        | 0.861                    | 0.592        | 0.760        | 0.413        | 0.536        |
| FakeShield [33]          | 0.306                | 0.085        | 0.620        | 0.119        | 0.659        | 0.285                    | 0.052        | 0.566        | 0.099        | 0.536        |
| - only <i>TP</i> samples | 0.874                | 0.470        | 0.696        | 0.514        | 0.659        | 0.813                    | 0.289        | 0.635        | 0.431        | 0.536        |
| <b>ForgerySleuth</b>     | <b>0.931</b>         | <b>0.792</b> | <b>0.870</b> | <b>0.610</b> | <b>0.751</b> | <b>0.925</b>             | <b>0.684</b> | <b>0.804</b> | <b>0.518</b> | <b>0.562</b> |

Table 10: Manipulation localization results of ForgerySleuth and SoTA methods, using pixel-level AUC as the evaluation metric.

| Method                   | Columbia     | Coverage     | CASIA1       | NIST16       | IMD20        |
|--------------------------|--------------|--------------|--------------|--------------|--------------|
| Mantra-Net [8]           | 0.824        | 0.819        | 0.817        | 0.795        | 0.748        |
| SPAN [9]                 | 0.936        | 0.922        | 0.797        | 0.840        | 0.750        |
| PSCC-Net [53]            | 0.982        | 0.847        | 0.829        | 0.855        | 0.806        |
| ObjectFor. [32]          | 0.955        | 0.928        | 0.843        | 0.872        | 0.821        |
| TruFor [30]              | 0.947        | 0.925        | 0.957        | 0.877        | -            |
| UnionFor. [31]           | 0.989        | 0.945        | <b>0.972</b> | 0.881        | 0.860        |
| FakeShield [33]          | 0.323        | 0.137        | 0.787        | 0.185        | 0.780        |
| - only <i>TP</i> samples | 0.924        | 0.761        | 0.883        | 0.801        | 0.868        |
| <b>ForgerySleuth</b>     | <b>0.992</b> | <b>0.962</b> | 0.969        | <b>0.898</b> | <b>0.911</b> |

differences, extract noise features, and suppress image content. The convolution kernel constraints are defined as follows:

$$\begin{cases} \omega_{(0,0)} = 1, \\ \sum_{(m,n)} \omega_{(m,n)} = 0, \end{cases} \quad (9)$$

where  $(m, n)$  denotes the spatial index of the values within the convolution kernel, with  $(0, 0)$  positioned at the center. The constrained convolutions are still trainable, allowing them to learn manipulation features more adaptively than fully fixed-parameter noise extractors. The extracted noise features further enhance the original features by residual connections. The encoder employs a ViT-B architecture and all parameters are fine-tuned during training.

In summary, we employ the trace encoder  $\mathcal{F}_t$  to capture low-level manipulation features, leveraging constrained convolutions within the noise enhancement module to achieve this. Meanwhile, the vision backbone  $\mathcal{F}_v$  utilizes pre-trained parameters from SAM, which is widely recognized for its ability to capture dense visual content features, thereby supporting more precise segmentation.

**Fusion Mechanism.** To integrate the image content embedding  $\mathbf{f}_c$ , trace embedding  $\mathbf{f}_t$ , and the LLM output tokens  $\mathbf{h}_{\text{seg}}$  obtained from the M-LLM, we take inspiration from Transformer segmentation models [49, 50, 42] and design a vision decoder with a fusion attention mechanism, illustrated in Figure 3 (b). The mechanism consists of three layers, with each layer performing four steps: self-attention on the LLM output tokens or upgraded tokens, cross-attention from tokens (as queries) to the trace or image embeddings, point-wise MLP, and cross-attention from the trace or image embeddings (as queries) back to the tokens. The first layer of the module computes attention between anomalies in the LLM output tokens and traces in the trace embeddings, facilitating the organization and pinpointing of clues. The subsequent layers focus on attention between refined clues in the upgraded tokens and content in the image embeddings, enabling more precise segmentation of the tampered regions.

## C Comparison with FakeShield

### C.1 Details

FakeShield [33] is a concurrent work with the same motivation as ours. It proposes a multimodal large model for image manipulation detection. This further reinforces the significance of providing a reasonable forgery analysis, alleviating the explainability issues in existing image manipulation detection methods. However, there are some fundamental differences between FakeShield and our approach.

From a data perspective, FakeShield relies on GPT-4o to generate tampering analysis texts, constructing “image-mask-description” triplets for training and evaluation. However, this process presents certain challenges: First, the generated texts lack structured reasoning, as FakeShield does not impose explicit guidance on how the detected clues is organized and analysis data is formulated. Second, reliance on GPT-4o alone makes data prone to hallucinations, potentially introducing unreliable or misleading explanations.

In contrast, we propose a three-level Chain-of-Clues (CoC) structure, which enables a more structured and interpretable analysis. Our design is based on two key insights: 1) Previous research [38, 39] suggests that forensic clues are naturally organized hierarchically. 2) Studies on Chain-of-Thought (CoT) prompting [40, 41] have demonstrated its effectiveness in improving reasoning within large language models. Inspired by these findings, we structure the tampering analysis data into a progressive reasoning process, encouraging the model to identify, refine, and interpret forensic clues at different levels. Additionally, to reduce hallucinations and improve reliability, we manually verify the generated analysis texts and expand the dataset using an additional data engine.

From a model perspective, FakeShield employs a multimodal forgery localization module (MFLM) that leverages the segment anything model (SAM) and a large language model to localize tampered regions based on their general vision and language features. However, recent studies [10, 54, 11] highlight the crucial role of low-level noise features in image manipulation detection, while pre-trained models for general vision tasks primarily capture high-level semantic features. To address this challenge, we introduce a trace encoder with noise enhancement to compensate for the limitations of the base vision model. In addition, we design a dedicated fusion mechanism to better integrate multiple features. Ablation studies in Table 6 further validate the effectiveness of these components.

In the following experiments, we comprehensively compare FakeShield and our method, ForgerySleuth, demonstrating that ForgerySleuth consistently achieves superior performance in various evaluation aspects.

### C.2 Evaluation

**Experimental Setting.** We reproduce the FakeShield model using its publicly released official code repository and pre-trained weights. The following comparative experiments adopt the same experimental settings and evaluation metrics as described in Section 5.1.

We follow established practices [31] by using optimal threshold and fixed threshold F1 scores and the threshold-independent Area Under the Curve (AUC) metric. Inspired by prior work [60, 14], we use GPT-4 as an automated evaluator to assess the reasoning performance of different models. To address concerns regarding potential limitations of GPT-4, we incorporate an additional metric, semantic textual similarity (STS). Specifically, following SBERT [59], we use STS to measure the similarity between the generated analysis and the ground-truth text in ForgeryAnalysis-Eval.

