KNOW WHAT YOU DON'T KNOW: CONFIDENCE-CALIBRATED MLLMS FOR TAMPERING TEXT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) can detect, localize, and explain text-image tampering, but lack explicit confidence outputs, which weakens the reliability of detection systems in practical scenarios. While calibration methods allow models to output confidence matching prediction accuracy, they optimize confidence separately and only consider correctness in calibration. Under such paradigm, models cannot learn forensically beneficial knowledge from calibration, nor can they perceive the certainty of their own analytical process. To address this, we propose a three-stage confidence-calibrated training framework that connects self-assessment with forensic learning. First, forensic continual pretraining on large-scale text-image data develops task-specific perceptual knowledge. Second, supervised fine-tuning introduces structured reasoning, answer, and confidence outputs together with a **Confidence Grounding Attention Module** (CGAM). CGAM uses the confidence representation to aggregate selected reasoning and answer representations, with auxiliary authenticity supervision tying this pathway to forensic prediction. Third, Group Relative Policy Optimization jointly optimizes forensic rewards and a **Forensic-Quality-Aware Calibration** (FQAC) reward. The calibration target combines current-rollout correctness and cross-rollout accuracy with token-level reasoning certainty, using the latter as an auxiliary process signal rather than a substitute for empirical correctness. Experiments on four test sets show that our framework outperforms TextShield-R1 and achieves an average Expected Calibration Error of 21.4%, lower than the evaluated calibration baselines.

1 INTRODUCTION

Text images, including contracts, receipts, and identity documents, support a wide range of financial, legal, and administrative transactions, where changing only a few characters can have serious consequences. Detecting such changes requires careful examination of subtle visual traces, including inconsistencies in typography, texture, and spatial alignment. Early task-specific forensic models exploit low-level manipulation traces and formulate tampering localization as a dense prediction problem (Qu et al., 2023). More recently, multimodal large language models (MLLMs) (Bai et al., 2025; Zhu et al., 2025) have extended text-image forensics to explainable analysis. Representative methods, including TextSleuth (Qu et al., 2024), DocShield (Zeng et al., 2026b), and TextShield-R1 (Qu et al., 2026), identify and localize tampering while producing human-interpretable explanations for their judgments. These advances bring MLLMs closer to the role of a forensic analyst. Yet a reliable analyst must do more than reach and explain a conclusion: it must also assess whether the available evidence is sufficient to trust that conclusion.

In this work, confidence denotes the model's assessment of whether its authenticity verdict is correct. This assessment has a direct operational role: high-confidence predictions may be accepted automatically, whereas uncertain cases should be routed to further inspection. However, as illustrated in Figure 1, existing forensic MLLMs typically output reasoning processes and predictions, without explicit confidence scores. Confidence calibration offers a natural response to this gap by aligning reported confidence with empirical correctness, but do not directly supervise the forensic representations or evidence analysis that produce the verdict. A model may therefore become

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

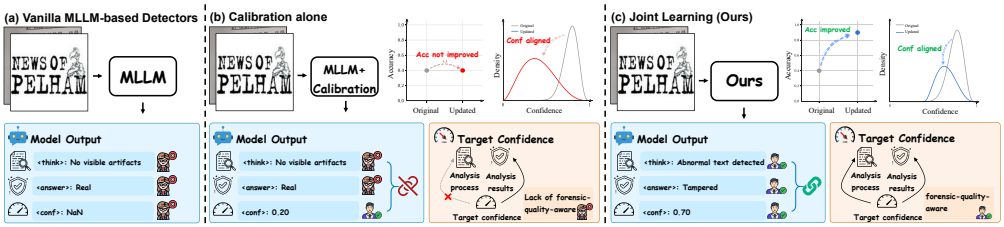


Figure 1: (a) Conventional MLLM-based methods produce no explicit confidence outputs. (b) Existing general confidence calibration methods optimize confidence without introducing new knowledge into the forensics workflow and only consider accuracy when defining target confidence. (c) Our approach ties confidence learning signals to forensic evidence, leverages the certainty of model reasoning for confidence estimation, and builds a joint calibration-forensics optimization framework. Example images are from the Text Forensics Reasoning (TFR) benchmark (Qu et al., 2026).

better at acknowledging its errors without becoming better at resolving the visual evidence behind them. Furthermore, for forensic tasks, calibrating solely toward verdict correctness fails to differentiate whether the model conducts valid image analysis or simply guesses right, which may produce target confidence scores inconsistent with the model's true capacity. To solve these problems, we study confidence not only as an output to calibrate, but also as a learning signal that can connect self-assessment with forensic analysis.

We propose a three-stage confidence-calibrated training framework for tampering text detection. First, forensic continual pretraining on large-scale text-image data equips a general-purpose MLLM with task-specific perceptual and semantic knowledge. Second, supervised fine-tuning endows the model with structured reasoning, answer, and confidence outputs. In this stage, confidence supervision must interact with the representations used for forensic prediction rather than terminate at an isolated confidence token. To establish this interaction, we introduce **Confidence Grounding Attention Module** (CGAM) which uses the confidence-carrying representation to aggregate selected reasoning and answer representations, and requires the resulting representation to predict the image authenticity. This auxiliary objective sends forensic supervision through the confidence-conditioned representation pathway, encouraging self-assessment to remain tied to category-discriminative forensic information. Third, Group Relative Policy Optimization (GRPO) jointly optimizes forensic rewards and a **Forensic-Quality-Aware Calibration** (FQAC) reward. The forensics reward further enhances the model's forgery detection capability. And the FQAC reward combines the current-rollout correctness and cross-rollout accuracy with the model's token-level certainty over its generated reasoning as the target confidence. With reasoning certainty serving as an auxiliary signal, it enables the model to produce confidence aligned with the prediction accuracy while retaining the ability to perceive the certainty of its own analytical process.

Experiments on four test sets show improvements in both overall forensic performance and confidence calibration. Our framework demonstrates broadly stronger forensic performance than TextShield-R1 (Qu et al., 2026) and achieves an average Expected Calibration Error (ECE) of 21.4% ($\downarrow$ 15.4 points), lower than all evaluated calibration baselines. The ablations reveal that introducing CGAM improves all four averaged forensic metrics and reduces ECE; the FQAC reward then achieves the best calibration performance among the ablated variants. These findings support our central premise: with appropriate connections to forensic supervision, learning to assess certainty can help an MLLM not only report more reliable confidence but also form better forensic judgments.

