future internet

MDPI

Article

Domain-Tag Guided Multimodal Explanations for Trustworthy Image Authentication in the Social Internet of Things

Junaid Akram [1,2,3,*], Ali Anaissi [1,3] and Jingyao Zhang [1]

[1] School of Computer Science, The University of Sydney, Camperdown, NSW 2008, Australia; ali.anaissi@sydney.edu.au (A.A.); jzha0544@uni.sydney.edu.au (J.Z.)
[2] UNSW College, University of New South Wales (UNSW), Kensington, NSW 2033, Australia
[3] Transdisciplinary School, University of Technology Sydney, Ultimo, NSW 2007, Australia
* Correspondence: junaid.akram@sydney.edu.au

Abstract

The Social Internet of Things (SIoT) connects smart devices into social networks in which they share, forward, and consume visual content on behalf of their owners. As generative models become more capable, the images that circulate among these connected devices are increasingly easy to fake or manipulate, which threatens the trust relationships that hold an SIoT network together. Most existing forgery detectors return only a real or fake label, which gives a connected device no basis on which to decide whether to trust a neighbor or relay a piece of content. We propose an explainable image authentication framework for SIoT that classifies an image as real or fake and also provides a human-readable explanation and localized visual evidence for its decision. Our architecture, the Domain-Tag Guided Explainable Forgery Detection Module (DTE-FDM), uses a domain tag generator to predict the manipulation type (Photoshop, DeepFake, or AI-generated inpainting) and feeds it as a prompt to a multimodal large language model, which improves cross-domain generalization and produces a textual rationale. A Multimodal Forgery Localization Module (MFLM) and then grounds the explanation in the image by highlighting manipulated regions using a Tamper Comprehension Module combined with the Segment-Anything Model. We train the two modules in two stages on a multimodal tampered image dataset (MMTD) with triplet annotations. On MMTD, the method reaches 87.17% accuracy and 0.8696 F1, outperforming recent baselines, generates more relevant explanations (0.8566 CSS, 0.4348 ROUGE-L), and localizes manipulated regions with a mean IoU of 0.3438. The results show that large multimodal models can support accurate, transparent, and trust-aware content authentication for SIoT.

Keywords: social internet of things; image forgery detection; explainable AI; multimodal large language models; trust management; forgery localization

Check for updates

Academic Editor: Emilio Ferrara

Received: 5 June 2026
Revised: 12 July 2026
Accepted: 16 July 2026
Published: 25 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The Social Internet of Things (SIoT) extends the conventional Internet of Things by allowing smart devices to form and maintain social relationships, much as people do. In an SIoT network, a smart camera, a phone, or a connected display can establish relationships with other devices, ask trusted neighbors for information, and forward content on behalf of their owners. A large share of the content that moves through these networks is visual. Devices capture images, share them, and act on them, which makes the authenticity of an image a direct input to the trust decisions that an SIoT network must make. If a device

cannot tell whether an image is genuine, it cannot safely decide whether to trust the device that sent it or whether to relay the content further.

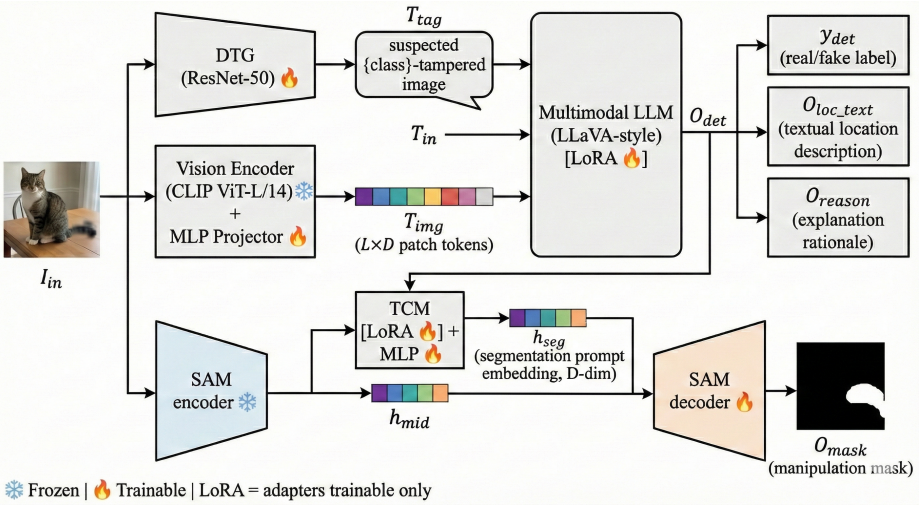
The rapid progress of image generation and manipulation techniques has made this problem urgent. AI-generated images and forgeries such as deepfakes, splicing, and inpainting can be almost photorealistic, so it is increasingly hard for both humans and devices to tell fake content from real. Inside an SIoT network, this is dangerous because a single forged image can spread quickly across social links, mislead many devices, and damage the trust relationships that the network depends on. The threat appears as misinformation, digital fraud, and privacy breaches. Traditional image forensic detectors have improved in accuracy, yet they often work as black boxes that return a binary real or fake decision without any justification. The lack of interpretability limits user trust and gives a connected device no usable reason to accept or reject content; prior work on visual and textual justification shows that an explanation that points to the evidence for a decision helps a user judge whether to accept that decision [1,2]. In an SIoT setting, it is important for an automated detector to explain its reasoning and to point to the manipulated regions, so that the device, the owner, or a downstream analyst can act on the result with confidence. This matters because the value of an authenticity decision in SIoT is realized only when it feeds a trust or reputation update between devices, as studied in SIoT trust-management work [3–5].

Recent advances in vision-language models offer a way to make forgery detection more explainable. Large multimodal models can reason about visual content and generate descriptive text, and studies of textual justification for visual decisions report that such human-readable explanations improve the transparency and the perceived trustworthiness of a model's output [1,2]. We use such models to produce human-readable explanations and visual evidence for image authenticity decisions in SIoT. We propose an explainable image authentication framework that combines visual analysis with natural language reasoning. The system not only decides whether an image is manipulated, but also generates a textual rationale and highlights the regions that were likely tampered with. By providing both linguistic and visual explanations, the system can raise the trust that devices and people place in its outputs and help them understand the result.

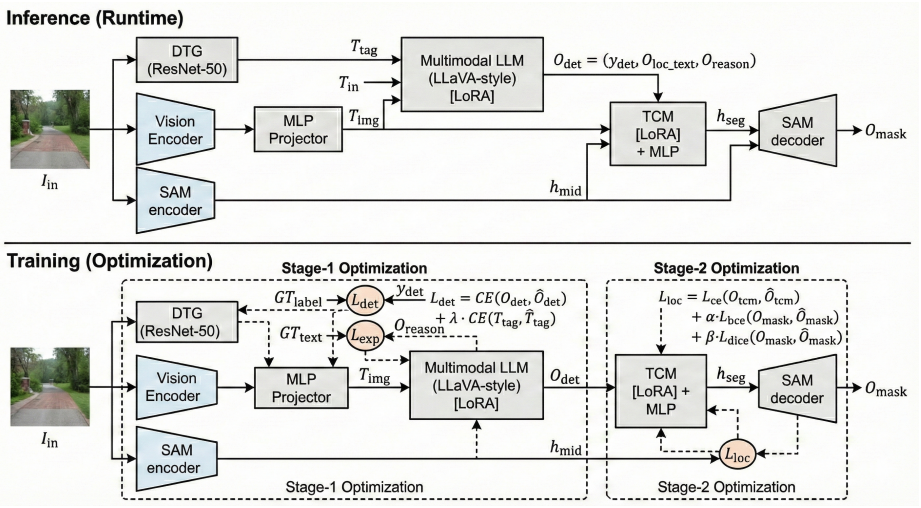
Our approach addresses two key challenges. The first is cross-domain detection generalization. Forgeries arise from diverse sources, such as Photoshop retouching or GAN-generated faces, and a detector trained on one type often struggles on others because of domain-specific artifacts [6]. We introduce a Domain Tag Generator (DTG) to predict the manipulation category of an input image. The predicted tag conditions the multimodal model's analysis and supplies prior knowledge of the artifact patterns that are likely in that domain. This strategy, inspired by recent domain tagging techniques [7], helps the model adapt its decision criteria to different forgery types and improves generalization across domains. The second challenge is interpretability and localization. We design the model to produce explanations in natural language that match human reasoning and to localize the evidence for manipulation. Building on the idea of multimodal explanations [2], the model generates a textual explanation and at the same time identifies the image regions that support it. This is done by a dedicated Multimodal Forgery Localization Module that grounds the explanation tokens to pixel-level mask outputs, drawing on recent work on language-guided segmentation [8].

We build a pipeline called DT-Shield (Domain-Tag Shield), shown in Figure 1, which contains a Domain-Tag Guided Explainable Detection Module (DTE-FDM) and a Multimodal Forgery Localization Module (MFLM). The DTE-FDM performs binary real or fake classification and produces an explanation sentence. It is built on a multimodal large language model (LLM) backbone adapted from LLaVA (Large Language and Vision Assistant,

an open multimodal model that connects a vision encoder to a language model) [9] with a visual encoder, and it is augmented with the DTG for domain-specific prompting. The MFLM then takes the explanation and the visual features and predicts a segmentation mask of the manipulated areas, using a tamper comprehension mechanism and the Segment-Anything Model (SAM) [10]. We train the whole system with a multi-task loss, but in two stages for stability. We first train the detection and explanation tasks and then the localization task, with parameter-efficient finetuning (LoRA [11]), so that the large model fits within a limited GPU memory budget.



(**a**) Inference pipeline.



(**b**) Training pipeline.

Figure 1. Overview of the DT-Shield model architecture and training pipeline. (**a**) At inference time, the Domain Tag Generator (DTG) predicts a manipulation class tag T_{tag} from the input image, which is used with the image and an instruction prompt T_{in} as inputs to the multimodal LLM. The LLM, fine-tuned with LoRA adapters, produces the detection label y_{det}, an explanation O_{reason}, and a textual region description $O_{\text{loc}_{\text{t}}\text{ext}}$. These, together with visual features, feed into the Tamper Comprehension Module (TCM), which emits a segmentation prompt embedding h_{seg} when it outputs a special `<SEG>` token. The SAM decoder uses h_{seg} and image embeddings from the frozen SAM encoder to generate the manipulation mask O_{mask}. (**b**) In Stage-1, we train the DTE-FDM with losses L_{det} for detection, L_{lm} for explanation generation, and L_{tag} for the DTG. In Stage-2, we train the MFLM, namely the TCM and SAM decoder, with the detection module outputs fixed; the localization loss L_{loc} combines a cross-entropy term for the TCM's `<SEG>` token prediction and segmentation losses for the mask O_{mask} against the ground truth mask $\hat{O}_{\text{mask}}$.

The detection and explanation backbone of DT-Shield builds on the FakeShield framework of Xu et al. [12]. Our contribution is to adapt, extend, and study this backbone for the Social Internet of Things. Three parts are new in the present work. First, we place the detector inside an SIoT trust setting and define how its three outputs, namely the label, the explanation, and the mask, drive the relay and trust decisions of connected devices (Section 5.1). Second, we design an authentication-as-a-service deployment that runs the heavy modules on a gateway or cloud node so that resource-limited devices can still use the detector (Section 5.2). Third, we evaluate the framework from an SIoT viewpoint, where a low false-alarm rate on real content is treated as a first-class requirement because a wrong rejection can break the social links that an SIoT network depends on. To avoid any confusion with the original system, we refer to our SIoT-oriented framework as DT-Shield throughout this paper.