**Manipulation Detection Results.** Tables 9 and 10 present the performance of ForgerySleuth, FakeShield, and other SoTA methods in the manipulation localization task, reporting F1 scores and AUC metrics, respectively. It is essential to note that the official implementation of FakeShield follows a two-stage pipeline. First, it determines whether an image has been manipulated. Only if an image is classified as fake does it proceed to generate a tampering mask. In contrast, ForgerySleuth and other detection methods output a mask for all images while also providing a real/fake classification result.

We adopt the following strategy for FakeShield to maintain consistency with other methods and ensure a fair comparison. If FakeShield classifies an image as real, we assume that the predicted mask is an all-zero matrix (*i.e.*, indicating no tampered region). The corresponding results are

Table 11: Forgery analysis results of ForgerySleuth and SoTA methods, evaluated using GPT-4 on ForgeryAnalysis-Eval dataset.

| Method               | Recall       | Correctness | Relevance   | Detail      | Overall     |
|----------------------|--------------|-------------|-------------|-------------|-------------|
| LISA [12]            | -            | 2.27        | 2.51        | 1.12        | 1.92        |
| QWen                 | 43.9%        | 3.22        | 2.92        | 3.10        | 3.41        |
| QWen2 [61]           | 29.6%        | 3.70        | 4.14        | 2.57        | 3.34        |
| GPT-4o               | 68.8%        | 7.14        | 7.97        | 5.84        | 6.96        |
| FakeShield [33]      | 52.8%        | 5.91        | 6.35        | 4.67        | 5.55        |
| <b>ForgerySleuth</b> | <b>92.2%</b> | <b>9.10</b> | <b>9.60</b> | <b>9.89</b> | <b>9.45</b> |

Table 12: Forgery analysis results of ForgerySleuth and SoTA methods, evaluated using STS.

| Model             | Ours         | FakeShield | GPT-4o | QWen2 | QWen  | LISA  |
|-------------------|--------------|------------|--------|-------|-------|-------|
| all-MiniLM-L6-v2  | <b>0.926</b> | 0.662      | 0.725  | 0.595 | 0.551 | 0.313 |
| all-MiniLM-L12-v2 | <b>0.919</b> | 0.514      | 0.645  | 0.505 | 0.475 | 0.350 |
| all-mpnet-base-v2 | <b>0.961</b> | 0.695      | 0.724  | 0.635 | 0.546 | 0.401 |

shown in the FakeShield row. The notably lower accuracy stems from the low recall rates, where FakeShield fails to detect manipulations, leading to significant missed detections. To further analyze FakeShield’s localization performance, we also report its performance on only the successfully detected manipulated images (*i.e.*, True Positive samples). As indicated in the row for only true positive samples, its performance improves significantly under this setting, but it still falls short compared to ForgerySleuth. However, since manipulation detection is performed without prior knowledge of whether an image is manipulated, the first setting (which considers all images) provides a more realistic and fair evaluation, while the second setting serves only as a supplementary analysis.

**Forgery Analysis Results.** We employ GPT-4 as an evaluator on the ForgeryAnalysis-Eval dataset to assess the quality of textual analysis and reasoning. The evaluation considers several key dimensions: correctness, relevance, and detail. The results are presented in Table 11 (corresponding to Figure 1 (b)). FakeShield exhibits low recall rates, further corroborating the previously discussed issue of missed detections. Additionally, its analysis falls behind ForgerySleuth in both accuracy and level of detail.

However, since GPT-4 itself is susceptible to hallucinations, this approach does not guarantee a fully objective and quantitative evaluation. To address this limitation, we incorporate Semantic Textual Similarity (STS) to measure the similarity between the generated text and the ground-truth text in ForgeryAnalysis-Eval. The results, shown in Table 12, further confirm that our ForgerySleuth consistently outperforms FakeShield.

**Qualitative Results.** We present the performance of ForgerySleuth and FakeShield in manipulation localization and forgery analysis, as shown in Figure 8. The results demonstrate that ForgerySleuth not only achieves higher accuracy in localizing manipulated regions, but also generates more detailed and precise analysis.

## D Experiment

### D.1 Experimental Implementation Details

**Implementation Details.** We employ LLaVA-7B-v1-1 [14] as the base multimodal LLM ( $\mathcal{F}_m$ ) and use the ViT-H SAM [42] backbone for the vision encoder ( $\mathcal{F}_v$ ). For training, we utilize 2 NVIDIA 80GB A800 GPUs, with training scripts optimized by DeepSpeed [62], which helps reduce memory usage and accelerate training. We use the AdamW [63] optimizer, setting the learning rate to 0.0002 with no weight decay. The learning rate is scheduled using WarmupDecayLR, with 100 warmup iterations. The weights for the text generation loss  $\lambda_{\text{txt}}$  and mask loss  $\lambda_{\text{mask}}$  are both set to 1.0, while the BCE loss  $\lambda_{\text{bce}}$  and DICE loss  $\lambda_{\text{dice}}$  are weighted at 1.0 and 0.2, respectively. The batch size per device is 4, with gradient accumulation steps set to 4.

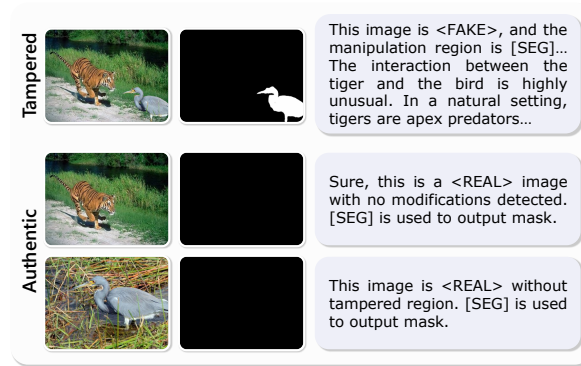


Figure 6: Predictions of ForgerySleuth on authentic and tampered images.

Table 13: Human evaluation results for the forgery analysis task.

| Method      | Samples | Recall       | Correctness | Relevance   | Detail      | Overall     |
|-------------|---------|--------------|-------------|-------------|-------------|-------------|
| QWen2       | 139     | 35.3%        | 4.95        | 5.10        | 2.85        | 4.11        |
| GPT-4o      | 134     | 65.7%        | 8.06        | 8.32        | 5.77        | 7.18        |
| <b>Ours</b> | 147     | <b>91.2%</b> | <b>9.48</b> | <b>9.53</b> | <b>9.61</b> | <b>9.44</b> |

## D.2 Manipulation Detection Results

Modeling the natural laws of real samples and distinguishing them from tampered images is crucial. We utilized two special tokens, <REAL> and <FAKE>, which indicate the detection results. In addition, the output mask provides valuable information for verifying image authenticity. Figure 6 illustrates the ForgerySleuth detection results for authentic and manipulated images, including the predicted tampering masks and textual analysis. Our method achieves real/fake classification accuracies of 0.989 on Columbia and 0.910 on CASIA1, demonstrating its effectiveness in manipulation detection.