Our contributions are summarized as follows:

- **Joint learning of self-assessment and forensic analysis.** We propose a three-stage framework to investigate how confidence learning can support both calibration and tampering text detection.
- **Confidence Grounding Attention Module (CGAM).** We introduce confidence-conditioned attention over selected reasoning and answer representations, with auxiliary authenticity supervision connecting confidence learning to forensic representation learning.

- **Forensic-Quality-Aware Calibration (FQAC) reward.** We complement current-rollout correctness and cross-rollout accuracy with reasoning certainty as an auxiliary process signal, integrating this target into reinforcement learning alongside forensic rewards.
- **Empirical evidence for coupled training.** Evaluations on four test sets demonstrate gains in overall forensic performance and calibration, while ablations distinguish the effects of confidence supervision alone from those of representation coupling and joint reward optimization.

2 Related Works

2.1 Tampering Text Detection

Early methods treat tampering text detection as dense prediction and design architectures that expose low-level forensic traces, such as the frequency-aware detector introduced with the DocTamper benchmark by Qu et al. (2023), ASCFormer (Luo et al., 2025) and FFDN (Chen et al., 2024). Later work shifts toward realistic, open-set evaluation: Duan et al. (2025) probe generalization to real-world tampering with the RealDTT dataset, and Qu et al. (2025) target text edited by unseen generation models.

MLLMs have recently been adopted because they detect, localize, and explain tampering within a single generative process. Qu et al. (2024) formulate explainable tampered text detection with human-interpretable forensic rationales, Zeng et al. (2026b) pursue evidence-grounded agentic reasoning for document safety, and Qu et al. (2026) combine reinforcement learning with forensic continual pretraining to reach state-of-the-art accuracy. However, these methods lack explicit confidence outputs. They cannot distinguish reliable detection results from those requiring manual review, which limits their trustworthiness in real-world scenarios.

2.2 Confidence Calibration

Calibration aims to align stated confidence with empirical correctness; Guo et al. (2017) establish its modern formulation along with the Expected Calibration Error (ECE) and post-hoc scaling. For LLMs, attention has turned to *verbalized* confidence: Xiong et al. find it systematically overconfident, Tian et al. (2023) show that RLHF degrades calibration, and Leng et al. (2025) attribute this to reward models favoring confident language.

MLLMs are harder to calibrate, since an error may stem from perception or from reasoning. Xiao et al. (2026) decouple confidence into visual and reasoning components and supervise them with intrinsic visual certainty. These methods still supervise confidence with answer-level correctness, however, so it reflects the outcome of a prediction rather than the reliability of the analysis that produced it. This is inadequate for tampering text detection, a fine-grained forensic task in which a correct verdict may still be a lucky guess and confidence should indicate how firmly the evidence was traced. In addition, existing methods generally optimize confidence as a standalone objective, without leveraging the learned knowledge to improve the model's classification or localization performance. We therefore construct a task-aware target from the model's reasoning certainty, current-rollout correctness, and cross-rollout accuracy, optimized jointly with a forensic reward under GRPO (Shao et al., 2024).

3 Method

3.1 Overview

We propose a confidence-calibrated training framework for tampering text detection, depicted in Figure 2. The framework trains an MLLM through three stages. In the first stage, we perform forensic continual pretraining on a large corpus of text images to endow a base MLLM with the low-level and semantic perception required for tampering analysis, yielding a model specialized for tampering text detection. In the second stage, we fine-tune the model with a structured output format and introduce the **Confidence Grounding Attention Module (CGAM)**, which couples the confidence representation to selected reasoning and answer representations through auxiliary

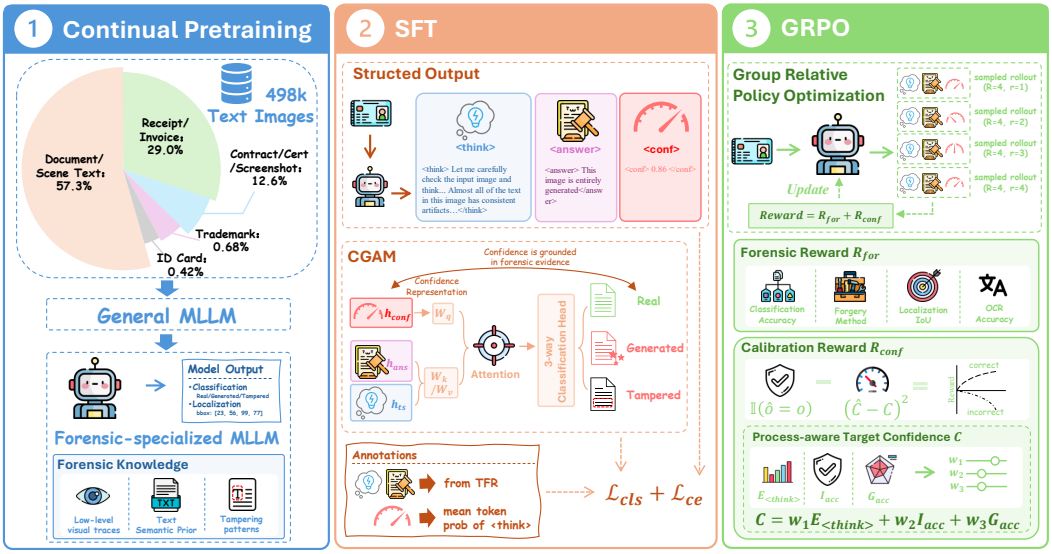


Figure 2: Overview of our confidence-calibrated training framework for tampering text detection. The framework consists of three stages: forensic continual pretraining, supervised fine-tuning with the Confidence Grounding Attention Module (CGAM), and reinforcement learning with a forensic reward and a Forensic-Quality-Aware Calibration (FQAC) reward.

authenticity supervision. In the third stage, we optimize the model with reinforcement learning using a forensic reward and a **Forensic-Quality-Aware Calibration (FQAC) reward**, encouraging accurate forensic predictions and confidence that reflects verdict correctness. The calibration target combines current-rollout correctness and cross-rollout accuracy with reasoning certainty as an auxiliary process signal. Together, representation coupling and joint reward optimization connect confidence learning with forensic analysis, rather than treating self-assessment as an isolated output.