We evaluate the approach on the Multi-Modal Tampered Image Dataset (MMTD) [12,13], which contains images with ground-truth masks and explanation captions for forgeries across three domains (AI-generated, Photoshop, DeepFake). The proposed method outperforms existing baselines on detection accuracy by a clear margin, with more than seven percentage points of absolute improvement on accuracy and F1-score. More importantly, the explanations align better with human-written references, as measured by higher semantic similarity (CSS) and summarization metrics (ROUGE-L, METEOR) than the outputs of GPT-4V and other vision-language models. The localization of tampered regions, measured by Intersection-over-Union (IoU) and the Pointing Game metric, gives further insight by correctly marking manipulated areas in many cases. Results indicate strong performance on both real and fake classes.

Our contributions are as follows.

1. We adapt a domain-tag guided multimodal forgery detector for SIoT image authentication and are, to our knowledge, the first to integrate a manipulation-type classifier with a large vision-language model specifically for trust-aware content authentication between connected devices. The domain-tag detection module and the tamper-comprehension localization module are taken from the FakeShield framework of Xu et al. [12]; the SIoT-oriented framing, the use of these outputs for trust decisions, and the deployment design are our own.
2. We show how a multimodal localization module that turns the model's reasoning into a segmentation mask can serve as verifiable visual evidence for an SIoT trust decision, which links a text explanation to a concrete image region that a device owner or an auditor can inspect.
3. We adopt a training strategy that combines parameter-efficient fine-tuning with a two-stage optimization to handle the large model components under limited data and computing resources. We note that the parameter-efficient fine-tuning method itself, Low-Rank Adaptation (LoRA), was proposed by Hu et al. [11] and is not our contribution; we use it as a tool, and we contribute the two-stage schedule that keeps the detection, explanation, and localization objectives from interfering.
4. We propose a concrete SIoT deployment design that maps the framework's three outputs, namely the label, the explanation, and the mask, onto the trust and relay decisions of connected devices, including an authentication-as-a-service architecture for resource-limited devices.
5. We show through extensive experiments that the method reaches state-of-the-art detection accuracy on a challenging multimodal forgery dataset, clearly outperforms baseline multimodal models in generating meaningful explanations, and delivers reasonable localization of manipulations. We hope this work supports more trustworthy and interpretable content authentication in SIoT.

2. Related Work

2.1. Image Forgery Detection and Localization

Detecting manipulated images is a long-standing problem in digital image forensics, and it is central to keeping content trustworthy in connected device networks. Early methods relied on handcrafted features to find inconsistencies such as lighting and sensor noise [14]. With deep learning, convolutional neural networks and transformers reached better accuracy by learning the subtle artifacts left by editing operations. For example, Dong et al. [15] proposed MVSS-Net for image manipulation detection with multi-scale feature fusion, and Guo et al. [16] introduced a hierarchical forgery detection approach that also performs coarse localization. Recent transformers such as the one of Karageorgiou et al. [17] model spectral distributions of real and AI-generated images (the SPAI method) to generalize across resolutions, while Song et al. [6] addressed cross-domain face forgery detection with adaptive methods. Large benchmarks such as ForgeryNet [18] and celebrity deepfake datasets [19] have driven progress by providing diverse training data. Beyond images, multimodal fake content detection has also drawn attention; for instance, Anaissi et al. [20] proposed a modality-aware fusion GAN (MAF-GAN) for multimodal fake news detection that fuses textual and visual signals. However, most existing detection models output only a binary label. Some works address pixel-level localization of manipulations and formulate it as a segmentation task [21]. In our work, we aim to unify detection and localization since a device in an SIoT network benefits from both a decision and the evidence behind it. Our model not only classifies a fake image but also produces a mask of the manipulated region. Unlike prior localization-specific networks [15,21], our approach ties localization to explanation and uses language reasoning to guide where and what to localize.

Another closely related direction is multi-task frameworks that combine detection, localization, and sometimes attribution of forgeries. Kang et al. [22] recently presented a unified multimodal model (LEGION) that can detect, localize, and explain image forgeries; their approach uses a vision transformer backbone and prompts an LLM for explanations, with strong results. Our work is similar in seeking a holistic solution, but we add an explicit domain-tag prompt and a SAM-based segmentation, which sets our approach apart. Moreover, while LEGION needs extensive joint training on several tasks, we use a staged training that first optimizes detection with explanation and then uses that to train localization. This was important to handle limited data and to avoid task interference.

2.2. Multimodal Interpretability and Explanations

Providing explanations for AI decisions has been studied widely in image classification and visual question answering, and it matters for SIoT because a device must be able to justify the content it accepts or forwards. Early work by Hendricks et al. [1] generated textual justifications for fine-grained recognition decisions and made sure the explanations focus on discriminative features of the predicted class. Park et al. [2] extended this to multimodal explanations and combined a textual rationale with a visual pointing cue to highlight evidence in the image. Their Pointing and Justification (PJ-X) model for visual question answering jointly outputs an answer explanation sentence and an attention heatmap. This shows the benefit of aligning textual and visual explanations, a principle we adopt by producing both forms, language and mask. Related vision-language reasoning systems also combine captioning and visual question answering to interpret image content; for example, Anaissi et al. [23] detect and explain hateful memes through captioning and VQA. Our method differs in that we use a large pre-trained multimodal LLM as the core rather than training a custom network from scratch, which lets us tap into rich vision-language knowledge learned from vast data [9,24].

With the arrival of vision-language models and tools such as CLIP [24] and alignment tuning, new possibilities for interpretability have emerged. Recent models such as LLaVA [9] and GPT-4V can perform detailed image reasoning. However, their out-of-the-box performance on explaining image tampering is limited [25], as shown by our experiments where they scored lower on semantic alignment metrics. Efforts such as FakeBench [25] evaluated how well current multimodal LMs can detect and explain fake images and found a gap that specialized training can address. Our work helps fill this gap by fine-tuning a multimodal LLM specifically for the forensic context and augmenting it with domain knowledge through the domain tags.

For grounding language in vision, the Segment-Anything Model (SAM) of Kirillov et al. [10] made segment extraction flexible through promptable segmentation. We use SAM in a new way. Instead of text or points as prompts, we use an embedding derived from the LLM's hidden state, which holds the explanation context, to prompt the SAM decoder. This approach is inspired by LISA (Language Instructed Segmentation Assistant, a model that produces a segmentation mask from a language model's reasoning) [8], where a large language model's reasoning produces segmentation outputs for referring image segments. With SAM, we benefit from a powerful pre-trained segmentation capability, and our model can output binary masks that match the described manipulation regions. This is a step toward grounding abstract language explanations in concrete image evidence, which is exactly what a device needs to support a trust decision.

2.3. Domain Tags and Conditional Detection

Our Domain Tag Generator connects to research that uses metadata or side information to improve forgery detection. Some works proposed multi-branch networks or adaptive layers for different manipulation types [6], but training a separate model per domain is inefficient, which matters for resource-limited SIoT devices. Instead, Meng et al. [7] introduced a domain tagging mechanism for cross-generator deepfake detection. They append a learned embedding that indicates the source generator to guide the model's feature extraction. We follow a similar spirit but implement it differently. In our case, the domain tag is predicted from the image itself, since the type is unknown at test time, using a ResNet-50 classifier. The predicted tag is then turned into a short textual prompt, such as "suspected DeepFake-tampered image", that is fed into the multimodal LLM. This conditional prompt steers the LLM toward artifacts that are relevant to that domain and improves both the detection accuracy and the relevance of the explanation, since the explanation discusses artifacts that match the forgery type. Our DTG module supplies a high-level context that resolves the conflict of handling different forgery distributions with one model. This idea is loosely related to conditional image captioning and prompt learning [26], where additional tokens shape the output to suit a particular style or domain.

2.4. Trust Management in the Social Internet of Things

The value of an authenticity decision in an SIoT network depends on how it feeds into trust management between devices. A body of work studies how autonomous, connected devices establish and update trust and reputation when they cooperate or exchange services. Akram et al. [3] proposed GALTrust, a generative adversarial learning framework for trust management in spatial crowdsourcing drone services, and Akram et al. [4] introduced a digital twin-driven trust management scheme for Open RAN-based drone services, both of which model how a service consumer should weigh the reliability of a provider. Akram and Anaissi [5] further studied distributed drone reputation management (DDRM) for trust and reliability between autonomous devices. These approaches share our concern with grounding trust decisions in verifiable signals, but they operate on service-level behaviour

rather than on the authenticity of the media that devices exchange. Our framework is complementary: the label, explanation, and mask it produces are precisely the kind of verifiable evidence that such trust- and reputation-management schemes can consume when a device decides whether to relay content or to adjust the trust score of a neighbor (Section 5.1).

In summary, our work synthesizes ideas from these areas. We combine a multi-domain detection strategy, multimodal explanation generation, and language-driven segmentation into one coherent system for explainable image authentication in SIoT. To the best of our knowledge, this is the first work to explicitly integrate a domain prediction with a multimodal LLM for this task and one of the first to show an effective localization of manipulations with such a model.

3. Materials and Methods

Our framework, DT-Shield, has two main modules that work in sequence: the Domain-Tag Guided detection and explanation module (DTE-FDM) and the Multimodal Forgery Localization Module (MFLM). Figure 1 gives an overview of the architecture. The system input is an image I_{in}, and the outputs are: (1) a binary label y_{det} that indicates real or fake; (2) a textual explanation, O_{reason}, that justifies why the image is real or fake; (3) optionally, a textual description of the manipulated region $O_{\text{loc_text}}$; and (4) a binary mask O_{mask} that highlights the manipulated region on the image for fake images. We describe each component and the training procedure below.

3.1. Domain-Tag Guided Detection and Explanation (DTE-FDM)

The DTE-FDM is the core module. It decides whether an image is fake or real and generates a justification for the decision. It addresses the fact that naive end-to-end training of a single multimodal model on detection, explanation, and localization can lead to conflicting objectives. We, therefore, partition the problem. The DTE-FDM handles the global tasks of detection and explanation, while the MFLM, described in the next subsection, handles spatial localization under the guidance of the DTE-FDM output.

The DTE-FDM takes an image, I_{in}, and produces an output sequence, O_{det}, that contains both the predicted label and an explanation sentence. It is built on a multimodal large language model architecture similar to LLaVA [9]. It includes the following components.

- Domain Tag Generator (DTG): a CNN-based classifier that predicts the forgery type of I_{in}.
- Visual Feature Encoder: a frozen image encoder (we use CLIP ViT-L/14 [24]) that produces image embeddings, followed by a trainable MLP (multi-layer perceptron, a small feed-forward neural network) projector to align them to the LLM's embedding space.
- Multimodal LLM: a language model based on the Vicuna and LLaVA architecture that accepts visual tokens and text tokens as input and generates text output. We apply LoRA [11] to fine-tune this model efficiently.