## D.3 Forgery Analysis Results

We leverage GPT-4 for evaluation based on the ForgeryAnalysis-Eval dataset to assess the quality of text reasoning and explanations. Ratings are based on several dimensions, including the correctness of tampered objects, the relevance of clues to manipulation, and the detail of analysis, reflecting the capability of comprehension, reasoning, and correctness. We collect responses from each M-LLM, and GPT-4 assigns scores from 1 to 10 for each response, with higher scores indicating better performance. Beyond scoring, GPT-4 provides explanations for its ratings, ensuring transparency and consistency in the evaluation process. Figure 18 illustrates the prompt structure used for the evaluation and the response of GPT-4. To ensure consistent and fair scoring, the evaluation prompt includes clear scoring criteria for assessment aspects, including correctness, relevance, and details. GPT-4 assigns a score for each evaluation dimension and provides detailed comments to justify the rating.

To directly assess the ability to detect manipulated images, we explicitly instruct the evaluated M-LLMs through prompts to additionally output <FAKE> or <REAL> to indicate their detection results. All models, except LISA-7B-v1-explanatory, can provide the required response. We use the recall rate to reflect the ability to correctly identify tampered images.

**Human Evaluations.** We also conduct human evaluations under the same experimental setup as the evaluation using GPT-4 described above. A total of 14 volunteers participated, each scoring for 30 random samples. The results are summarized in Table 13.

## D.4 Qualitative Results

We present additional results of our ForgerySleuth in both forgery analysis and manipulation localization tasks. We also provide forgery analysis results from existing M-LLMs (GPT-4o and Qwen-VL)

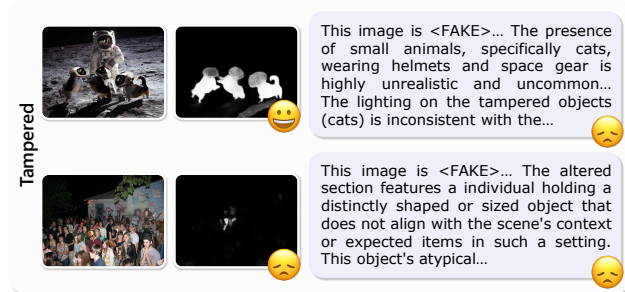


Figure 7: Failure cases in predictions of ForgerySleuth.

and segmentation masks from traditional image manipulation detection (IMD) methods, illustrating the advantages of our proposed IMD assistant in terms of accuracy and explainability. These cases are from public datasets IMD20 and NIST16, demonstrating the generalization capabilities of our method.

In the example in Figure 9, both M-LLMs classify the image as real without detecting manipulation. Similarly, most IMD methods, except CAT-Net2, fail to localize the manipulated regions accurately. However, ForgerySleuth identifies the tampered regions and provided a detailed tampering analysis. Figure 10 presents a case with more apparent tampering traces. GPT-4o and Qwen-VL both exhibit varying degrees of overthinking, leading to inaccurate analysis. Our method demonstrates higher localization precision and analytical accuracy. In the case of Figure 11, ForgerySleuth precisely localizes the tampered regions, even including the shadow of the person. Compared to the textual outputs of M-LLMs, the mask generated by ForgerySleuth provides a more intuitive and accurate detection result.

## E Limitation and Future Work

### E.1 Failure Cases

We analyzed ForgerySleuth's detection failures and identified two representative error patterns.

**Semantic Description Inaccuracies.** In these cases, our model can accurately localize manipulated regions, but its textual analysis contains description errors. For instance, in the first example in Figure 7, the model misidentifies dogs as cats. Importantly, this does not impact the detection logic or localization accuracy. These errors often arise when specific semantic concepts in images are challenging to detect clearly. Future work could improve this by enhancing the visual understanding capabilities of MLLM and LLM backbone models.

**Failures with Small Manipulations in Complex Scenes.** In complex image content with small manipulated regions, the model's localization precision can degrade, sometimes leading to incorrect textual analysis and, consequently, a detection failure. For example, in the second instance in Figure 7, where a spliced cartoon character is present, the model only localized a partial manipulation (*e.g.*, the guitar in hand) while overlooking the character itself, primarily due to the small proportion of the manipulation and the image's overall complexity. Detecting small region manipulations remains a significant challenge, and MLLMs tend to focus on global semantic information rather than fine-grained local details. Therefore, developing methods to effectively detect anomalies in small regions is a crucial research direction.

### E.2 Constraints in Evaluation Methods

Our paper evaluates analysis accuracy using GPT-4o and STS methods on the ForgeryAnalysis-eval dataset. However, our evaluation methods are subject to specific constraints: our assessment relies on GPT-4o, a large external model that may exhibit inherent biases or hallucinations, and STS only reflects semantic similarity between two sentences. Thus, exploring more comprehensive and objective evaluation methods, including the quantitative assessment of model hallucinations, is an important direction for future work.

### **E.3 Model Scale and Inference Costs**

While leveraging LLMs to complement traditional image manipulation detection methods, which often primarily focus on low-level traces, this approach increases the model's scale and computational cost of generating detailed explanations. In the future, we plan to investigate model light-weighting solutions. This includes developing a version where generating such detailed output is optional, which would significantly reduce inference time. Alternatively, we could explore distilling LLM capabilities into smaller models, retaining only the essential knowledge for manipulation detection.



Figure 8: Comparison of results between ForgerySleuth and FakeShield.



The image appears to be real, depicting a scene from a roller derby event. The participants are wearing protective gear such as helmets, knee pads, and elbow pads, which is typical for the sport. The expressions and body language of the individuals suggest they are actively engaged in the game. There are no obvious signs of digital manipulation, such as inconsistent lighting, shadows, or distorted proportions that would indicate the image has been altered. Additionally, the background shows spectators watching the event, adding to the authenticity of the scene.



The image appears to be authentic based on several factors:

1. **Lighting and Shadows**: The lighting is consistent throughout the image. Shadows and highlights align with the assumed light sources.
2. **Proportions and Perspective**: The subjects' proportions and perspective match the setting. There are no noticeable inconsistencies with size or angle that often indicate manipulation.
3. **Blending and Edges**: The edges of the subjects are clean and well-defined with no visible signs of cut-and-paste operations, such as blurred or overly sharp outlines.
4. **Environment and Interaction**: The people in the scene interact naturally with each other and their surroundings. Details like the roller skates' contact with the floor and the interaction between individuals are logical and cohesive.