3.2 Forensic Continual Pretraining

In the first stage, we equip a general-purpose MLLM with the perceptual and semantic prior required for tampering text detection by continually pretraining it on a large collection of text-image datasets that span real, entirely generated, and locally tampered samples. The full composition of the pretraining corpus is summarized in Table 6 in the appendix. The pretraining set totals **498k** images, and the held-out test set used for evaluation totals **82k** images. After this stage, the model is specialized for tampering text detection and serves as the initialization for the next stage.

3.3 Supervised Fine-Tuning with CGAM

In the second stage, we fine-tune the model with a structured output format that exposes `<think>`, `<answer>`, and `<conf>` as separate spans, and introduce the **Confidence Grounding Attention Module (CGAM)**. CGAM ties the confidence-carrying representation to authenticity prediction through an auxiliary classification objective, providing an explicit pathway for forensic supervision to shape the representations used for self-assessment.

Structured output format. The model is trained to generate `<think>`, which contains the chain of forensic analysis, `<answer>`, which reports the authenticity verdict together with the tampered region, and a dedicated `<conf>` token that carries the model's confidence. The annotations of the `<think>` and `<answer>` spans come from TFR (Qu et al., 2026), whereas the `<conf>` label is determined in real time from the average token probability of the `<think>` span that the model itself produces. Detailed examples of this output format are provided in the Appendix.

Confidence Grounding Attention Module. Predicting a confidence value does not by itself require the confidence-carrying representation to retain category-discriminative forensic information. Consequently, confidence supervision alone may improve self-assessment without strengthening the

representations that support the authenticity verdict. CGAM addresses this limitation by using the hidden state associated with the confidence output to condition the aggregation of selected reasoning and answer representations. The aggregated representation is then supervised for authenticity classification, explicitly connecting confidence learning with the forensic prediction task.

Concretely, let $\mathbf{h}_{\text{ts}}$ and $\mathbf{h}_{\text{ans}}$ denote the hidden states of the thinking-start token and the conclusion token of `<answer>`, respectively. These states provide the selected reasoning and answer representations for aggregation. CGAM reuses the confidence representation $\mathbf{h}_{\text{conf}}$—the average of the features from the last layer corresponding to the confidence score tokens—as a query and lets it attend to the thinking-start token and the conclusion token as keys and values:

$$\mathbf{q} = \mathbf{W}_q \mathbf{h}_{\text{conf}}, \quad \mathbf{k}_j = \mathbf{W}_k \mathbf{h}_j, \quad \mathbf{v}_j = \mathbf{W}_v \mathbf{h}_j, \quad j \in \mathcal{J} = \{\text{ts}, \text{ans}\}, \tag{1}$$

$$\alpha_j = \exp\left(\mathbf{q}^\top \mathbf{k}_j / \sqrt{d}\right) / \sum_{j' \in \mathcal{J}} \exp\left(\mathbf{q}^\top \mathbf{k}_{j'} / \sqrt{d}\right), \qquad \hat{\mathbf{o}}_{\mathbf{attn}} = \sum_j \alpha_j \mathbf{v}_j. \tag{2}$$

The attended representation is required to encode the image authenticity, supervised by a three-way classification head over the categories of real, entirely generated, and locally tampered:

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_{cls}\, \hat{\mathbf{o}}_{\mathbf{attn}} + \mathbf{b}_{cls}), \qquad \mathcal{L}_{\text{cls}} = -\sum_{k=1}^{3} y_k \log \hat{y}_k, \tag{3}$$

where y_k is the one-hot ground-truth label and $\hat{y}_k$ is the predicted probability of the k-th category. This auxiliary loss sends authenticity supervision through the confidence-conditioned aggregation pathway, encouraging the participating representations to retain forensic information.

3.4 Reinforcement Learning

In the third stage, we further optimize the model with reinforcement learning to align the expressed confidence with the actual correctness while preserving forensic accuracy.

Following the design of Qu et al. (2026), we adopt four forensic rewards that score the correctness of the verdict: (i) **Real/Fake Classification Reward $\mathcal{R}_{\text{cls}}$:** 1 for correct image-level three-way classification (real, fully generated, locally tampered), and 0 otherwise. (ii) **Forgery Method Detection Reward $\mathcal{R}_{\text{mth}}$:** 1 if the model correctly classifies the tampering type as copy-paste or generation, and 0 otherwise. (iii) **Tampering Localization Reward $\mathcal{R}_{\text{loc}}$:** For tampered images, reward equals the IoU between predicted and ground-truth tampered regions when IoU > 0.5, otherwise 0. (iv) **Tampered Text OCR Reward $\mathcal{R}_{\text{ocr}}$:** For tampered images, reward is 1 minus the normalized Levenshtein distance between recognized tampered text and ground-truth text.

The total forensic reward is the sum of the four terms:

$$\mathcal{R}_{\text{for}} = \mathcal{R}_{\text{cls}} + \mathcal{R}_{\text{mth}} + \mathcal{R}_{\text{loc}} + \mathcal{R}_{\text{ocr}}. \tag{4}$$

Forensic-Quality-Aware Calibration reward. Confidence continues to denote the model's assessment of whether its authenticity verdict is correct. Standard calibration objectives supervise this assessment using outcome-level correctness, but do not explicitly incorporate information about the analysis that produced the verdict. In tampering text detection, that analysis involves fine-grained visual content and subtle manipulation traces, motivating an auxiliary signal from the generated reasoning. We use the average token probability of the `<think>` span as a proxy for the model's internal certainty over its reasoning sequence. This quantity reflects how probable the generated tokens are under the model; it is not an independent measure of evidential sufficiency or reasoning correctness, and may also reflect fluency or familiar expression patterns. We therefore combine it with current-rollout correctness and cross-rollout accuracy. The correctness terms anchor the target in empirical reliability, while reasoning certainty supplements them with a process-related signal.