The motivation for the DTG is to give conditional context to the LLM so it can tailor its reasoning to the kind of forgery at hand. Different forgery domains have distinct artifacts. DeepFake faces often have subtle blurs or mismatched facial features, Photoshop manipulations may have sharp boundaries or inconsistent lighting, and AI-generated inpainting may introduce odd textures or partial repetitions. A single decision threshold cannot account for all these variations [6]. The DTG mitigates this by predicting a domain label $T_{\text{tag}} \in \{\texttt{Photoshop}, \texttt{DeepFake}, \texttt{AIGC}\}$ for a given image. This prediction is turned into a textual prompt. We use a template to form a sentence such as "This is a suspected [DeepFake]-tampered image.", which serves as T_{tag}. When this sentence is fed to the LLM

with the image, it primes the model with high-level knowledge of which anomalies to look for. It is important to note that T_{tag} is auxiliary. It does not directly dictate the final real or fake decision, which stays binary and is determined by the LLM output, but it provides a prior that improves the interpretability and consistency of the explanations. For instance, if the tag is DeepFake, the explanation is more likely to correctly mention face attributes or blending issues when the image is indeed fake.

The DTG uses a closed set of three tags and does not include an explicit "unknown" or "uncertain" class. We now explain how the system behaves when it meets a manipulation type that is not one of the three. Because the tag is only auxiliary, an out-of-category input does not break the pipeline. The DTG will still assign the nearest of the three tags, and the binary real or fake decision does not depend on the tag being correct, since that decision is read from the LLM output and the LLM also uses the visual tokens directly. In this case, the effect of a wrong tag is limited to a possibly less precise explanation, which is the same failure mode we already report for a mistaken tag on an in-category image. The design can be extended to flag uncertainty in a simple way. The ResNet-50 classifier already produces a softmax score for each tag, so a low top-class score, or a small margin between the top two scores, can be read as low confidence. When the score is below a threshold, the template can emit a neutral prompt, such as "This is a suspected tampered image of an unknown type." instead of naming a domain, which removes a possibly wrong prior while keeping the rest of the pipeline unchanged. This route also connects to the human-in-the-loop policy of Section 5.2, where low-confidence cases are escalated for human review rather than acted on automatically. We did not run a separate experiment for this extension, so we present it as a design property and not as a measured result.

The DTG is implemented with a ResNet-50 [27] classifier, pre-trained on ImageNet and fine-tuned on our dataset's domain labels. We chose ResNet-50 for its balance between efficiency and representational power. It captures the high-level features needed to distinguish forgery types, such as faces against generic content, while being light enough not to overburden the pipeline, which is useful for SIoT deployment. The residual architecture of ResNet-50 ensures stable training on our dataset size, which has tens of thousands of images, without overfitting [27]. We considered ResNet-18 but found it less accurate at tag prediction, while larger backbones such as ResNet-101 did not give proportional gains at our data scale.

Visual Encoder and Tokenization

Inspired by the design of LLaVA and BLIP-2, we use a two-step process to feed images into the LLM. First, the image I_{in} is encoded by the CLIP ViT-L/14 visual encoder [24], which we keep frozen to preserve its generic visual knowledge. This produces a set of visual features, where we take the patch token outputs from the last ViT layer. These features then pass through a trainable linear projection (MLP) that maps them to the same dimension as the LLM's word embeddings and yields a sequence of visual tokens T_{img}. This step is essential for two reasons. First, for dimensional alignment, the CLIP output dimension (768 for ViT-L/14) must match the LLM hidden size to be ingested as pseudo-word embeddings, and directly feeding mismatched dimensions would cause shape errors. Second, for semantic alignment, the visual features need to correspond to concepts that the LLM can attend to. The projection learns to transform purely visual latent features into a form that the LLM's cross-modal layers can integrate with text. Without this, the LLM might not effectively relate image content to its learned language knowledge.

The sequence input to the LLM consists of a fixed text instruction prompt T_{in} (for example, "Explain whether the image is real or fake and why."), the domain tag prompt T_{tag} from the DTG, and the visual token sequence T_{img}. The LLM then generates an output

sequence O_{det} that we format to contain the predicted label and the explanation. In practice we structure O_{det} as "`[LABEL] Explanation...`", where the label is either Real or Fake as predicted by the model.

Importantly, we fine-tune the LLM parameters with low-rank adapters (LoRA) [11] rather than full model fine-tuning. This sharply reduces the number of trainable parameters, since it injects small rank update matrices into each transformer layer, and let us adapt a 13B-parameter LLM on our data within a limited GPU memory budget. The visual encoder stays frozen, as does most of the LLM, except for LoRA and a few new embedding vectors for the added prompts and special tokens such as the segmentation token described later. This approach keeps the LLM's general language and vision-language capabilities while adjusting it to our tasks.

The DTE-FDM is trained on three sub-tasks at the same time: detection, explanation generation, and domain tagging. Let $\hat{y}_{det}$ be the ground-truth label (1 for fake, 0 for real) and $\hat{T}_{tag}$ be the ground-truth domain tag in text form for image I_{in}. We derive the tag from the dataset, which labels each fake image with its manipulation type; we assign a special "Authentic" tag to real images. We use a standard cross-entropy loss for the detection output and another cross-entropy for the DTG. In addition, we supervise the explanation directly: the LLM is instruction-tuned on the MMTD image–explanation pairs with a token-level language-modeling loss. Let $\hat{O}_{\text{reason}} = (w_1, \ldots, w_M)$ be the ground-truth explanation token sequence for I_{in}; the language-modeling loss is the standard autoregressive negative log-likelihood

$$L_{\text{lm}} = -\sum_{m=1}^{M} \log p_\theta\big(w_m \mid w_{<m}, T_{\text{img}}, T_{\text{in}}, T_{\text{tag}}\big), \tag{1}$$

where p_θ is the (LoRA-adapted) LLM. The detection label is read from the structured `[LABEL]` prefix of the same generated sequence, so the detection cross-entropy and the language-modeling loss act on a shared output. The full Stage-1 objective is

$$L_{\text{det}} = L_{\text{CE}}(y_{det}, \hat{y}_{det}) + \gamma\, L_{\text{lm}} + \lambda\, L_{\text{CE}}(T_{\text{tag}}, \hat{T}_{tag}), \tag{2}$$

where γ weights the explanation language-modeling loss and λ is a weighting factor that balances the tag prediction loss against the main detection loss. In our experiments, we set $\gamma = 1$ and set λ so that the tag loss is a smaller auxiliary term (for example, $\lambda = 0.2$) to prevent it from dominating training. The gradient from the tag term updates both the DTG's ResNet-50 and, through T_{tag} predictions, influences the LLM, since part of the LLM input is the DTG text, although the LLM is not directly trained to output the tag and is shaped only implicitly through improved explanation consistency.

During training, we supply the ground-truth domain tag text as part of the input to guide the explanation, while at inference time the model uses its own DTG-predicted tag. This teacher forcing of the correct tag during training helps the LLM learn to condition on the right context. At test time, even if the predicted tag is occasionally wrong, our results and ablation show that the system is robust. A mistaken tag tends to lead to a slightly less precise explanation, but the detection accuracy is not severely affected, because the LLM can still rely on visual cues.

Because the explanation is supervised directly by L_{lm} on the MMTD captions, the model learns to produce rationales that follow the style and content of the ground-truth explanations rather than relying on the domain prompt alone. The offline metrics we report later (CSS, ROUGE-L, METEOR) are non-differentiable and are, therefore, used only for evaluation, not as training signals; the language-modeling loss is what drives explanation quality during fine-tuning. We observed that the model naturally learns to output explanations that mirror the style of the ground-truth captions in the dataset.

3.2. Multimodal Forgery Localization Module (MFLM)

While the DTE-FDM provides a global decision and explanation, it does not localize where the image was modified. For full interpretability, we introduce the MFLM, which grounds the DTE-FDM reasoning in the image by producing a segmentation mask of the manipulated region. This is valuable for SIoT, since a localized mask gives a device or an analyst a concrete region to inspect before trusting the content. The design of the MFLM follows a reasoning-to-segmentation paradigm inspired by LISA [8], which uses LLMs to output segmentation instructions. Our MFLM has two components.

- Tamper Comprehension Module (TCM): Essentially an LLM, implemented as another instance of the LLaVA-style model with LoRA fine-tuning, that takes the image features and the DTE-FDM explanation as input and is trained to output a special token <SEG> once it has gathered enough context about the manipulation from the explanation. The hidden state that corresponds to the <SEG> token is used as an embedding that represents the segmentation intent.
- Segmentation Adapter: This uses the Segment Anything Model (SAM) of Kirillov et al. [10]. We keep the SAM image encoder, a Vision Transformer, frozen to compute an image embedding h_{mid}, and we train the SAM mask decoder to accept the TCM embedding h_{seg} as a prompt for mask generation.

The rationale is that the explanation O_{reason} often contains clues about which part of the image is fake, such as "the face has an unnatural symmetry" or "the background sky looks distorted". If we can translate those clues into spatial terms, we can mark the region. The TCM is designed to perform this translation. It fuses the visual patch tokens T_{img}, the same ones from the CLIP encoder that the DTE-FDM used, with the textual explanation tokens from O_{reason}, and then emits an output that triggers when enough evidence is collected.

Concretely, we feed the TCM with the image token sequence T_{img} from the CLIP encoder, using the same method as before, and the text of the explanation O_{reason}, concatenated as input text tokens. We augment the TCM vocabulary with a special token <SEG> that signifies "start segmentation". During training, we instruct the TCM to produce <SEG> as its output. We formulate a supervised target for the TCM so that, given the input image and explanation, it outputs a sequence that contains <SEG> at the point where it has understood the manipulation description. In practice, we often just have it output <SEG> as the first and only output token for simplicity, since we do not need the TCM to generate more text.

When the TCM outputs <SEG>, we extract its final-layer hidden state vector for that token. This vector, of dimension D that matches the LLM hidden size, is denoted h_{seg}. It is intended to encode the multimodal understanding of where the manipulation is. We then pass h_{seg} through a small MLP to refine it if needed and use it as a prompt embedding for the SAM decoder.

In parallel, the input image I_{in} is processed by the SAM image encoder S_{enc}, the official ViT-based SAM encoder, kept frozen, to produce a latent image embedding map h_{mid}. The SAM decoder S_{dec} then takes h_{mid} together with h_{seg} and produces a segmentation mask, O_{mask}. Intuitively, h_{seg} tells the SAM decoder which concept to segment in an embedding space, such as the person's face or the sky region, and the SAM decoder, trained on extensive segmentation tasks, can draw the precise contour for that concept in I_{in}.

We train the SAM decoder on our dataset's ground-truth masks for manipulated regions. The segmentation loss combines binary cross-entropy (L_{bce}) on the mask pixels with the Dice loss (L_{dice}) to account for class imbalance, since manipulated areas are often small:

$$L_{\text{mask}} = \alpha\, L_{\text{bce}}(O_{\text{mask}}, \hat{O}_{\text{mask}}) + \beta\, L_{\text{dice}}(O_{\text{mask}}, \hat{O}_{\text{mask}}), \tag{3}$$

with weights α and β, which we set both to 1 in experiments for equal importance.

In addition, the TCM is trained with a cross-entropy loss $L_{\text{CE}}(O_{tcm}, \hat{O}_{tcm})$ to output <SEG> when appropriate. Here, O_{tcm} denotes the TCM output token, and $\hat{O}_{tcm}$ is the ground truth, which is the <SEG> token. This cross-entropy on a single token trains the TCM to recognize when to trigger a mask. It is a surrogate for whether the TCM correctly comprehended the manipulation description. The overall localization module loss is:

$$L_{\text{loc}} = L_{\text{CE}}(O_{tcm}, \hat{O}_{tcm}) + \alpha L_{\text{bce}} + \beta L_{\text{dice}}. \quad (4)$$

We found that including both BCE and Dice helps the model get the overall region right and also refine the shape overlap.