Overall, the elements in the image work together harmoniously, suggesting it is not manipulated.



This is a <FAKE> image with modifications detected in the [SEG]. We have found several clues, where high-level anomalies create substantial doubts, and other-level findings contribute reliable evidence.

#### # High-Level Semantic Anomalies

##### 1. **Content Contrary to Common Sense**:

- The presence of a man in casual clothes amidst a group of roller derby players seems unusual, which raises suspicion. It is uncommon for an amateur player to be seen in the middle of the action along with the players.

#### # Middle-Level Visual Defects

##### 1. **Lighting Inconsistency**:

- The lighting on the tampered figure does not align with the lighting conditions of the surrounding environment. For instance, the shadows and highlights on the amateur player do not match the expected lighting direction and intensity observed on the other roller derby participants.

##### 2. **Perspective Relationships**:

- The perspective of the amateur player is slightly misaligned with the rest of the group. The positioning and relative size of the amateur player do not match naturally with the surrounding players, indicating that the figure was not part of the original scene.

##### 3. **Physical Constraints**:

- The amateur player's interaction with the other players is awkward. There is a lack of natural contact or engagement between the referee and the surrounding skaters, suggesting that the amateur player was added without considering the physical dynamics of the scene.

#### # Low-Level Pixel Statistics

##### 1. **Noise Inconsistency**:

- The tampered region shows a different noise pattern compared to the surrounding areas. The amateur player has a smoother texture that does not match the natural noise present in the background.

##### 2. **Color Discrepancies**:

- There are subtle differences in color grading between the tampered figure and the rest of the image. The amateur player has skin tones and lighting that do not perfectly match the other elements in the scene.

##### 3. **Sharpness Variations**:

- The edges of the amateur player are unnaturally sharp compared to the rest of the image. This disparity in sharpness indicates that the figure was cut and pasted from another source.

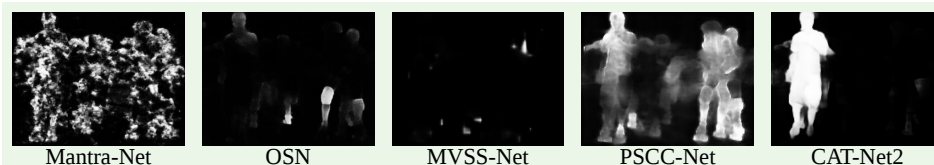
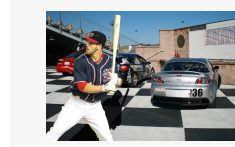


Figure 9: Comparison of results between our ForgerySleuth, existing multimodal large language models, and traditional image manipulation detection methods. The image is c93gg4y\_0.png from the IMD20 dataset.



Figure 10: Comparison of results between our ForgerySleuth, existing multimodal large language models, and traditional image manipulation detection methods. The image is c8zf0hg\_0.jpg from the IMD20 dataset.



The image appears to be manipulated for several reasons:

1. **Foreground Subject**: The person holding the baseball bat is not naturally integrated into the scene. The lighting, shadows, and overall quality of the person do not match the background. The edges around the person, especially at the bottom, appear to have been cut out digitally.
2. **Background Context**: The setting looks like a race track or a car show, which is an unusual place for someone to be playing baseball. This mismatch in context suggests that the person has been added to the image artificially.
3. **Lighting and Shadows**: The lighting on the person does not align with the lighting in the rest of the image. The shadows cast by the cars and other objects in the background do not correspond with the position of the person.
4. **Image Quality**: There is a noticeable difference in the resolution and clarity between the person and the background. The person's image is sharper and more detailed compared to the slightly blurred background.

These factors collectively indicate that the image has been manipulated.



<FAKE>

Analysis:

1. **Lighting and Shadows**: The shadow of the person holding the bat does not align well with the other shadows in the scene, indicating a possible manipulation. The shadow is also very sharp and dark compared to the rest of the scene, which suggests inconsistency.
2. **Alignment and Perspective**: The person with the bat does not appear to be integrated smoothly into the environment. The size and angle of the figure seem off compared to the surrounding elements, like the cars.
3. **Integration with Background**: The person holding the bat appears to be layered on top of the existing background. The edge around the figure looks too clean and sharp, lacking the natural blending seen in real-world images.
4. **Context Clarity**: A person in a baseball uniform standing in a car racing environment seems out of context, adding to the suspicion of image manipulation.



This is a <FAKE> image with modifications detected in the [SEG]. We have found several clues, where high-level anomalies create substantial doubts, and other-level findings contribute reliable evidence.

#### # High-Level Semantic Anomalies

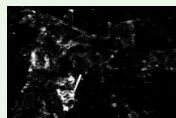
1. **Content Contrary to Common Sense**: The presence of a professional baseball player in a full uniform, standing in a parking lot next to sports cars, is contextually odd. Such a scenario is improbable, suggesting that the image was staged or the player was placed into this environment unnaturally.
2. **Misleading Content**: The image may give the impression that the athlete is involved in or supports a particular car brand, which could be considered advertorial content without clear disclosure.

#### # Middle-Level Visual Defects

1. **Lighting Inconsistency**: The lighting on the athlete does not align with the lighting direction and intensity of the cars and background. Shadows and highlights on the athlete do not correspond naturally with the sunlight and shadows cast by the cars, indicating manipulation.
2. **Boundary Traces**: Noticeable edges where the athlete meets the background indicate possible cut-and-paste manipulation. These boundary artifacts are not smooth and visible, suggesting improper blending.
3. **Perspective Relationships**: The athlete is standing in a manner that does not fit naturally within the scene's perspective. The athlete's position and the angle are not aligned with the cars and the background, indicating that the image of the athlete was inserted from a different source.

#### # Low-Level Pixel Statistics

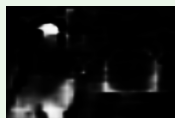
1. **Noise Inconsistency**: The noise pattern in the tampered region differs from that of the surrounding background. The athlete's section has a different grain or noise texture than the rest of the image, suggesting manipulation.
2. **Color Discrepancy**: The color tones and saturation levels of the athlete do not match the surrounding environment. The lighting on the athlete's uniform does not correspond with the overall lighting direction of the scene, indicating tampering.
3. **Sharpness/Blurriness**: There is a noticeable difference in sharpness between the tampered region and the background. The edges around the athlete are too sharp compared to the rest of the image, suggesting manipulation.



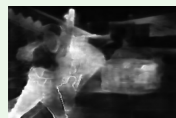
Mantra-Net



OSN



MVSS-Net



PSCC-Net



CAT-Net2

Figure 11: Comparison of results between our ForgerySleuth, existing multimodal large language models, and traditional image manipulation detection methods. The image is NC2016\_7312.jpg from the NIST16 dataset.