Let $\hat{o}$ and o denote the predicted and ground-truth authenticity categories, and let $\hat{C}$ and C denote the estimated confidence and the target confidence of the current rollout. The calibration reward is defined as

$$\mathcal{R}_{\text{conf}} = \mathbb{I}(\hat{o} = o) - (\hat{C} - C)^2. \tag{5}$$

The target confidence C combines two correctness-based terms with a reasoning-certainty proxy. Their roles are complementary: correctness provides empirical supervision, while the proxy supplies auxiliary information from the generated analysis:

$$C = w_1 \cdot E_{\text{think}} + w_2 \cdot A_{\text{cur}} + w_3 \cdot A_{\text{cross}}, \tag{6}$$

where

$$E_{\text{think}} = \frac{1}{N}\sum_{i=1}^{N} p_i, \quad A_{\text{cur}} = \mathbb{I}(\hat{o} = o), \quad A_{\text{cross}} = \frac{1}{R}\sum_{r=1}^{R} \mathbb{I}(\hat{o}^{(r)} = o). \tag{7}$$

Here p_i is the token probability of the i-th token in the <think> span, and E_{think} serves as the auxiliary reasoning-certainty proxy; A_{cur} indicates whether the verdict of the current rollout matches the ground truth; and A_{cross} is the average correctness over the R sampled rollouts of the same sample. The three weights $w_1 = 0.50, w_2 = 0.25, w_3 = 0.25$ control the contribution of each component. The final reward combines the forensic reward and the calibration reward:

$$\mathcal{R} = \mathcal{R}_{\text{for}} + \mathcal{R}_{\text{conf}}, \tag{8}$$

By optimizing this objective, the model is encouraged not only to improve its forensic predictions but also to report confidence that better reflects whether its authenticity verdict is correct.

4 Experiment

4.1 Experimental Setup

Datasets and data split. We used Text Forensics Reasoning (TFR) as the benchmark. For training, the corpus of Table 6 is used for forensic continual pretraining. TFR's training set is split into two parts, with about 25% used for the supervised fine-tuning and the remaining 75% reserved for reinforcement learning. The data setup for SFT and RL follows TextShield-R1.

Implementation details. We use Qwen3-VL-8B as the backbone MLLM. In the forensic continual pretraining stage, the model is trained on the collected corpus for one epoch. In the second stage, we fine-tune the model for 2 epochs with LoRA (Hu et al., 2021) applied to the LLM at rank 32 and alpha 64, while the vision tower and the projector are fully fine-tuned; all parameters are optimized by AdamW with weight decay 0 and a cosine schedule with a warmup ratio of 0.03, using a learning rate of 2×10^{-4} for the LoRA parameters and 1×10^{-5} for the vision tower and the projector. The per-device batch size is 1 with 4 gradient accumulation steps. In the third stage, we apply GRPO for 1 epochs with a learning rate of 1×10^{-6}, a warmup ratio of 0.05, a per-device batch size of 4 with 4 gradient accumulation steps. We sample 4 rollouts per prompt at temperature 1.0.

Evaluation metrics. We assess both forensic performance and calibration quality. Forensic performance is measured by four metrics: *Cls.* denotes the real/generated/tampered classification task with the accuracy metric; *OCR* denotes the tampered text recognition task with the OCR accuracy metric; *Loc.* denotes the tampered text localization task with the IoU metric; and *Res.* denotes the forgery reasoning task, using the average score of cosine similarity, ROUGE-L, and BLEU as the metric. Calibration quality is measured by the Expected Calibration Error (ECE), where lower is better. Partitioning the estimated confidences into M equally spaced bins, ECE is computed as

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left| \text{acc}(B_m) - \text{conf}(B_m) \right|, \tag{9}$$

where B_m is the set of samples whose estimated confidence falls into the m-th bin, N is the total number of samples, and $\text{acc}(B_m)$ and $\text{conf}(B_m)$ are the average accuracy and the average confidence within that bin. M is set to 10 in our experiments.

4.2 Forensic Comparison

Table 1 reports the forensic performance of our method against other MLLM-based detectors. We evaluate all models, including TextShield-R1, without OCR Rectification for two reasons: its IoU gains do not necessarily reflect improved tampering localization by the model, and noise in the OCR

Table 1: Quantitative comparison with different MLLMs on four test sets. The last four columns (*Average*) report the mean of each metric over the four test sets. The best results are highlighted in **bold** and the second-best results are underlined.

Method	Test set				CIS set				CTM set				CL set				Average			
	Cls.	OCR	Loc.	Res.	Cls.	OCR	Loc.	Res.	Cls.	OCR	Loc.	Res.	Cls.	OCR	Loc.	Res.	Cls.	OCR	Loc.	Res.
Official pre-trained base MLLMs without fine-tuning																				
MiniCPM-V-2.6	30.4	1.6	0.0	3.2	31.4	4.8	0.0	3.2	26.2	7.0	0.0	2.5	30.9	0.4	0.0	1.9	29.7	3.5	0.0	2.7
InternVL3-2B	33.2	5.4	0.0	8.5	40.0	14.0	0.1	9.9	20.3	18.2	0.4	4.0	34.5	6.7	0.0	6.9	32.0	11.1	0.1	7.3
InternVL3-8B	40.4	9.3	0.2	17.9	47.0	20.9	0.8	20.3	25.2	31.5	1.7	8.8	45.1	10.3	0.5	18.1	39.4	18.0	0.8	16.3
Qwen2.5-VL-3B	46.2	1.8	0.1	9.5	48.4	5.9	0.3	15.3	42.2	7.7	0.2	5.6	47.1	1.5	0.2	7.9	46.0	4.2	0.2	9.6
Qwen2.5-VL-7B	42.6	6.4	0.1	9.5	49.9	19.4	0.4	17.6	34.0	21.0	0.6	4.5	50.1	11.1	0.2	10.4	44.2	14.5	0.3	10.5
Qwen3-VL-8B	18.5	1.1	0.0	12.4	13.4	2.1	0.0	8.8	42.5	8.9	0.2	30.0	21.5	0.0	0.1	13.5	24.0	3.0	0.1	16.2
Models fine-tuned with full training set images																				
SparseViT	73.4	-	16.0	-	72.1	-	21.4	-	62.5	-	8.7	-	75.5	-	19.3	-	70.9	-	16.4	-
FFDN	76.1	-	24.7	-	75.2	-	29.9	-	75.6	-	2.0	-	73.4	-	5.6	-	75.1	-	15.6	-
ADCD-Net	70.0	-	34.1	-	71.0	-	35.1	-	82.0	-	20.4	-	66.3	-	14.5	-	72.3	-	26.0	-
MiniCPM-V-2.6	76.2	17.6	11.2	41.1	71.7	24.5	22.8	32.3	64.8	20.7	24.9	27.3	81.5	33.6	20.5	40.3	73.6	24.1	19.9	35.3
InternVL3-2B	75.4	18.1	10.3	40.6	68.5	23.1	21.4	32.0	62.5	18.7	26.2	25.0	80.2	31.7	21.0	39.5	71.7	22.9	19.7	34.3
InternVL3-8B	78.6	21.9	15.4	41.7	70.8	27.6	25.2	33.8	67.7	23.6	31.4	32.3	84.3	38.0	24.8	42.0	75.4	27.8	24.2	37.5
Qwen2.5-VL-3B	77.5	18.6	11.6	42.9	72.3	25.0	20.9	33.6	63.0	18.8	25.6	24.6	80.9	32.4	21.4	39.7	73.4	23.7	19.9	35.2
Qwen2.5-VL-7B	79.1	24.3	18.2	42.9	71.1	30.7	26.5	35.7	73.6	26.3	34.2	36.2	85.1	38.2	25.5	43.1	77.2	29.9	26.1	39.5
Qwen3-VL-8B	85.8	35.9	28.2	52.3	82.3	43.0	35.7	45.8	84.5	32.2	44.1	46.3	82.6	33.2	27.4	43.3	83.8	36.1	33.9	46.9
FakeShield	70.5	9.2	5.4	35.6	62.0	14.8	10.6	29.2	57.5	15.1	17.3	21.4	70.3	23.9	11.3	34.8	65.1	15.8	11.2	30.3
SIDA	71.2	9.2	5.6	35.7	62.2	14.9	10.8	29.4	57.5	15.1	17.3	21.5	70.4	23.8	11.5	25.0	65.3	15.8	11.3	27.9
TextShield-R1	88.1	47.6	42.7	58.8	72.9	62.1	**56.6**	**56.5**	**88.8**	45.6	57.9	51.2	85.5	39.0	**32.3**	46.2	83.8	48.6	47.4	53.2
Ours	**89.4**	**58.1**	**50.4**	**59.8**	**89.5**	**65.7**	55.0	54.5	88.4	**55.8**	**69.5**	**53.6**	**96.5**	**47.5**	31.8	**61.5**	**91.0**	**56.8**	**51.7**	**57.4**