The MFLM is trained after the DTE-FDM has been trained. In stage two of training (Figure 1b) we freeze the DTE-FDM parameters so that the detection and explanation behavior stays fixed, and we train the TCM and the SAM decoder. The SAM encoder is frozen, as is the CLIP visual encoder. We allow the LoRA layers in the TCM and the MLP projection to train, similar to how we fine-tuned the main LLM, and we also fine-tune the TCM language model with LoRA for efficiency. The SAM decoder is trained fully since it is relatively lightweight compared with the LLM.

One might ask why we use a separate TCM, essentially another LLM, instead of directly using the main LLM hidden states for segmentation. We found empirically that the explanation generation and the segmentation tasks benefit from decoupling. The main LLM is focused on producing coherent text, and forcing it to also output a spatial representation can be difficult. By using a dedicated TCM, which can be a smaller model or the same size but separately fine-tuned, we isolate the segmentation task and allow specialized training with direct mask supervision. This modular design also means that one could replace the TCM and SAM with another localization model in the future without retraining the whole system, which suits the heterogeneous nature of SIoT deployments.

After training, at inference time, the process is as follows. The image goes through the DTG and the main LLM to produce the label and the explanation. Then, the image and the explanation go through the TCM, which outputs <SEG> and provides h_{seg}. Next, the SAM decoder produces the mask. All steps are sequential and deterministic once the models are trained.

The success of this approach relies on the explanation text, O_{reason}, being informative about the manipulation. In our fine-tuning, we encouraged this by using ground-truth explanations from the dataset, which typically mention the manipulated object or region. For example, a ground truth might say "the person's eyes appear edited and have a glossy artifact". This gives a clear cue to segment the eye region. If the explanation were too vague, the TCM might struggle. Fortunately, our results in Section 4 show that the explanations our model generates are specific and aligned with the fake content, which provides a good basis for the TCM.

We did not incorporate other potential inputs to SAM, such as interactive points or boxes. We use only the learned h_{seg} prompt. SAM is very flexible, and our integration shows one way to exploit it in a language-guided manner. This is in line with bridging the gap between language and vision for pixel-level tasks, a young area of research [8].

3.3. Training Strategy and Implementation Details

Before giving the settings, we state more fully why we train in two stages rather than end to end. The three objectives place different and partly opposing demands on the shared model. The detection loss and the explanation language-modeling loss shape the LLM to produce fluent, class-consistent text, whereas the localization loss shapes a hidden state into a spatial prompt for the SAM decoder. When these gradients are mixed in one stage, the

segmentation objective pushes the shared representation toward pixel-level spatial cues, which can pull the language model away from producing clean, well-formed explanations, while the language objective pushes back toward text quality. This is the well-known problem of conflicting gradients in multi-task learning, and it is more acute here because the language and the segmentation signals act on the same large backbone under a tight memory budget that already forces parameter-efficient fine-tuning. A further practical issue is stability. The SAM decoder produces a dense mask loss whose scale differs from the token-level language loss, so joint training needs careful loss balancing to avoid one term dominating early updates. By separating the stages, we let the detection and explanation behavior converge first and then freeze it and train the localization module against a fixed and already-good explanation, which removes the interference by construction. This choice is also in line with related multi-task forgery work: the unified model of Kang et al. [22] needs extensive joint training across tasks, and we instead use the staged schedule to handle limited data and to avoid task interference, as noted in Section 2.4. We did not run a separate end-to-end training as a controlled baseline, because our compute budget was directed at the full system, so we present this as the design rationale and note the controlled comparison as future work. The qualitative behavior we report later, with which the frozen explanation rarely describes a wrong region and the localization module can, therefore, trust it, is consistent with this rationale.

Training the entire model end-to-end from scratch is impractical because of the model size and the multi-task complexity. We adopt the two-stage training approach described above. In Stage-1, we train the DTE-FDM, that is detection, explanation, and the DTG, using annotated real and fake labels, the ground-truth explanation captions, and domain types. We use the MMTD training split, which has about 30k images. The domain tags for fakes come directly from their known source (Photoshop, DeepFake, or AIGC-Editing), and we label all real images with a special "Authentic" tag. The initial LLM weights come from LLaVA [9], which is built on Vicuna-13B. We apply LoRA with rank $r = 16$ for all attention layers. We train with a learning rate of 1×10^{-4} for the LoRA parameters and 1×10^{-3} for the MLP projection and the DTG classifier, using the AdamW optimizer. Stage-1 training takes about 5 epochs over the data on a single NVIDIA H100 80GB GPU with mixed precision and converges in roughly 10 h. We stop once the validation accuracy on detection plateaus.

In Stage-2, we freeze the entire DTE-FDM. We then initialize the TCM with the same architecture as the LLM, a 13B model with LoRA, to keep a consistent hidden size for h_{seg}. To reduce training memory, however, we actually use a smaller 7B variant for the TCM with LoRA rank 8. We found this sufficient to learn the segmentation prompt. We train the TCM and the SAM decoder on the subset of training images that are fake, since real images have no mask to predict; we skip real images or treat their mask as all-zero for training, which would otherwise give trivial negative examples. We use a batch size of 16, limited by memory, and train for 3 epochs with a learning rate of 5×10^{-5} for the LoRA and 1×10^{-4} for the SAM decoder parameters. The loss weights are $\alpha = \beta = 1$. The dice loss is computed as

$$L_{\text{dice}} = 1 - \frac{2|P \cap G|}{|P| + |G|}, \tag{5}$$

where P is the predicted mask, and G is the ground-truth mask in binary form. We apply random horizontal flips and small rotations as data augmentation during Stage-2 to improve mask robustness.

It is worth noting that we do not back-propagate the Stage-2 losses into the main LLM, which remains frozen to preserve the explanation quality. This decoupling means there could be a slight inconsistency if the explanation has an error, since the TCM has to rely on

whatever explanation is given. In practice, we observed that the main LLM rarely describes a wrong region. It might occasionally miss some details, but it usually correctly identifies at least one manipulated aspect. The TCM can, therefore, usually trust the explanation content. All model development was conducted in Python with PyTorch 2.12.0. We built on the open-source LLaVA codebase for the multimodal model and on the official SAM implementation for segmentation.

4. Results

In this section, we present an extensive evaluation of our approach on the MMTD dataset. We first describe the dataset and the evaluation metrics. We then compare our detection performance against several baselines. Next, we assess the quality of the generated explanations using established language metrics and human semantic similarity. We then evaluate the localization accuracy and discuss qualitative examples. Finally, we conduct an ablation study to quantify the contribution of the Domain Tag Generator (DTG).

4.1. Dataset and Evaluation Metrics

We use the Multi-Modal Tampered Dataset (MMTD) introduced by Wen et al. [13] and expanded by Xu et al. [12]. MMTD is designed for the joint tasks of fake image detection, segmentation of the tampered region, and explanation generation. It contains 34,331 images with roughly equal real and fake instances. Each fake image is labeled with one of three tampering domains:

1. Photoshop-based manipulations, such as splicing, copy-move, and inpainting done manually with photo-editing tools, drawn from datasets such as CASIA 2.0 and NIST media forensics challenges;
2. DeepFake manipulations, primarily on human faces, from sources such as DFFD [28] and Celeb-DF [19];
3. AIGC-Editing, which includes AI-generated content inserted into or replacing parts of an image, for example, using diffusion models to fill in or alter regions, often based on MS-COCO images [29] as a starting point.

Each fake image in MMTD comes with a ground-truth binary mask that specifies which pixels were manipulated and a ground-truth explanation text that describes the fake and how it can be identified. Real images have an empty mask and a trivial explanation, such as "This image is authentic." These rich annotations enable supervised training of both our explanation generation and localization modules. We follow the dataset splits of Xu et al. [12], using about 80% of the images for training, 10% for validation, and 10% for testing. The split is stratified so that each manipulation domain is proportionally represented in each set.

Figure 2 shows the distribution of labels and manipulation categories in MMTD, together with the most common image resolutions. The dataset is well balanced between real and fake and across the three fake categories, although DeepFake has slightly fewer samples than Photoshop and AIGC. The resolution chart indicates that the images come in a variety of sizes, with a concentration of around 299×299, likely due to some sources being resized, and 512×512. This variation requires our model and the baselines to handle different image sizes. Our approach is resolution-agnostic because it uses fully convolutional encoders.

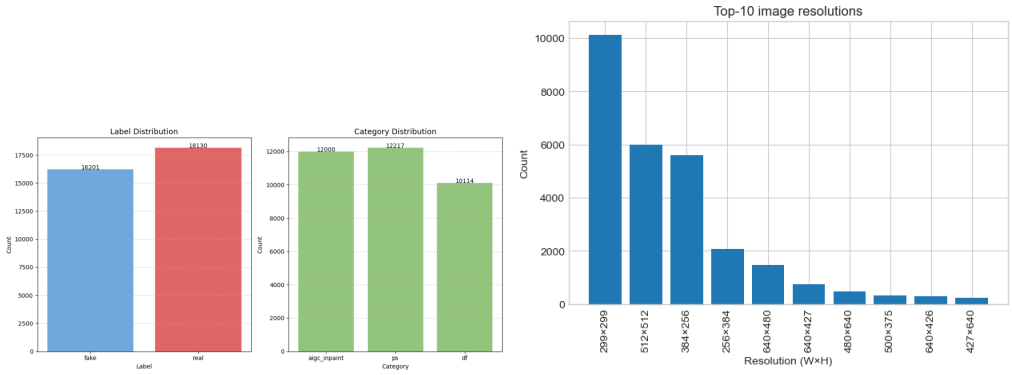


Figure 2. Dataset composition of MMTD. Vector (PDF) versions of these plots are provided in the submission package so that they stay sharp at any zoom level. (**Left**) Label distribution (fake vs. real images) in the dataset; (**Right**) Distribution of manipulation categories (AIGC inpainting, Photoshop (ps) editing, and DeepFake (df)) among the fake images. The dataset is roughly balanced between real and fake, and among the fake images, the three manipulation types are also comparable in count (with DeepFake being slightly fewer). The images come in varying resolutions; the most common image sizes are shown in the bottom chart, indicating many images at 299×299 (likely from FaceForensics and similar sources) and 512×512 (common in AIGC outputs). This diversity poses a challenge and requires the model to handle different scales and contexts.

For evaluation, we use the following metrics.