Assuming that you have detected this is a **<FAKE>** image with **"AI-generate"** manipulation, and the exact tampered region is highlighted in color in another reference image (and your detection IS correct). Please analyze your detection decision according to clues of different levels and aspects including but not limited to low-level pixel statistics (such as noise, color, textural, sharpness, and AI-generation fingerprint), middle-level visual defects (such as traces of tampered region or boundary, lighting inconsistency, perspective relationships, and physical constraints), and high-level semantic anomalies (such as content contrary to common sense, inciting and misleading content), etc. There is no need to include all aspects and views in your analysis, give some of your most confident points.



This is a FAKE image manipulated using AI-generated content, specifically targeting the bus in the center of the image, which is highlighted in red in the reference. Below is a breakdown of the detection clues across various levels:

#### Low-Level Pixel Statistics:

- Noise and Texture Discrepancies: The overall scene, which depicts a post-disaster landscape, contains natural graininess, subtle noise, and rough textures consistent with debris and rubble. However, the bus in the middle shows different characteristics, it appears smoother and lacks the same level of noise as its surroundings, indicating that it was generated or inserted separately. This noise discrepancy is a common marker of AI-generated content.
- Color and Sharpness Anomalies: The bus has unnaturally vivid colors (yellow and black), which stand out in contrast to the muted tones of the devastated environment. The sharpness of the bus also does not match the slightly blurred background, implying that it was not part of the original scene. This sharpness inconsistency indicates a different image source or digital generation.

#### Middle-Level Visual Defects:

- Boundary Traces: The edges of the bus appear unnaturally smooth, especially around the rear end and where it blends into the background. This smoothness is in stark contrast to the jagged debris and uneven structures in the environment. The lack of blending or transition between the bus and its surroundings, especially along its base, highlights its insertion into the scene.
- Lighting Inconsistency: The overall lighting in the image is diffused and soft, typical of overcast post-disaster conditions. The bus, however, shows inconsistent lighting, with reflective surfaces that suggest stronger light sources that do not exist in the surrounding scene. The shadows on the bus are less pronounced and don't match the diffused light in the image, further indicating manipulation.
- Perspective and Physical Constraints: The bus appears floating or elevated, which defies the physical laws of gravity and common sense. There is no clear indication of support or ground contact for the bus, suggesting it was artificially placed in this environment. Moreover, the size and angle of the bus do not match the perspective of the background elements, further reinforcing the hypothesis that it was generated and inserted unnaturally.

#### High-Level Semantic Anomalies:

- Contrary to Common Sense: From a high-level contextual perspective, the bus being suspended in such a chaotic, disaster-stricken environment is illogical. Buses do not float mid-air, and there are no physical cues in the scene (such as debris lifting or supporting the bus) to explain its position. This is a direct indication of content manipulation.
- Inappropriate Object Placement: In a disaster scenario like this, objects like vehicles are usually scattered, damaged, or buried within the debris. The bus, however, appears undamaged and disproportionately out of place. The placement of the bus in this specific location within the scene is highly unnatural and lacks coherence with the overall narrative of destruction.

#### Conclusion:

This image has been manipulated using AI-generated content, with the tampered region being the floating bus in the middle. The analysis of noise patterns, boundary traces, lighting inconsistencies, and semantic illogic clearly indicate that the bus was artificially inserted into the scene. This manipulation disrupts the overall realism of the disaster environment and highlights the telltale signs of AI-generated imagery.

This image is **<FAKE>**, and the tampered region is **[SEG]**. We have identified the following clues, where the high-level anomalies are significant doubts worth attention, and the middle-level and low-level findings are reliable evidence.

#### # High-Level Semantic Anomalies

- Contrary to Common Sense: From a high-level contextual perspective, the bus being suspended in such a chaotic, disaster-stricken environment is illogical. Buses do not float mid-air, and there are no physical cues in the scene (such as debris lifting or supporting the bus) to explain its position.
- Inappropriate Object Placement: In a disaster scenario like this, objects like vehicles are usually scattered, damaged, or buried within the debris. The bus, however, appears undamaged and disproportionately out of place.

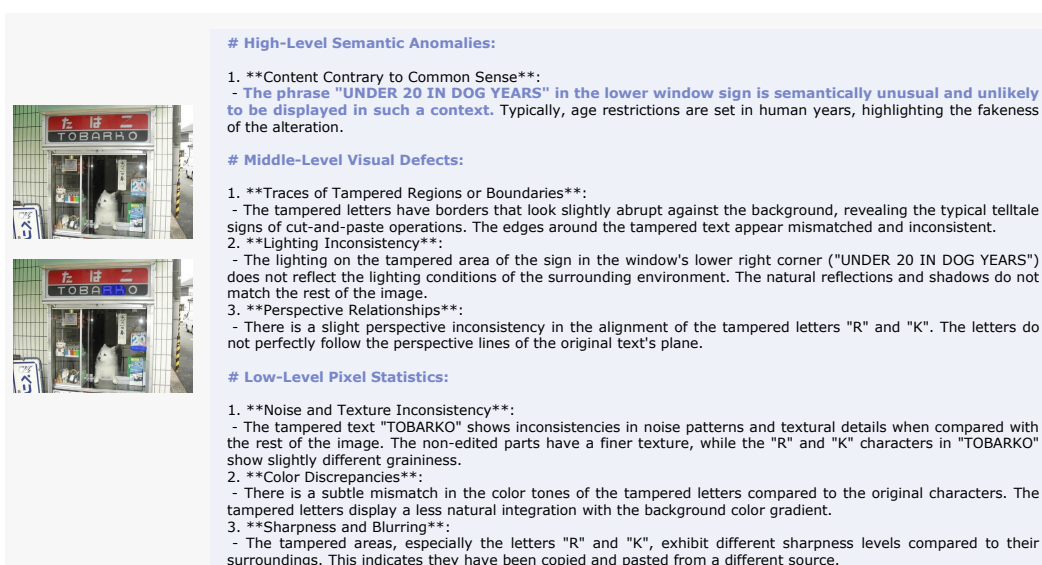
#### # Middle-Level Visual Defects

- Lighting Inconsistency: The overall lighting in the image is diffused and soft, typical of overcast post-disaster conditions. The bus, however, shows inconsistent lighting, with reflective surfaces that suggest stronger light sources that do not exist in the surrounding scene. The shadows on the bus are less pronounced and don't match the diffused light in the image.
- Perspective and Physical Constraints: The bus appears floating or elevated, which defies the physical laws of gravity and common sense. There is no clear indication of support or ground contact for the bus, suggesting it was artificially placed in this environment. Moreover, the angle of the bus does not match the perspective of the background elements.