tool used by TextShield-R1 can produce anomalous rectified regions. Appendix A.2.4 provides a detailed explanation with qualitative examples and supporting statistics. In addition, we also compared several expert models: SparseViT (Su et al., 2025), FFDN (Chen et al., 2024), and ADCD-Net (Wong et al., 2025). Since the expert models can only output classification and localization results, we report only the Cls. and Loc. metrics. Averaged over the four test sets, our method consistently outperforms both TextShield-R1 and Qwen3-VL-8B fine-tuned on the full training set across all four metrics. It surpasses TextShield-R1 in 12 of 16 comparisons, with the remaining differences occurring in CIS localization and reasoning, CTM classification, and CL localization. These results demonstrate the effectiveness of the complete framework across classification and fine-grained forensic tasks.

4.3 Calibration Comparison

Table 2 compares the ECE of the complete framework with a confidence-supervised baseline and its DCPO (Ma et al., 2026) and RLCR (Damani et al., 2026) variants. The baseline uses the average token probability of the `<think>` span as the `<conf>` label during SFT, without CGAM; DCPO and RLCR apply their respective calibration objectives to this baseline. Ours denotes the full configuration with CGAM and both forensic and calibration rewards, as in Table 3. The baseline's average ECE of 26.7 indicates substantial residual miscalibration, but ECE alone does not distinguish overconfidence from underconfidence. DCPO and RLCR reduce average ECE to 22.9 and 23.7, respectively, while our framework reaches 21.4, an improvement of 1.5 and 2.3 points.

Figure 3 provides an auxiliary analysis on a random 10% subset of the test set of the pretraining data, distinct from the four-test-set evaluation above. In the plotted examples, DCPO and RLCR approach the diagonal in high-confidence bins, but their low-confidence bins retain accuracy above reported confidence, indicating underconfidence. Our method shows closer agreement between confidence and empirical accuracy across these regions. This pattern is consistent with improved calibration on the illustrated distribution.

Table 2: Calibration comparison in terms of ECE (%, lower is better) across the four test sets. The best result is highlighted in **bold**.

Method	Test Set	CIS Set	CTM Set	CL Set	Average
baseline	23.9	25.3	31.8	25.9	26.7
+RLCR	20.7	23.6	27.9	22.5	23.7
+DCPO	19.3	22.9	27.5	22.0	22.9
Ours	**17.3**	**21.5**	**24.9**	**21.9**	**21.4**
Δ	−6.6	−3.8	−6.9	−4.0	−5.3

Table 3: Ablation study of each component. All metrics are averaged over the four test sets; ECE (%, lower is better) measures calibration.

Configuration				Average				
<conf>	CGAM	$\mathcal{R}_{\text{for}}$	$\mathcal{R}_{\text{conf}}$	Cls.	OCR	Loc.	Res.	ECE↓
–	–	–	–	89.3	54.6	50.5	52.7	36.8
✓	–	–	–	89.6	52.3	47.9	51.7	26.7
✓	✓	–	–	90.8	55.4	50.8	55.4	24.0
✓	✓	✓	–	**91.5**	56.5	51.2	56.9	23.8
✓	✓	✓	✓	91.0	**56.8**	**51.7**	**57.4**	**21.4**

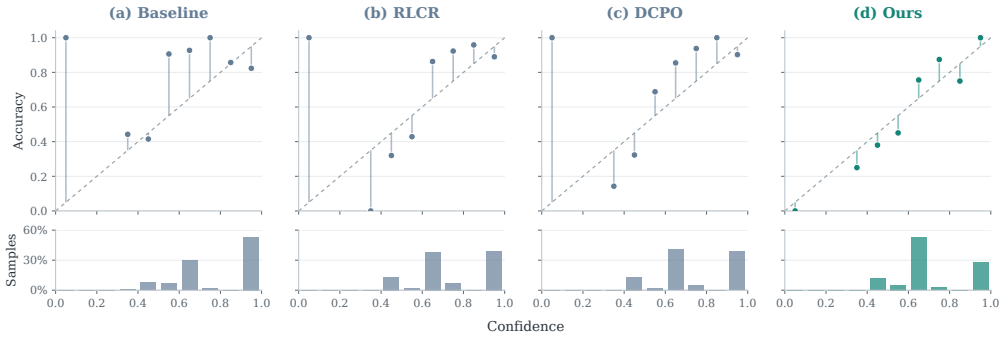


Figure 3: Auxiliary reliability diagrams on a random 10% subset of the pretraining data (test set), separate from the four-test-set evaluation. Bars (left axis) show the sample fraction in each confidence bin; curves (right axis) show mean classification accuracy; the dashed diagonal indicates agreement between confidence and accuracy.