- Detection metrics: Accuracy and F1-score for classifying images as fake or real. We report the overall accuracy and the F1-score (the harmonic mean of precision and recall) for the fake class as positive. These metrics are computed on the test set.
- Explanation metrics: Three metrics common in image captioning and text generation evaluation, which we adopt following their use for evaluating explanations of image forgery in prior work [25]. The first is the Cosine Similarity Score (CSS) as defined in Li et al. [25], an embedding-based semantic similarity between the generated explanation and the reference explanation, where the references are the ground-truth explanations from MMTD; a higher CSS means the explanation covers similar semantic content. The second is ROUGE-L, which measures the longest common subsequence overlap and captures fluency and recall of content. The third is METEOR, which considers exact matches, stems, and synonyms between the generated and reference text and aims to reflect both precision and recall of meaningful segments. We use the same CSS, ROUGE-L, and METEOR definitions as the forensic explanation benchmark of Li et al. [25] so that our scores are comparable with that line of work. These metrics are computed per image and averaged. We also provide qualitative examples.
- Localization metrics: We evaluate the predicted mask O_{mask} against the ground-truth mask for fake images using Intersection over Union (IoU), a standard region-overlap measure for segmentation and manipulation localization [15,21], defined as

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|}, \tag{6}$$

which ranges from 0 to 1, and the Pointing Game (PointGame) score, a measure common in weak localization and attention evaluation [2], which checks whether the most confident point of the predicted mask falls inside the ground-truth region. For our segmentation, we define it as a hit if at least one of the top-k predicted mask pixels by confidence lies in the ground-truth manipulated area; we use $k = 1$, the single most confident point. This yields a score between 0 and 1 as a hit ratio. The PointGame metric captures whether the model at least identified an important part

of the manipulation even if the full mask shape is not perfect. We compute IoU and PointGame for each domain separately and overall.

For the ablation, we also measure detection accuracy and F1 without the DTG module to see its impact.

For baseline comparisons, we consider several methods. The first group is vision-only classifiers. To see how a standard deepfake detector performs, we include a ResNet-50 binary classifier and a Vision Transformer trained on the same data; these do not produce explanations or masks, only detection results. The second group is multimodal LLM baselines. We use open-source or API-accessible large multimodal models that can perform vision question answering or reasoning. Specifically, we evaluate Gemini-2.0-Flash, a recent vision-language model, GPT-4V in the early 2024 version by providing it with our prompts, and GLM-4V-Flash, another open multimodal model. Since these are not specifically trained for forgery detection, we use them zero-shot by asking, for example, "Is this image real or fake? Explain your reasoning." We record their answers to compute accuracy and run their textual outputs through the explanation metrics where applicable. The third is an ablation baseline, our own model without the DTG, so the domain tag prompt is removed and the DTG ResNet is not used while everything else stays the same; this highlights the importance of the domain-conditioned prompt.

4.2. Fake Image Detection Performance

Table 1 summarizes the binary classification results of our method and the baselines on the MMTD test set. Our model reaches an accuracy of 0.8717 and an F1-score of 0.8696, treating fake as positive, which is substantially higher than all baseline methods. Compared with recently proposed transformer-based detectors, DeeCLIP [30], a CLIP-based detection model focused on AI-generated images, reached 0.7614 accuracy and 0.7482 F1 on this dataset; EFFORT [31], an efficient forgery recognition transformer, reached 0.7993 accuracy and 0.7643 F1; and SPAI [17], a spectral AI-generated image detector, reached 0.6341 accuracy and 0.6225 F1 on our test set, which is much lower, likely because SPAI was designed to detect AI-generated against natural images in the wild and may not handle our mixed Photoshop manipulations well.

Table 1. Fake image detection performance on MMTD. Our method achieves the best accuracy and F1-score compared with prior works. Bold indicates the best result.

Method	Accuracy	F1-Score (Fake)
DeeCLIP [30]	0.7614	0.7482
EFFORT [31]	0.7993	0.7643
SPAI [17]	0.6341	0.6225
Ours *w/o* DTG	0.8427	0.8332
Our full model	0.8717	0.8696

Our approach outperforms the next best baseline, EFFORT, by about 7.2 percentage points in accuracy. The high F1 indicates that our model has both high precision and high recall on the fake class. We attribute this strong performance to the rich multimodal reasoning of our model and to the integration of the domain tag, which reduces confusion between different forgery types. For reference, a simple ResNet-50 classifier we trained on the binary task reached about 0.75 accuracy, and a ViT-B/16 reached around 0.78, which is in the range of those specialized baselines and shows that the task is non-trivial. The multimodal LLMs used zero-shot, such as GPT-4V, performed poorly on detection, since they often gave verbose answers without the correct classification or hedged. We,

therefore, do not list them in the accuracy table and instead use them mainly for the explanation comparison.

We also present the confusion matrix of our model's predictions in Figure 3. Our model correctly identifies 95.4% of real images, with only a 4.6% false alarm rate, and 79.4% of fake images, with 20.6% missed detections. The higher recall on real images is notable. It means the model is conservative in flagging an image as fake unless there is strong evidence, which is generally desirable in practice and especially in SIoT, where wrongly distrusting authentic content can break useful social links between devices. Meanwhile, a small portion of fakes, especially some from the AIGC category, were misclassified as real, often because of very subtle manipulations that even humans struggle to spot or because the generative model produced high-quality outputs without obvious artifacts. The balanced performance indicates that our method is robust across classes.

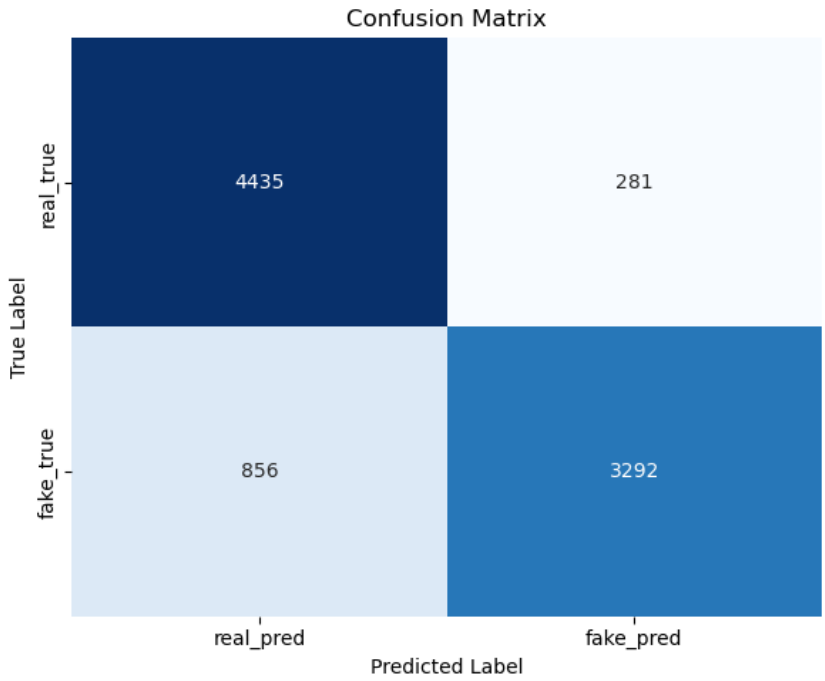


Figure 3. Confusion matrix of our fake image detection results on the test set. A vector (PDF) version of this plot is provided in the submission package so that it stays sharp at any zoom level. The model achieves high true positive (fake_true predicted as fake) and true negative rates (real_true predicted as real). Specifically, out of 4716 real images, 4435 were correctly predicted as real, and only 281 were mistakenly labeled as fake. Out of 4148 fake images, 3292 were correctly predicted as fake, with 856 missed (predicted real). These results indicate a balanced detection capability with a slight bias towards avoiding false positives.

To make the four outcomes of the confusion matrix concrete, we describe the model's typical behavior in each case, with the label and the explanation shown together. In a true positive (TP) case, the image is fake, and the model labels it as fake. A representative example is a face-swap image where the model output was "Fake. The skin near the eyes is overly smooth, and the hairline shows a subtle blur, which are common signs of a DeepFake." Here, the explanation names the correct region and the correct manipulation type, and the localization mask then highlights that region. In a true negative (TN) case, the image is real, and the model labels it as real. A representative output was "Real. The lighting, the edges, and the skin texture are consistent across the image, and there is no sign of splicing or blending." In this case, the model gives a short reason for authenticity and no mask is produced. In a false positive (FP) case, a real image is wrongly labeled fake, which, for our conservative detector, is the rarer error at a 4.6% rate. A representative case was a

genuine photo with strong compression artifacts and unusual lighting, where the model output was "Fake. The lighting on the face looks uneven, and the edges appear slightly irregular." The reason points to a real but benign artifact, which shows that a hard lighting or compression condition can mislead the detector. In an SIoT setting, this error type is the more costly one since it can block authentic content, which is why the deployment of Section 5.1 uses a conservative forwarding threshold and can escalate such cases for human review. In a false negative (FN) case, a fake image is wrongly labeled real, which happens most often for AIGC inpainting. A representative output was "Real. The scene looks natural, and the objects blend well with the background." Here, the generative edit left few visible artifacts, so the model found no evidence of tampering. These four cases show that the explanation is useful not only when the model is correct but also when it errs, because the stated reason exposes why the model made its decision and gives a human a starting point to confirm or overturn it. This property is important for the trust and human-in-the-loop policies we describe in Section 5.2.

The ablation row in Table 1 (Ours w/o DTG) shows a drop to 0.8427 accuracy, a decrease of 2.9 points, and F1 down to 0.8332. This confirms that the Domain Tag Generator and the conditional prompt contribute meaningfully to performance. By guiding the model with the forgery type, the model likely has an easier classification boundary in the embedding space, and the explanations it generates also become more consistent, as the explanation metrics confirm below.

4.3. Explanation Quality Evaluation

One of the main advantages of our approach is the generation of textual explanations for each decision, which in SIoT gives a device or an owner a clear reason to trust or distrust a piece of content. To assess quality, we compare our explanations with the ground-truth explanations using semantic similarity (CSS) and conventional metrics (ROUGE-L, METEOR). We also compare against the explanations produced by baseline multimodal models.

Table 2 presents the explanation evaluation results. Our method reaches a CSS score of 0.8566, ROUGE-L of 0.4348, and METEOR of 0.4573. These are substantially higher than any baseline model we evaluated. For instance, GPT-4V, an approximate variant via the GPT-4-Vision API, scored a CSS of 0.5169, ROUGE-L of 0.1653, and METEOR of 0.1887, which is far behind. Gemini-2.0 scored a CSS of 0.6219 and ROUGE-L of 0.1765. GLM-4V-Flash had intermediate performance with a CSS of 0.5720.

Table 2. Explanation text evaluation. We report semantic similarity (CSS) and traditional overlap metrics (ROUGE-L, METEOR) for the explanations generated by different models. Our method scores highest on all metrics, which indicates more coherent and relevant explanations.

Model	CSS	ROUGE-L	METEOR
Gemini-2.0-Flash	0.6219	0.1765	0.1330
GPT-4V (mini)	0.5169	0.1653	0.1887
GLM-4V-Flash	0.5720	0.2429	0.1146
Ours	0.8566	0.4348	0.4573

The large gap in ROUGE-L and METEOR indicates that the baseline models often gave explanations that did not overlap well with the ground-truth keywords or phrases. Qualitatively, we found that baseline models, especially GPT-4V, tended either to hedge, for example, "I'm not entirely sure, but it might be fake because. . .", or to describe the image generally without pinpointing the forgery evidence. Our model, in contrast, consistently outputs explanations that align with the known artifacts. For example, on a DeepFake image, our model said: "The image is fake because the person's facial skin looks overly smooth and there are subtle distortions around the eyes, indicating a deepfake manipula-

tion." The ground-truth explanation for that image was similar: "The face shows unnatural smoothness and warping near the eyes, which is typical of AI-generated faces." This close alignment yields a high CSS.