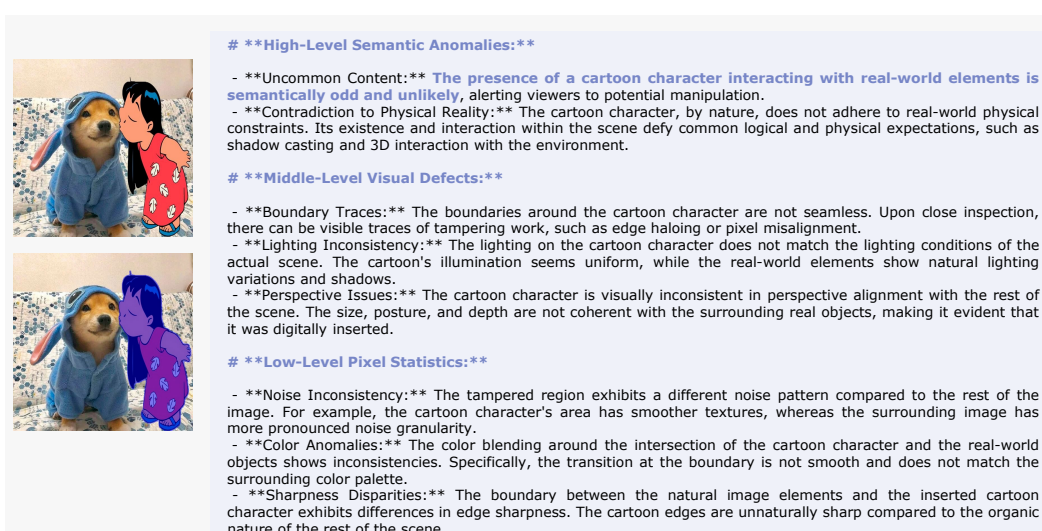
#### # Low-Level Pixel Statistics

- Noise and Texture Discrepancies: The overall scene contains natural graininess, subtle noise, and rough textures. However, the bus in the middle shows different characteristics. It appears smoother and lacks the same level of noise as its surroundings. This noise discrepancy is a common marker of AI-generated content.
- Color Anomalies: The bus has unnaturally vivid colors, which stand out in contrast to the muted tones of the devastated environment.
- Sharpness Anomalies: The sharpness of the bus also does not match the slightly blurred background, implying that it was not part of the original scene. This sharpness inconsistency indicates a different image source or digital generation.

Figure 12: Illustration of the ForgeryAnalysis dataset creation process. The data is initially generated by GPT-4o, then revised by experts to ensure the accuracy of the analysis, with clues organized in the Chain-of-Clues (CoC) format.



CASE 1. Semantically abnormal text.



CASE 2. Presence of elements with different styles.

Figure 13: Examples from the ForgeryAnalysis-Eval dataset. The data is initially generated by GPT-4o and then cross-revised by multiple experts. The manipulation type for these images is "splice".

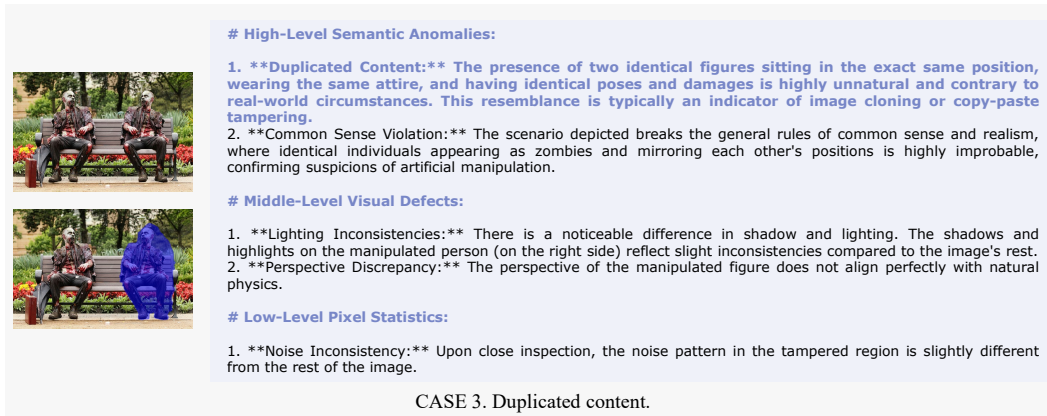


Figure 14: Examples from the ForgeryAnalysis-Eval dataset. The data is initially generated by GPT-4o and then cross-revised by multiple experts. The manipulation type for these images is “copy-move”.

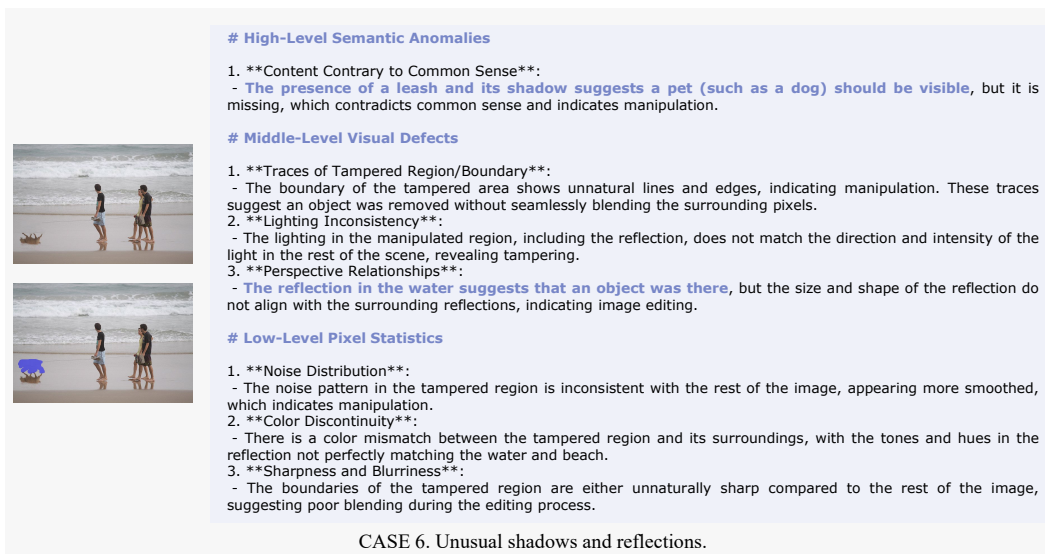
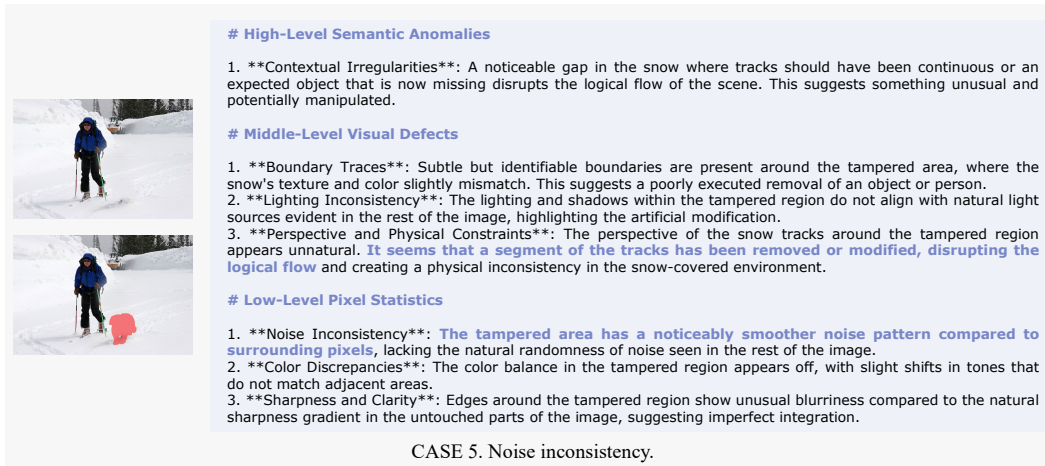