4.4 Ablation Study

To examine the contribution of each component, we report the average performance over the four test sets together with the ECE.

Effect of each component. Table 3 reports incremental ablations. Adding <conf> supervision substantially reduces ECE from 36.8 to 26.7, but slightly degrades OCR, Loc., and Res., indicating that confidence supervision alone does not consistently improve forensics. CGAM provides the largest incremental forensic gains and further reduces ECE to 24.0, supporting the benefit of coupling confidence representations with authenticity supervision. Adding $\mathcal{R}_{\text{for}}$ further improves all four forensic metrics and lowers ECE to 23.8. Finally, $\mathcal{R}_{\text{conf}}$ achieves the best ECE of 21.4 while further improving OCR, Loc., and Res., at the cost of a slight decrease in Cls. Overall, the complete configuration achieves the best calibration and fine-grained forensic performance in this ablation, though not the highest classification accuracy.

Role of $\mathbf{h}_{\text{conf}}$. Table 4 replaces the confidence query with a randomly initialized learnable embedding. This degrades Cls., OCR, Loc., and Res. and increases ECE from 21.4 to 22.1, indicating that the sample-specific confidence representation is important for conditioning representation aggregation and linking self-assessment with forensic prediction.

Components of the target confidence C. Table 5 compares target variants against the CGAM model trained with $\mathcal{R}_{\text{for}}$ but without $\mathcal{R}_{\text{conf}}$ (ECE 23.8).

Neither source alone yields the best ECE: using only A_{cur} and A_{cross} gives 22.0, while using only E_{think} gives 22.4; their combination achieves the best ECE of 21.4. This suggests that empirical correctness and reasoning information provide complementary signals, as token probability alone remains only a proxy for reasoning certainty. The combined target also achieves the best OCR, Loc., and Res., further supporting their complementarity.

Table 4: Confidence-query ablation; Δ is relative to the learnable embedding.

Query	Average				
	Cls.	OCR	Loc.	Res.	ECE↓
Learnable	90.6	54.9	50.7	56.1	22.1
$\mathbf{h}_{\text{conf}}$	**91.0**	**56.8**	**51.7**	**57.4**	**21.4**
Δ	+0.4	+1.9	+1.0	+1.3	−0.7

Table 5: Ablation of target-confidence components. The first row uses only the forensic reward.

Target components	Cls.	OCR	Loc.	Res.	ECE↓
None	91.5	56.5	51.2	56.9	23.8
$A_{\text{cur}} + A_{\text{cross}}$	**91.8**	55.9	51.1	56.4	22.0
E_{think}	91.1	56.7	51.6	57.2	22.4
$E_{\text{think}} + A_{\text{cur}} + A_{\text{cross}}$	91.0	**56.8**	**51.7**	**57.4**	**21.4**

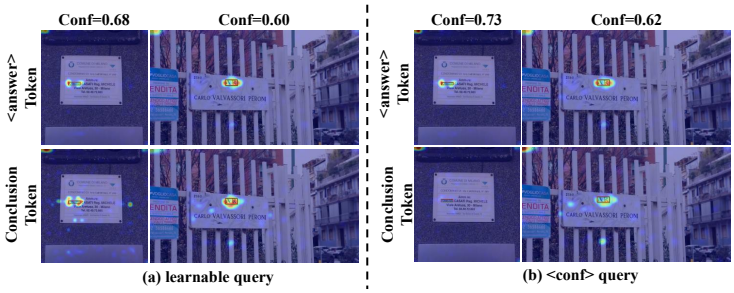


Figure 4: Backbone self-attention from the conclusion token (i.e., the categorical verdict token within the `<answer>` output, e.g., real/tampered/generated) and `<answer>` token sequence to the visual tokens. In the illustrated examples, (a) a learnable CGAM query produces similar visual attention for high- and low-certainty traces; (b) using $\mathbf{h}_{\text{conf}}$ yields weaker visual attention for a high-certainty `<think>` trace and stronger attention for a low-certainty one. These patterns are consistent with confidence-conditioned representation learning.

The correctness-only target achieves the highest Cls. (91.8), while the combined target trades 0.8 points in Cls. for gains of 0.9, 0.6, and 1.0 points in OCR, Loc., and Res., respectively, and reduces ECE by 0.6. Thus, incorporating E_{think} improves fine-grained forensics and calibration without reducing training to classification alone, supporting the benefit of joint confidence and forensic learning.

4.5 CGAM's Influence on MLLM Self-Attention

Figure 4 visualizes the backbone MLLM's self-attention. In the illustrated high-certainty case, attention from the conclusion token to the visual tokens is weaker, although the remaining template tokens of `<answer>` still attend strongly to the image. In the low-certainty case, the conclusion token shows stronger visual attention. Replacing $\mathbf{h}_{\text{conf}}$ with a randomly initialized learnable embedding reduces this contrast in the illustrated examples. These observations are consistent with confidence-conditioned representation learning being associated with different patterns of visual evidence use, demonstrating that CGAM effectively integrates confidence semantics into the MLLM's forensic analysis.

5 Conclusion

We present a three-stage confidence-calibrated framework that connects model self-assessment with forensic learning to improve the reliability of MLLM-based text tampering detection. Unlike conventional calibration methods that optimize confidence independently and neglect forensic task knowledge, our framework tightly unifies confidence estimation and forensic prediction to boost model practical credibility. Integrating the Confidence Grounding Attention Module (CGAM) based representation coupling with joint forensic and calibration optimization, our method achieves an average ECE of 21.4% across four test sets and outperforms the TextShield-R1 baseline in detection accuracy and confidence alignment. Ablation results further verify that coupling confidence learning with forensic supervision mutually promotes calibration precision and forensic discrimination performance.

6 AI USE STATEMENT

In this work, generative AI tools were used to process experimental code and draft framework diagrams. In addition, we leveraged generative AI tools to polish the manuscript and improve its readability. All AI-assisted outputs have been reviewed to verify the correctness of experimental codes and figures included in the paper. We take full responsibility for the final content of this work, including AI-generated texts, codes, and other materials.

REFERENCES

"Security AI Challenger Program.". https://tianchi.aliyun.com/competition/entrance/531812/introduction, 2020.

"Real-World Image Forgery Localization Challenge.". https://tianchi.aliyun.com/competition/entrance/531945/introduction, 2022.

Chloé Artaud, Antoine Doucet, Jean-Marc Ogier, and Vincent Poulain d'Andecy. Receipt dataset for fraud detection. In *First International Workshop on Computational Document Forensics*, 2017.