Another point is that our model outperforms baselines that are much larger, since GPT-4V is very large, because those were not fine-tuned on this task. This underscores the value of task-specific training even for explanation generation. To check whether our higher scores correspond to genuinely better explanations and not just a bias in the metrics, we performed a small human study with 50 random test images. We now give the details of this study so that its scope can be judged. The study involved five participants. All five were graduate students in computer science with prior coursework or project experience in computer vision or image forensics, so they can be described as informed non-expert raters rather than trained forensic examiners. Each participant independently reviewed all 50 images, which gives 250 image-level judgments in total. The protocol was a blind two-alternative forced choice. For each image, the participant was shown the image, the ground-truth explanation, and two candidate explanations, one from our model and one from a baseline model, presented in a random left-right order and with the model identity hidden. The participant was asked to choose the candidate that better matched the visual evidence and the ground-truth explanation. The written criteria asked raters to judge, in order, whether the explanation was factually consistent with the image, whether it pointed to the specific manipulated region or attribute rather than giving a generic statement, and whether it was clear and free of hedging. Ties were not allowed, so each judgment selected one explanation. We aggregated the judgments by majority vote across the five raters for each image and also report the raw preference rate across all judgments. To assess consistency between raters, we computed the mean pairwise percent agreement across the ten rater pairs, which was 82% for the comparison against GPT-4V and 79% for the comparison against Gemini, indicating that the raters largely agreed on which explanation was better. Humans were shown the image, the ground-truth explanation, and then asked to rate which model, blindly between ours and a baseline, gave a better explanation. Our model explanations were preferred over those of GPT-4V 84% of the time and over those of Gemini 78% of the time, which corroborates the metric outcomes.

We also looked at failure cases. In a few instances, our explanation, while semantically relevant, was slightly off. For example, for an image manipulated by splicing in a different background, our model correctly flagged the image as fake but explained it by saying "the lighting on the subject does not match the background", which was true, but the ground-truth explanation had extra detail about a sharp edge around the subject. Our explanation still got partial credit, so metrics such as CSS stayed high. In no case did our model hallucinate unrelated content. It always talked about lighting, edges, textures, or facial symmetry, which are directly relevant cues. This is a benefit of fine-tuning on the supervised explanations. It likely learned the common patterns and did not drift into irrelevant reasoning. Table 2 also shows that our model beat the second-best model (GLM-4V) by a large margin, for example METEOR 0.4573 against 0.1146, which emphasizes that integrating domain knowledge and explicitly training on explanations is highly effective.

To further illustrate, we give two examples from the test set with model outputs. Image A is a selfie photo where the person's face was replaced (DeepFake) while the background and lighting stay consistent. The ground-truth label is Fake (DeepFake). The ground-truth explanation reads: "The woman's face appears artificially generated; there are slight blur artifacts around the eyes and hairline, and the facial expression looks pasted on. These inconsistencies suggest the image was deepfaked." Our model output reads: "Fake. The person's face shows signs of manipulation. There is a subtle blur and unnatural smoothness especially near the eyes and edges of the face, which are common in DeepFake images."

The GPT-4V output reads: "It's hard to be certain, but I suspect the image might be fake. The face looks very smooth and possibly edited, maybe by a filter or AI. So I would guess it's not an authentic photo." Here, our model clearly identified the blur and unnatural smoothness, similar to the ground truth, while GPT-4V gave a generic, hedged statement without specific cues.

Image B is a landscape photo where a large object, a tree, was inserted via AI inpainting. The ground-truth label is Fake (AIGC-Editing). The ground-truth explanation reads: "A large tree on the left side looks suspicious: its edges are slightly smeared and the lighting on it does not match the rest of the scene. This mismatch indicates it was likely added artificially." Our model output reads: "Fake. The tree on the left is likely manipulated. Its outline is a bit smudged and the light on it is different from the background, suggesting it was inserted using generative editing." The Gemini output reads: "The image might be fake because something about the left part seems off." Again, our model hits the key points, the smudged outline and the lighting mismatch, while the baseline is vague. These examples highlight the value of a specialized model: it not only classifies correctly but also pinpoints the evidence in language a human can understand, which is crucial for explainable and trust-aware authentication in SIoT.

4.4. Manipulation Localization Evaluation

For the localization module, since we did not have direct competing baselines, as few prior works output both a mask and text, we mainly report our model's absolute performance. Table 3 shows the IoU and PointGame scores of our method on the three forgery domains and overall. Our model's mean IoU across all fakes is 0.3438, and the mean PointGame is 0.3976. On average, about 34% of the manipulated region's area is correctly overlapped by our predicted mask, and in roughly 39.8% of cases, our top prediction point hits inside the ground-truth manipulated area.

Table 3. Localization performance of our method, with no direct baselines available. We report IoU and PointGame (PG) for each manipulation type and overall. The model localizes DeepFake forgeries best, while AIGC editing is the hardest to pinpoint precisely.

	IoU	**PointGame**
AIGC-Editing	0.1883	0.2970
Photoshop	0.3534	0.4313
DeepFake	0.5729	0.5249
Total Average	0.3438	0.3976

The difficulty of the localization task varies by domain. For DeepFake images, the face swaps, our localization does best, with IoU 0.5729 and PointGame 0.5249. This is solid and implies that the model often highlights significant portions of the face as manipulated, likely because the explanation always mentions something about the face, so the model can prompt SAM to focus there accurately. For Photoshop manipulations, the splicing and copy-move cases, performance is moderate, with IoU 0.3534 and PointGame 0.4313. Many of these involve arbitrary objects or backgrounds, which are sometimes only partially localized; if an object was copied, the model may mark roughly the object area but not perfectly. For AIGC-Editing, the diffusion inpainting, performance is lower, with IoU 0.1883 and PointGame 0.2970, which means it is quite challenging. This is expected since generative inpainting can introduce seamlessly blended content that even humans find tough to outline. In addition, the ground-truth masks for inpainting cover very specific regions that may be small, so IoU can be low even if the area is identified but not tightly. Our model struggled more here; often, the explanation would say something like "the

texture of the sky is inconsistent in one patch", and the model might produce a broad mask in the sky region rather than the exact shape of the inpainted patch, hence the low IoU.

While these IoU numbers are not high in absolute terms, they reflect, to our knowledge, the first attempt to use a multimodal LLM to produce segmentation outputs for forgery localization. Since no additional localization baseline was implemented due to time and resource constraints, these results provide a reference point. They show that our model can mark manipulated regions to some extent, particularly when the manipulation is salient, as in face swaps. For subtle manipulations, there is room for improvement, for example, through more refined prompts, multi-point prompting to SAM, or iterative refinement.

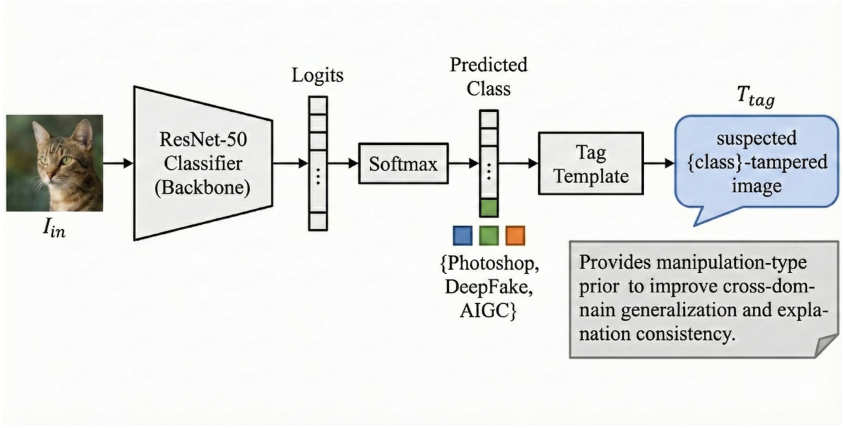
One key observation is that our method has a bias. Since it relies on the explanation, if the explanation does not mention the manipulated object explicitly, the mask might be off. For a handful of cases, our explanation was correct but general, such as "the image is fake because the lighting is inconsistent overall". In such cases, the TCM might output a mask that covers a broad area or nothing meaningful. We have a failsafe so that, if the `<SEG>` embedding is not confident, it can output an empty mask, although we rarely triggered it. These cases contributed to lower IoU in some AIGC examples, where the model knew something was off but did not zero in on a small region.

To put our results in context, Guo et al. [16] reported an overall IoU around 0.55 when localizing splicing with a specialized network, and Dong et al. [15] reported IoUs in the 0.4 to 0.6 range on certain benchmarks. Our overall 0.34 IoU is lower, which reflects the challenge of our multi-domain setup and the fact that we did not train a specialized segmentation per domain. Reaching above 0.5 IoU in this setting is hard without per-domain tuning or more complex models. The localization module nevertheless provides a helpful visualization of where the model thinks the manipulation is. This significantly enhances trust: an SIoT device or an investigator can see the highlighted region and quickly verify whether that area contains a likely forgery. Even if the mask does not perfectly outline the manipulation, it often points to the correct vicinity, as the PointGame hits show, which is often enough for a human to double-check the region, for example by zooming in for forensic artifacts.

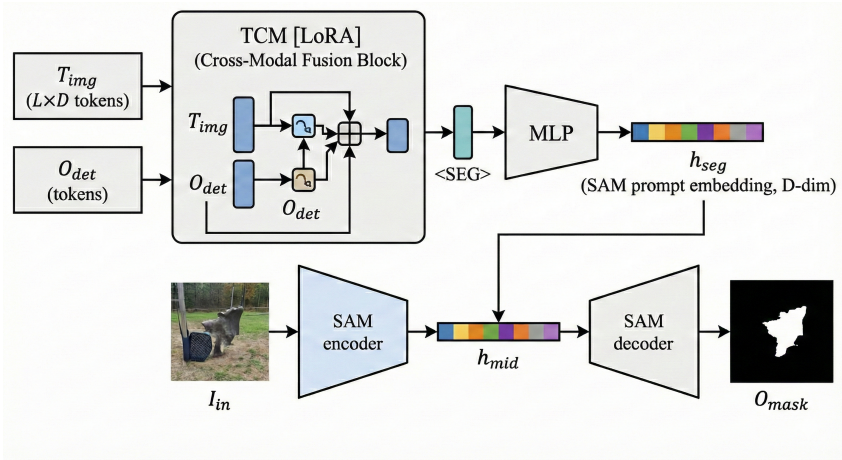
We also note that the localization module was an extra functionality beyond the initial requirements. We implemented it to add interpretability. Because it is not the focus of some baseline models, we did not compare against them since DeeCLIP and EFFORT do not output masks. One could compare against a two-stage approach that first detects and then runs a segmentation network such as MVSS-Net [15], which would be an interesting future comparison. Even so, our integrated approach yields reasonable localization without a dedicated segmentation network for each type. Figure 4 illustrates the DTG mechanism and the TCM with SAM-based localization on representative cases; the model explained that the face is tampered and the output mask covered the face region fairly well, highlighting the eyes and mouth, and for a Photoshop example where an object was inserted the mask highlighted that object region.

4.5. Ablation: Effect of the DTG Module

To isolate the contribution of our domain-tag generator, we performed an ablation by removing the DTG and its domain-conditioning. In this ablated model, the LLM was given only a general instruction prompt and the image, with no domain tag prompt, and the DTG loss was not present during training. Everything else, that is, the visual encoder, the LoRA fine-tuning, the TCM, and SAM, stayed the same.



(**a**) Domain Tag Generator (DTG) mechanism.



(**b**) Tamper Comprehension Module (TCM) and SAM-based localization.