Figure 15: Examples from the ForgeryAnalysis-Eval dataset. The data is initially generated by GPT-4o and then cross-revised by multiple experts. The manipulation type for these images is “remove”.

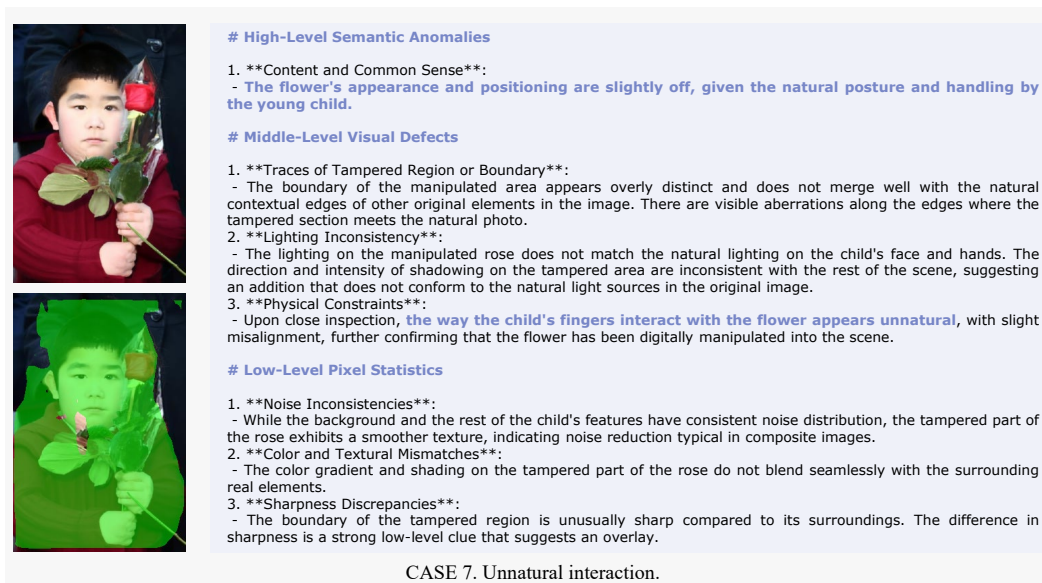


Figure 16: Examples from the ForgeryAnalysis-Eval dataset. The data is initially generated by GPT-4o and then cross-revised by multiple experts. The manipulation type for these images is “AI-generate”.

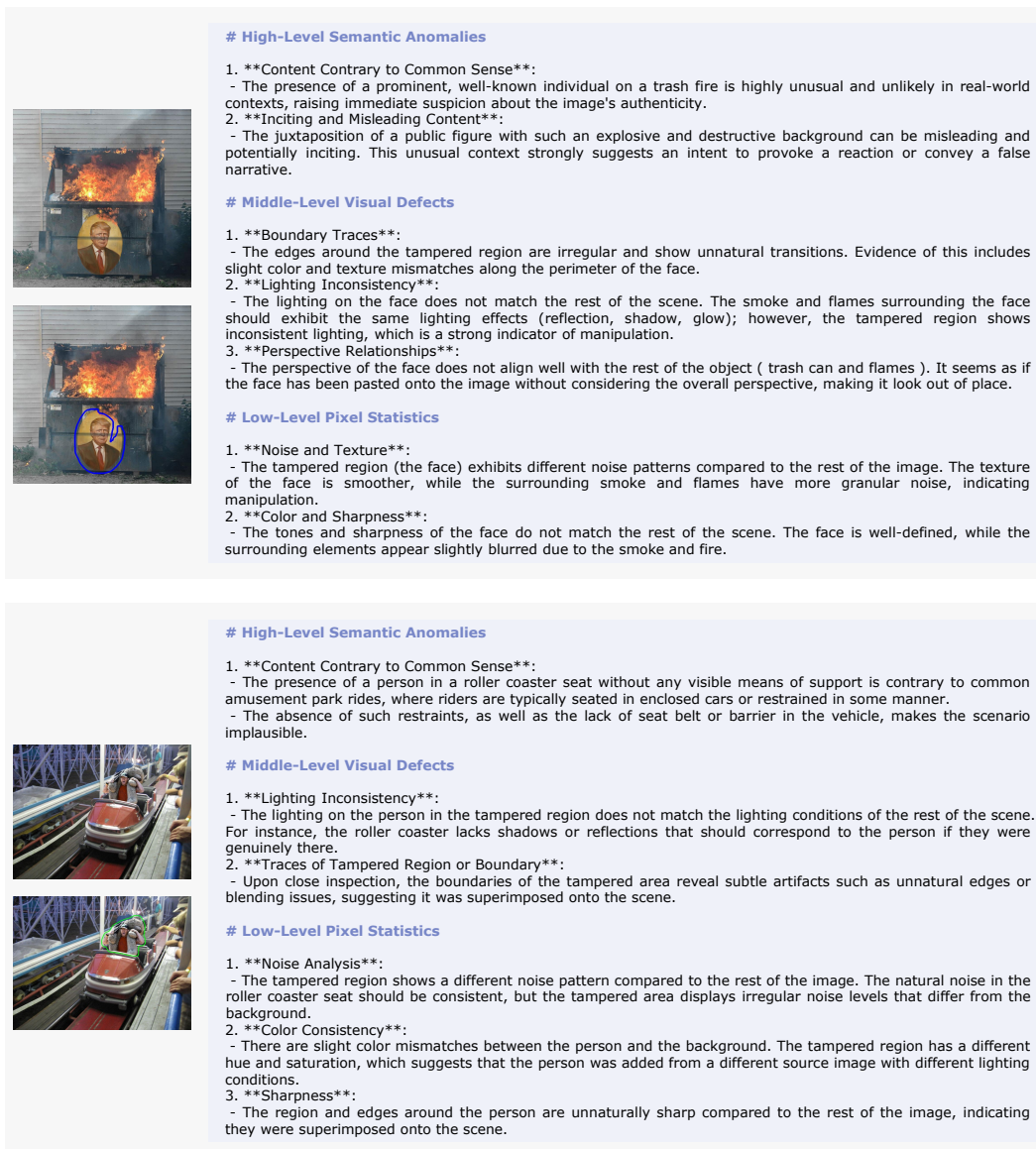


Figure 17: Examples from the ForgeryAnalysis-PT dataset. The data is automatically generated by our data engine ForgeryAnalyst.