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

Zhongxi Chen, Shen Chen, Taiping Yao, Ke Sun, Shouhong Ding, Xianming Lin, Liujuan Cao, and Rongrong Ji. Enhancing tampered text detection through frequency feature fusion and decomposition. In *European Conference on Computer Vision*, pp. 200–217. Springer, 2024.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training lms to reason about their uncertainty. In *International conference on learning representations*, volume 2026, pp. 59456–59493, 2026.

Li Dong, Weipeng Liang, and Rangding Wang. Robust text image tampering localization via forgery traces enhancement and multiscale attention. *IEEE Transactions on Consumer Electronics*, 70(1): 3495–3507, 2024.

Junxian Duan, Hao Sun, Fan Ji, Kai Zhou, Zhiyong Wang, Huaibo Huang, and Lianwen Jin. Realdtt: Towards a comprehensive real-world dataset for tampered text detection: J. duan et al. *International Journal of Computer Vision*, 133(10):6993–7011, 2025.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Pavel Korshunov, Amir Mohammadi, Christophe Ecabert, Sébastien Marcel, et al. Fantasyid: A dataset for detecting digital manipulations in id-documents. In *2025 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–9. IEEE, 2025.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in llms: Reward calibration in rlhf. In *International Conference on Learning Representations*, volume 2025, pp. 16484–16517, 2025.

Xin Liao, Siliang Chen, Jiaxin Chen, Tianyi Wang, and Xiehua Li. Ctp-net: Character texture perception network for document image forgery localization. *arXiv preprint arXiv:2308.02158*, 2023.

Dongliang Luo, Yu Zhou, Rui Yang, Yuliang Liu, Xianjin Liu, Jishen Zeng, Enming Zhang, Biao Yang, Ziming Huang, Lianwen Jin, et al. Icdar 2023 competition on detecting tampered text in images. In *International Conference on Document Analysis and Recognition*, pp. 587–600. Springer, 2023.

Dongliang Luo, Yuliang Liu, Rui Yang, Xianjin Liu, Jishen Zeng, Yu Zhou, and Xiang Bai. Toward real text manipulation detection: New dataset and new solution. *Pattern Recognition*, 157:110828, 2025.

Zhengzhao Ma, Xueru Wen, Boxi Cao, Yaojie Lu, Hongyu Lin, Jinglin Yang, Min He, Xianpei Han, and Le Sun. Decoupling reasoning and confidence: Resurrecting calibration in reinforcement learning from verifiable rewards. *arXiv preprint arXiv:2603.09117*, 2026.

Chenfan Qu, Chongyu Liu, Yuliang Liu, Xinhong Chen, Dezhi Peng, Fengjun Guo, and Lianwen Jin. Towards robust tampered text detection in document image: New dataset and new solution. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5937–5946. IEEE, 2023.

Chenfan Qu, Jian Liu, Haoxing Chen, Baihan Yu, Jingjing Liu, Weiqiang Wang, and Lianwen Jin. Textsleuth: Towards explainable tampered text detection. *arXiv preprint arXiv:2412.14816*, 2024.

Chenfan Qu, Yiwu Zhong, Fengjun Guo, and Lianwen Jin. Revisiting tampered scene text detection in the era of generative ai. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 694–702, 2025.

Chenfan Qu, Yiwu Zhong, Jian Liu, Xuekang Zhu, Bohan Yu, and Lianwen Jin. Textshield-r1: Reinforced reasoning for tampered text detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8621–8629, 2026.

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

Lei Su, Xiaochen Ma, Xuekang Zhu, Chaoqun Niu, Zeyu Lei, and Ji-Zhe Zhou. Can we get rid of handcrafted feature extractors? sparsevit: Nonsemantics-centered, parameter-efficient image manipulation localization through spare-coding transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 7024–7032, 2025.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, 2023.

Yuxin Wang, Hongtao Xie, Mengting Xing, Jing Wang, Shenggao Zhu, and Yongdong Zhang. Detecting tampered scene text in the wild. In *European Conference on Computer Vision*, pp. 215–232. Springer, 2022a.

Yuxin Wang, Boqiang Zhang, Hongtao Xie, and Yongdong Zhang. Tampered text detection via rgb and frequency relationship modeling. *Chinese Journal of Network and Information Security*, 8 (3):29–40, 2022b.

Kahim Wong, Jicheng Zhou, Haiwei Wu, Yain-Whar Si, and Jiantao Zhou. Adcd-net: Robust document image forgery localization via adaptive dct feature and hierarchical content disentanglement. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19280–19289. IEEE, 2025.

Wenyi Xiao, XU Xinchi, and Leilei Gan. Vl-calibration: Decoupled confidence calibration for large vision-language models reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 44791–44815, 2026.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms, 2024. *URL https://arxiv. org/abs/2306.13063*.

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

Zeqin Yu, Bin Li, Yuzhen Lin, Jinhua Zeng, and Jishen Zeng. Learning to locate the text forgery in smartphone screenshots. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Zeqin Yu, Haotao Xie, Jian Zhang, Jiangqun Ni, Wenkang Su, and Jiwu Huang. Toward real-world text image forgery localization: Structured and interpretable data synthesis. *Advances in Neural Information Processing Systems*, 38, 2026.

Fanwei Zeng, Jianshu Li, Changtao Miao, Chenqi Kong, Youru Li, Zhi Cai, Haiyan Yin, Joey Tianyi Zhou, Ran He, Anderson Rocha, et al. Gentext-forensics: Challenge on explainable forensics and adversarial generation for text-centric images. 2026a.

Fanwei Zeng, Changtao Miao, Jing Huang, Zhiya Tan, Shutao Gong, Xiaoming Yu, Yang Wang, Weibin Yao, Joey Tianyi Zhou, Jianshu Li, et al. Docshield: Towards ai document safety via evidence-grounded agentic reasoning. *arXiv preprint arXiv:2604.02694*, 2026b.

Peirong Zhang, Haowei Xu, Jiaxin Zhang, Xuhan Zheng, Guitao Xu, Yuyi Zhang, Junle Liu, Zhenhua Yang, Wei Zhou, and Lianwen Jin. Ocrgenbench: A comprehensive benchmark for evaluating ocr generative capabilities. *arXiv preprint arXiv:2507.15085*, 2025.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A APPENDIX

A.1 PRETRAINING DATA

Corpus composition and purpose. Table 6 summarizes the 17 datasets collected for forensic continual pretraining, including DocTamper (Qu et al., 2023) and sources released between 2017 and 2025. The collection spans real, entirely generated, and locally tampered text images. These complementary authenticity categories provide a task-specific basis for learning text-image appearance and manipulation-related patterns before reasoning-oriented training on TFR. This stage adapts Qwen3-VL-8B to the forensic domain rather than serving as the complete confidence-calibrated detector.