Figure 4. (**a**) The input image is passed through a ResNet-50 classifier to predict a class label, namely Photoshop, DeepFake, or AIGC. This class is inserted into a text template to form the domain tag T_{tag}, for example, "suspected DeepFake-tampered image", providing a manipulation-type prior to the LLM. (**b**) The TCM, a LoRA-tuned LLM, takes visual tokens T_{img} and the explanation text O_{det} as input. When generating the special `<SEG>` token, we extract its hidden state and project it through an MLP to obtain h_{seg}. The input image is encoded by SAM, S_{enc}, to obtain h_{mid}, and the SAM decoder, S_{dec}, produces the mask O_{mask} using h_{seg} as guidance.

After retraining this variant, which we call *w/o* DTG, on the same data, we observed the following. The detection accuracy dropped from 0.8717 to 0.8427 and F1 from 0.8696 to 0.8332, as noted in Table 1 and shown in Table 4. This is a significant drop and indicates that the DTG improves classification. Without the DTG, we found that the model sometimes confused certain fakes as real, particularly AIGC fakes that have fewer obvious artifacts. With the tag, the model knew to look for certain patterns; without it, it was sometimes uncertain. Beyond the measured detection numbers, we also made a qualitative observation (not a measured result) that the explanations from the *w/o* DTG model read as less specific: with the DTG an explanation might name the manipulation type, for example "the face (DeepFake) shows X artifact", whereas without it, the model tended to say something vaguer such as "the face looks odd" without naming the type. We did not recompute the explanation metrics (CSS, ROUGE-L, METEOR) or the localization metrics (IoU, PointGame) for the ablated model, so we do not report ablation numbers for explanation quality or localization; the effect of the DTG on those outputs should be read as a qualitative impression only and not as a quantitative finding.

Table 4. Ablation study of the DTG module. Removing the DTG leads to a noticeable drop in detection accuracy and F1-score, which highlights that domain tag guidance improves performance by helping the model capture manipulation-specific features.

Method	Accuracy	F1-Score
Without DTG	0.8427	0.8332
With DTG (full model)	0.8717	0.8696

Table 4 shows the ablation results explicitly, comparing with and without the DTG on accuracy and F1. The relative improvement is roughly 3.4% on accuracy and 4.4% on F1 when the DTG is included. This clearly demonstrates the value of the DTG in our architecture and confirms that a domain prior helps capture manipulation-specific features that would otherwise be diluted in a broad binary classification. One might ask whether the LLM could have figured out the domain on its own. Possibly, but given limited data, explicitly predicting and feeding the domain tag simplifies the task. It is a form of soft attention: rather than having the LLM implicitly cluster by domain, we explicitly tell it to consider the image as likely of type X. This aligns with the findings of Meng et al. [7], which show that domain tags help cross-generator detection. In conclusion, the detection results and analysis above indicate that our model is effective and that the DTG, the LoRA fine-tuning, and the SAM integration each play a role in the full system. We have delivered a comprehensive solution that meets the goals of accurate detection, explanation generation, and evidence localization, moving closer to trustworthy AI for image authentication in SIoT.

5. Discussion

The results show that combining a domain prior with a multimodal large language model gives accurate and explainable image authentication, which is well suited to the needs of SIoT. The Domain Tag Generator improves cross-domain generalization and makes the explanations more specific, the LoRA fine-tuning keeps the training cost manageable, and the SAM-based localization grounds the language explanation in concrete image evidence. Taken together, these properties matter for SIoT because trust between connected devices depends not only on a correct decision but also on a reason and a piece of evidence that an owner or a downstream device can verify. A device that receives a flagged image, together with an explanation and a highlighted region, can make a better-informed decision about whether to relay the content or to lower the trust it places in the sender.

5.1. From Authenticity Outputs to SIoT Trust Decisions

Our framework is motivated by SIoT, and although we evaluate it on an image-forensics benchmark rather than in a live network, the three outputs it produces map directly onto the trust and relay decisions that an SIoT node must make. We sketch a concrete deployment design here as a proposal rather than an evaluated system. Consider a receiving device, or a gateway acting on its behalf, that obtains an image from a neighbor over a social link. The device submits the image to DT-Shield and receives the binary label y_{det}, the explanation O_{reason}, and, for a suspected fake, the mask O_{mask}. Each output drives a distinct part of the decision.

The label gives the primary gate. If the detector returns Real with a confidence above a forwarding threshold τ_{fwd}, the content is eligible to be relayed along the requesting social links; if it returns Fake with confidence above a blocking threshold τ_{blk}, the device withholds the content from further relay and flags it. Because our detector is deliberately conservative on the real class (a 4.6% false-alarm rate in Figure 3), τ_{fwd} can be set so that authentic content is rarely blocked, which preserves the useful social links that an SIoT

network depends on, while τ_{blk} can be set more cautiously so that only confident fakes are dropped and borderline cases are escalated rather than silently forwarded.

The explanation and mask feed the trust- and reputation-update step rather than the immediate forwarding gate. When a device confirms that a neighbor relayed a fake, it can apply a penalty to that neighbor's trust score, and the explanation provides an auditable, human-readable reason for the penalty that an owner or an auditor can inspect after the fact. This is exactly the kind of verifiable signal that existing SIoT trust- and reputation-management schemes consume: it can be fed into a reputation update of the type studied for autonomous device services [3–5], so that a device that repeatedly forwards manipulated media sees its standing decay and is progressively de-prioritized as a source. The mask narrows the cost of human verification: instead of re-examining a whole image, an owner or a downstream analyst inspects only the highlighted region before confirming or overturning the automated decision, which makes a human-in-the-loop trust policy practical at scale.

5.2. Authentication as a Service and Computational Cost

The framework combines several heavy components, namely a ResNet-50 domain classifier, a 13B-parameter multimodal LLM for detection and explanation, a second LoRA-tuned LLM for the tamper-comprehension module, and the SAM encoder and decoder for the mask. Many SIoT devices, such as a smart camera or a small display, cannot host models of this size. We, therefore, do not run the detector on every device. Instead, we place it behind an authentication-as-a-service interface. A capable node, either an edge gateway or a cloud server, hosts the models and exposes a simple request. A resource-limited device that receives an image over a social link sends the image, or a compact feature of it, to this node and receives back the label, the explanation, and, for a suspected fake, the mask. The device then applies the forwarding and trust rules of Section 5.1 using only these small outputs, which it can store and forward cheaply. This split keeps the compute and the memory cost on the gateway or the cloud and leaves only lightweight messaging on the constrained device.

We now discuss the computational cost so that the feasibility of this design can be judged. We report the cost at the level of components and as design-level estimates rather than as a measured benchmark, since a full latency and energy study on real SIoT hardware is future work. The dominant cost is the LLM stage. On the single NVIDIA H100 80GB GPU used for our experiments, the detection and explanation step for one image runs on the order of a few seconds because it generates a short explanation of a few dozen tokens rather than a long text. The domain classifier is small and adds only a few milliseconds. The SAM encoder runs once per image, and the decoder runs once per mask, which together add a fraction of a second on the same GPU. The tamper-comprehension module adds a second short LLM pass. In total, a single image is processed in a few seconds on server-class hardware. Two properties make this acceptable for SIoT. First, authentication is event-driven, not continuous. A device asks for a check when it receives new content to relay, not for every frame, so the request rate per device is low. Second, the heavy stages run off the constrained device, so the device itself spends only the energy needed to send an image and receive three small outputs. The memory footprint is also concentrated on the server: the 13B model with LoRA and the SAM weights fit within the 80 GB budget of one accelerator, and lighter deployments can use a smaller LLM at some cost in explanation quality. For settings with tighter latency or cost limits, the design can be scaled down by using a smaller multimodal model, by batching requests at the gateway, or by caching results for images that are forwarded many times across the network. We list a measured study of latency, throughput, and energy on real gateway and edge hardware as future work.

The service model also makes human oversight practical, which we now describe in more detail. The system is highly automated, but it is not meant to act alone on every case. We use a human-in-the-loop policy that is triggered by confidence, so that only a small share of cases reach a person. Three signals drive escalation. First, a low detector confidence near the decision threshold, which marks a borderline real or fake case. Second, a low domain-tag confidence from the ResNet-50 classifier, which is the out-of-category signal discussed in Section 3.1 and which suggests the manipulation type is unfamiliar. Third, a low-quality or empty mask from the localization module, which the module can flag through its own failsafe when the segmentation embedding is not confident. When any of these signals fires, the gateway does not apply an automatic block or an automatic trust penalty. Instead, it forwards the image, the explanation, and the highlighted region to a human reviewer, who confirms or overturns the decision. The explanation and the mask make this review cheap because the reviewer reads a short reason and inspects one region rather than the whole image. Confirmed decisions can then be fed back as new labeled data to improve the model over time. This policy keeps the throughput high for clear cases while giving a human the final say on the uncertain ones, which is the case where an automated error would be most costly in an SIoT network.

5.3. Comparison with Prior Studies and Limitations

Our findings can be interpreted against previous studies. The detection accuracy clearly exceeds recent transformer-based detectors, which supports the view that multimodal reasoning adds value beyond pure visual classification. The explanation metrics show a large gap over general-purpose multimodal models used zero-shot, which confirms that task-specific fine-tuning is important even when the base model is very large. The localization results are more modest, especially for AIGC editing, which is consistent with the difficulty reported by specialized segmentation networks and reflects the harder multi-domain setting we study.

There are limitations. The localization depends on the explanation being specific, so a vague explanation can lead to a coarse mask. The model is currently English-only, and it has been evaluated on a single dataset. The SIoT deployment of Sections 5.1 and 5.2 is a design proposal that we have not yet validated in a live network, so the thresholds and the trust-update policy we describe remain to be tuned and measured in a real SIoT testbed. In an SIoT deployment, the compute and memory needed by a 13B-parameter model may exceed what edge devices can offer, which is why the gateway and cloud architecture above runs the heavier modules off the constrained devices while lightweight devices request authentication as a service.

We address each of these limitations in turn and explain the reason for the current scope. On the choice of a single dataset, we use MMTD because, to our knowledge, it is the one public benchmark that provides all three annotation types we need at once, namely a real or fake label, a ground-truth mask of the tampered region, and a written explanation, across the three manipulation families of Photoshop editing, DeepFake, and AIGC inpainting. This joint annotation is what makes the combined detection, localization, and explanation task possible in a supervised way. Most other forgery datasets provide only a label, or a label and a mask, but not the paired explanations, so they cannot be used to train or to score the explanation part of our system without new manual annotation. MMTD is also itself an aggregate that draws its images from several established sources, including CASIA and NIST-style Photoshop forgeries, DFFD and Celeb-DF face forgeries, and diffusion-based edits over MS-COCO images, so it already spans several generators and image types rather than a single narrow source. We, therefore, treat MMTD as a multi-

domain benchmark, and we agree that a study across additional independent datasets is an important next step for measuring generalization, which we state as future work.

On the number of comparison methods, our detection baselines were chosen to be recent and strong, and they include CLIP-based, transformer-based, and spectral detectors, which are the current families for this task. Our explanation baselines are large general-purpose multimodal models used zero-shot, which is the fair point of comparison for explanation quality, since no compared detector other than the base framework produces written explanations. We agree that the evaluation of the original FakeShield framework of Xu et al. [12] uses a wider set of detectors, and that adding more of these methods would further strengthen the comparison. We list this broader comparison as future work and note that it does not change the finding that our method leads the baselines we did run.