Given the tampered image, the reference forgery analysis text, and the generated analysis text, assess the quality of the generated analysis based on the following criteria. Please note! TAMPERED OBJECTS refer to "dog," "face," "grass," etc., while "boundary artifacts" or "lighting inconsistencies" serve as CLUES.



1. Correctness of Tampered Object (NOT Clues) Detection
  - 8-10 points: ALL clues accurately identify major tampered objects and describe alterations but may miss subtle details.
  - 4-7 points: ALL clues reveal the tampered regions but lack clarity or miss some regions.
  - 1-3 points: Score in this range if ANY clues fail to identify tampered objects, or ANY incorrect detection is present.
2. Relevance of Clues to Tampered Object
  - 8-10 points: ALL or MOST clues directly relate to tampered elements, logically supporting the analysis.
  - 4-7 points: SOME clues are loosely related and lack a clear connection to the tampered content.
  - 1-3 points: ALL Clues are irrelevant or misleading, not contributing to the tampering analysis.
3. Detail of Clues and Analysis
  - 9-10 points: Provides comprehensive and well-explained clues with thorough reasoning.
  - 6-8 points: Contains good detail but may lack depth in some aspects.
  - 3-5 points: Limited detail with some clues inadequately explained.
  - 1-2 points: Minimal detail with insufficient reasoning or vague explanations.

Assign separate scores on a scale of 1 to 10 for each criterion, with a higher score reflecting better performance. Provide the scores and concise comments explaining the assessment in the format: "Score1: {score}/10 \n Comment1: {text} \n Score2: {score}/10 \n Comment2: {text} \n Score3: {score}/10 \n Comment3: {text}"

**Reference Analysis Text:** {

We have identified the following clues, where high-level anomalies are doubts, and lower-levels clues are evidence.  
# High-Level Semantic Anomalies: ## Content Contrary to Common Sense: The dog's face on a person's body is impossible and contrary to common sense. This unnatural combination lacks logical coherence in real-world scenarios, confirming manipulation. The content is misleading as it presents an unnatural scenario intended for humor or deception...  
}

**Generated Analysis Text:** {

<FAKE> The image appears to be digitally manipulated. The most obvious clue is the fish being held by the person's hand. Fish do not have hands, so this element seems out of place and artificial. Additionally, the human's face looks like it has been edited onto the fish, which further suggests that the image was altered post-production. There are no visible signs of tampering at the pixel level, but the overall composition and elements present in the image indicate that it has been edited.  
}



Score1: 1/10

Comment1: The generated analysis fails to correctly identify the tampered objects present in the reference analysis. Instead of recognizing the dog's face on the human body, it incorrectly discusses a fish being held, which is not mentioned in the reference text.

Score2: 1/10

Comment2: The clues provided in the generated analysis are irrelevant to the tampered object discussed in the reference analysis. The mention of a fish is misleading and not related to the actual tampered content, which involves a dog's face on a human body.

Score3: 2/10

Comment3: The detail in the generated analysis is minimal and not aligned with the actual tampering described in the reference text. The analysis lacks depth, particularly in explaining the clues, and incorrectly focuses on detecting non-existent elements, providing no useful insight into the actual manipulation.

Figure 18: Illustration of the prompt and GPT-4 response of text analysis quality evaluation. The evaluation prompt includes clear scoring criteria for the assessment aspects, ensuring consistent and fair scoring.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction accurately state our main contributions, the scope of our method, and these are well-supported by the results presented in the paper.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discuss the limitations of our work in the *Appendices* due to page constraints.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper focuses on a novel method and a new dataset for image manipulation detection and does not include theoretical results like theorems or formal proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed descriptions of our method, relevant training configurations, and evaluation protocols. For our novel dataset, we thoroughly describe its construction, composition, and splits. These details are sufficient for reproducing our main experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code, data, and model weights are already open-sourced. To facilitate reproducibility while ensuring reviewer anonymity, we provide access to the code and data as described in our supplementary material. While the pre-trained model weights are also open-sourced, due to their significant size, they are not directly included in the current supplementary package. We are committed to providing public links for resources in the abstract of the camera-ready paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Comprehensive details, including hyperparameters, optimizer types, learning rates, batch sizes, and other critical training and testing specifics, are thoroughly introduced in the *Appendices*.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: To ensure fair comparisons, our experimental setup and evaluation protocols are consistent with established state-of-the-art methods in the field. While comprehensive statistical significance tests are not uniformly standard practice in this research domain, we address potential experimental variability. For experiments involving stochastic elements, we conduct evaluations twice and report the average performance.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: We provide details on the computational resources used for our experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: The research presented in this paper has been conducted with careful consideration of the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: This paper discusses the broader societal impacts of our work. A significant positive impact is the potential to enhance the authenticity of visual content in media, which can help combat the spread of misinformation. However, we also acknowledge potential negative impacts. One key concern is the possibility of inaccurate detections, which could lead to the mislabeling of legitimate content.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We have implemented several safeguards. The model weights are hosted on an authoritative platform. Moreover, access to the data is controlled via an application process, which allows us to carefully review and approve user requests.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: For every asset used in this paper, we have properly cited the corresponding original paper or source. We have also ensured that our use fully complies with the respective licenses and terms of use.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: All assets introduced in this paper are well documented, and this documentation is provided alongside the assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[Yes\]](#)

Justification: The instructions given to participants are provided in the *appendices*.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: The potential risks involved in our user study, such as time expenditure, were disclosed to the participants within the annotation system and user experiment system. The experiments, conducted under limited risk conditions, received approval from our laboratory's review process.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

#### 16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Large Language Models (LLMs) constitute an important component of the core methodology in this research. The relevant aspects of their usage are described in detail within the main body of the paper.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.