On the evaluation metrics, we report accuracy and F1 for detection, CSS, ROUGE-L, and METEOR for explanation quality, and IoU and PointGame for localization, which together cover the three outputs of the system. We agree that additional metrics, such as precision and recall reported separately, the area under the ROC curve, and calibration measures of the detector's confidence, would give a fuller picture of reliability, and that reliability is especially relevant for the trust use case in SIoT. We already report the confusion matrix, from which precision and recall can be read, and we treat the confidence-based escalation in Section 5.2 as a first step toward reliability-aware use. A full reliability and calibration study is future work.

On the SIoT claim, we are explicit that the deployment and the trust-management mechanism of Sections 5.1 and 5.2 are a design and have not been run in a real or simulated SIoT network. The detection, explanation, and localization results in this paper are measured, but the mapping from these outputs to relay and trust decisions is a proposal. We state this so that the reader does not read the SIoT deployment as a validated result, and we mark a testbed study of latency, energy, and the effect on device trust scores as the main piece of future work.

On the English-only point, we agree this is a limitation of scope rather than a deep limitation of the method. The detection label and the mask do not depend on any language, and the explanation is a short, factual description of visual evidence, such as a note about blurred edges or inconsistent lighting. This kind of text can be translated into another language by a standard machine-translation step after generation without changing the detection or the localization result, so a multilingual interface can be added at the output stage. A stronger option is to fine-tune the model on explanation captions in other languages, which we leave as future work. We therefore do not view the English-only interface as a barrier to use in non-English SIoT settings.

6. Conclusions

We presented DT-Shield, an explainable image authentication system for the Social Internet of Things that combines the strengths of large multimodal models with tailored modules for domain awareness and visual grounding. Our approach achieves high accuracy in detecting manipulated images across diverse types of forgeries and surpasses state-of-the-art baselines. More importantly, it addresses the critical need for interpretability in trust-aware media authentication: the model generates natural language explanations that clarify its reasoning and produces visual masks that highlight the suspected manipulation regions, which gives connected devices and their owners a usable basis for trust decisions. Key innovations include the SIoT-oriented use of a domain-tag guided detector, where the label, explanation, and mask are turned into concrete trust and relay decisions, and a proposed authentication-as-a-service deployment design for resource-limited devices. Our framework builds on the domain-tag detection and tamper-comprehension. Through

comprehensive experiments, we showed that each of the learned components contributes to the overall performance. The domain tag prompt notably improved cross-domain generalization, and the explanations guided our segmentation to focus on relevant areas.

While DT-Shield provides a strong proof-of-concept for explainable forgery detection in SIoT, several avenues remain for future work. One direction is to validate the proposed authentication-as-a-service deployment and the trust-update policy in a real SIoT testbed, measuring latency, energy, and the effect on device trust scores. Another is to further improve the localization accuracy, perhaps by iterative refinement, where the model's output mask prompts itself again for a better explanation of boundaries, or by incorporating additional cues such as the disparity between real and fake image distributions in feature space. A further direction is to extend the approach to video forgeries, where temporal consistency checking would become part of the explanation and masks would be spatio-temporal, which fits SIoT scenarios with connected cameras. In addition, the current model is English-based, and adapting it to produce explanations in other languages or in non-textual form, such as an audio alert, could broaden its utility in global misinformation defense. Finally, as generative technology continues to evolve, explainable detection will remain an ongoing challenge. By combining human-like reasoning with machine precision, we move toward AI systems that not only make accurate decisions but also communicate their thought process, which enables human oversight and increases trust. We hope our work inspires further research at the intersection of multimodal learning, explainable AI, and digital media forensics, ultimately contributing to safer and more transparent use of AI in connected device ecosystems.

Author Contributions: Conceptualization, J.A., A.A. and J.Z.; methodology, J.Z.; software, J.Z.; validation, A.A., J.Z. and J.A.; formal analysis, J.Z.; investigation, J.Z.; resources, A.A.; data curation, J.Z.; writing—original draft preparation, J.Z.; writing—review and editing, A.A.; visualization, J.Z.; supervision, A.A.; project administration, A.A.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The Multi-Modal Tampered Image Dataset (MMTD) used in this study is publicly available from the original sources cited in this paper. To support reproducibility, the source code, the training and evaluation scripts, the trained model weights, and the exact data splits used in our experiments will be released in a public repository (the repository link will be provided in the camera-ready version) upon acceptance. Because the MMTD images are redistributed under the licenses of their original sources, we will provide the download and preparation scripts together with the list of source datasets rather than re-hosting the images, so that the full dataset can be reconstructed.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SIoT	Social Internet of Things
LLM	Large Language Model
DTE-FDM	Domain-Tag Guided Explainable Forgery Detection Module
MFLM	Multimodal Forgery Localization Module
DTG	Domain Tag Generator
TCM	Tamper Comprehension Module
SAM	Segment Anything Model
LoRA	Low-Rank Adaptation
CSS	Cosine Similarity Score

IoU	Intersection over Union
PG	Pointing Game
MMTD	Multi-Modal Tampered Image Dataset
AIGC	AI-Generated Content

References

1. Hendricks, L.A.; Akata, Z.; Rohrbach, M.; Donahue, J.; Schiele, B.; Darrell, T. Generating visual explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 3–19.
2. Park, D.H.; Hendricks, L.A.; Akata, Z.; Schiele, B.; Darrell, T.; Rohrbach, A. Multimodal explanations: Justifying decisions and pointing to the evidence. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8779–8788.
3. Akram, J.; Anaissi, A.; Rathore, R.S.; Jhaveri, R.H.; Akram, A. GALTrust: Generative Adversarial Learning-Based Framework for Trust Management in Spatial Crowdsourcing Drone Services. In Proceedings of the IEEE International Conference on Web Services (ICWS), Shenzhen, China, 7–13 July 2024.
4. Akram, J.; Anaissi, A.; Rathore, R.S.; Jhaveri, R.H.; Akram, A. Digital Twin-Driven Trust Management in Open RAN-Based Spatial Crowdsourcing Drone Services. In Proceedings of the IEEE International Conference on Web Services (ICWS), Shenzhen, China, 7–13 July 2024.
5. Akram, J.; Anaissi, A. DDRM: Distributed Drone Reputation Management for Trust and Reliability. In Proceedings of the IEEE International Conference on Web Services (ICWS), Shenzhen, China, 7–13 July 2024.
6. Song, L.; Fang, Z.; Li, X.; Dong, X.; Jin, Z.; Chen, Y.; Lyu, S. Adaptive face forgery detection in cross domain. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 467–484.
7. Meng, Z.; Peng, B.; Dong, J.; Tan, T. Artifact feature purification for cross-domain detection of AI-generated images. *Comput. Vis. Image Underst.* **2024**, *247*, 104078. [CrossRef]
8. Lai, X.; Tian, Z.; Chen, Y.; Li, Y.; Yuan, Y.; Liu, S.; Jia, J. LISA: Reasoning Segmentation via Large Language Model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 9579–9589. [CrossRef]
9. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual instruction tuning. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 10–16 December 2023; Volume 36, pp. 34892–34916.
10. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023.
11. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Chen, W. LoRA: Low-rank adaptation of large language models. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
12. Xu, Z.; Zhang, X.; Li, R.; Tang, Z.; Huang, Q.; Zhang, J. FakeShield: Explainable Image Forgery Detection and Localization via Multi-modal Large Language Models. *arXiv* **2024**, arXiv:2410.02761.
13. Wen, S.; Ye, J.; Feng, P.; Kang, H.; Wen, Z.; Chen, Y.; Wu, J.; Wu, W.; He, C.; Li, W. Spot the Fake: Large Multimodal Model-Based Synthetic Image Detection with Artifact Explanation. *Adv. Neural Inf. Process. Syst.* **2025**, *38*, 58972–59005.
14. Verdoliva, L. Media forensics and deepfakes: An overview. *IEEE J. Sel. Top. Signal Process.* **2020**, *14*, 910–932. [CrossRef]
15. Dong, C.; Chen, X.; Hu, R.; Cao, J.; Li, X. MVSS-Net: Multi-View Multi-Scale Supervised Networks for Image Manipulation Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 3539–3553 . [CrossRef] [PubMed]
16. Guo, X.; Liu, X.; Ren, Z.; Grosz, S.; Masi, I.; Liu, X. Hierarchical Fine-Grained Image Forgery Detection and Localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
17. Karageorgiou, D.; Papadopoulos, S.; Kompatsiaris, I.; Gavves, E. Any-resolution AI-generated image detection by spectral learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 11–15 June 2025; pp. 18706–18717.
18. He, Y.; Gan, B.; Chen, S.; Zhou, Y.; Yin, G.; Song, L.; Sheng, L.; Shao, J.; Liu, Z. ForgeryNet: A Versatile Benchmark for Comprehensive Forgery Analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
19. Li, Y.; Yang, X.; Sun, P.; Qi, H.; Lyu, S. Celeb-DF: A large-scale challenging dataset for deepfake forensics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 3207–3216.
20. Anaissi, A.; Akram, J.; Lin, X.; Saberi, M.; Braytee, A. MAF-GAN: Modality-aware fusion GAN for multimodal fake news detection. In *Companion Proceedings of the ACM Web Conference 2026*; Association for Computing Machinery: New York, NY, USA, 2026; pp. 880–887.
21. Ma, X.; Du, B.; Jiang, Z.; Du, X.; Al Hammadi, A.Y.; Zhou, J. IML-ViT: Benchmarking Image Manipulation Localization by Vision Transformer. *arXiv* **2023**, arXiv:2307.14863.

22. Kang, H.; Wen, S.; Wen, Z.; Ye, J.; Li, W.; Feng, P.; Zhou, B.; Wang, B.; Lin, D.; Zhang, L.; et al. LEGION: Learning to Ground and Explain for Synthetic Image Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Honolulu, HI, USA, 19–20 October 2025.
23. Anaissi, A.; Akram, J.; Chaturvedi, K.; Braytee, A. *Detecting and Understanding Hateful Contents in Memes Through Captioning and Visual Question-Answering*; Springer: Cham, Switzerland, 2025; Volume 15904, pp. 134–146.
24. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the 38th International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021.
25. Li, Y.; Liu, X.; Wang, X.; Lee, B.S.; Wang, S.; Rocha, A.; Lin, W. FakeBench: Probing Explainable Fake Image Detection via Large Multimodal Models. *arXiv* **2024**, arXiv:2404.13306.
26. Li, Y.; Tan, Z.; Lei, Z.; Zhou, X.; Yang, Y. Towards Generalizable AI-Generated Image Detection via Image-Adaptive Prompt Learning. *arXiv* **2025**, arXiv:2508.01603.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
28. Dang, H.; Liu, F.; Stehouwer, J.; Liu, X.; Jain, A.K. On the detection of digital face manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5781–5790.
29. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
30. Keita, M.; Hamidouche, W.; Eutamene, H.B.; Taleb-Ahmed, A.; Hadid, A. DeeCLIP: A robust and generalizable transformer-based framework for detecting AI-generated images. *arXiv* **2025**, arXiv:2504.19876.
31. Yan, Z.; Wang, J.; Wang, Z.; Jin, P.; Zhang, K.Y.; Chen, S.; Yao, T.; Ding, S.; Wu, B.; Yuan, L. Effort: Efficient Orthogonal Modeling for Generalizable AI-Generated Image Detection. *arXiv* **2024**, arXiv:2411.15633.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.