Data usage. The corpus has a pretraining partition and a held-out test partition, whose reported aggregate sizes are 498k and 82k images, respectively. For datasets where training and test data were already partitioned in the original papers, we follow the settings specified therein; for other datasets, we randomly select 70% of the images for pre-training and use the remaining 30% as the test set. Only the pretraining partition is used for first-stage optimization, which runs for one epoch. The resulting checkpoint initializes the subsequent SFT stage. The held-out partition supports auxiliary evaluation, including the reliability analysis described in the main text; it is distinct from the four TFR test sets used for the main comparisons. TFR provides the subsequent reasoning-oriented training data, with approximately 25% of its training set used for the SFT cold start and the remaining 75% for reinforcement learning.

Table 6: Dataset collection for forensic continual pretraining. "Num" lists the per-dataset image count; "Pre-train" and "Test" give the reported aggregate partition sizes, not per-dataset allocations.

Dataset	Year	Num	Pre-train	Test
Findit (Artaud et al., 2017)	2017	180	100	80
SACP (sac, 2020)	2020	2,005	1,404	601
RIFLC (rif, 2022)	2022	4,000	3,200	800
T-SROIE (Wang et al., 2022b)	2022	986	626	360
T-IC13 (Wang et al., 2022a)	2022	462	323	139
DocTamper (Qu et al., 2023)	2023	170,000	120,000	50,000
FCTM (Liao et al., 2023)	2023	4,138	2,897	1,241
TTI (Luo et al., 2023)	2023	8,285	5,800	2,485
STFD (Yu et al., 2023)	2023	4,094	2,866	1,228
TextTamper (Dong et al., 2024)	2024	42,656	39,356	3,200
RTM (Luo et al., 2025)	2025	9,000	6,803	3,197
OSTF (Qu et al., 2025)	2025	4,418	3,093	1,325
OCRGenBench (Zhang et al., 2025)	2025	2,120	1,484	636
FSTS (Yu et al., 2026)	2025	294,182	286,989	7,193
RealTextV2 (Zeng et al., 2026a)	2025	13,500	9,450	4,050
FantacyID (Korshunov et al., 2025)	2025	3,284	1,899	1,385
TextDDLE (Xu et al., 2025)	2025	16,834	11,784	5,050
Total	**-**	**580,144**	**498,074**	**82,970**

A.2 MORE EXPERIMENT

A.2.1 PRETRAINING AND FULL-DATA SUPERVISED FINE-TUNING

Experimental setup. We examine whether the performance of our framework can be explained by forensic pretraining alone, and whether directly fine-tuning the pretrained model on all TFR training samples is sufficient to match it. Table 7 compares three configurations using the same backbone and first-stage pretrained checkpoint. *Pre-train* evaluates the checkpoint immediately after forensic continual pretraining, without subsequent TFR training. *Pre-train+SFT* initializes from this checkpoint and performs supervised fine-tuning on the entire TFR training set, without reinforcement learning. *Ours* uses the complete framework, including CGAM and joint forensic and calibration reward optimization, with the TFR training set divided between SFT and RL as described in the main text. Metrics follow the main evaluation protocol and are averaged over the four test sets. Dashes denote unreported metrics, not zero scores.

Table 7: Comparison of the first-stage pretrained model, full-TFR supervised fine-tuning, and our complete framework. All configurations share the same forensic pretraining. Results are averaged over the four test sets; higher is better for forensic metrics and lower is better for ECE (%). Dashes indicate unreported results. Best reported results are bold.

Model	Cls.	OCR	Loc.	Res.	ECE↓
Pre-train	38.8	–	40.9	–	–
Pre-train+SFT	87.9	53.2	48.9	50.3	26.3
Ours	**91.0**	**56.8**	**51.7**	**57.4**	**21.4**

Results and analysis. The first-stage model obtains 38.8 Cls. and 40.9 Loc., indicating that pretraining alone does not reproduce the final detector's performance. Full-TFR SFT substantially improves these scores to 87.9 and 48.9, respectively. Nevertheless, our framework further improves Cls., OCR, Loc., and Res. by 3.1, 3.6, 2.8, and 7.1 points over Pre-train+SFT, while reducing ECE from 26.3 to 21.4, a 4.9-point decrease. Thus, the improvement is not limited to confidence calibration: all four forensic metrics also benefit relative to direct full-data SFT.

Because Pre-train+SFT and Ours share the pretrained initialization and use the same TFR training pool, the latter's advantage cannot be attributed solely to adding the pretraining corpus or accessing more TFR samples. Instead, this comparison supports the benefit of our subsequent training framework beyond pretraining followed by full-data SFT, complementing the component ablations in the main text. Since all three configurations include pretraining, this experiment does not establish that pretraining is unnecessary; it shows that pretraining alone, or combined with conventional full-data SFT, is insufficient to account for the performance of the complete framework.

A.2.2 Confidence Semantics in Hidden Representations

Analysis setup. We examine whether the hidden representation associated with `<conf>` reflects the meaning of a confidence score beyond its surface numerical form. Figure 5 compares layer-wise hidden-state distances for two types of confidence-score pairs. In both cases, the numerical strings differ in only one digit, but their absolute confidence gaps $\Delta = |C_1 - C_2|$ differ substantially: the orange curve corresponds to $\Delta = 0.40$, whereas the blue curve corresponds to $\Delta = 0.01$. This comparison probes how the trained MLLM represents markedly different levels of certainty despite similarly small changes in their written form.

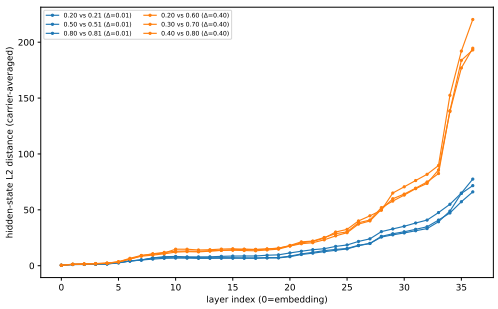


Figure 5: Layer-wise distances between hidden representations associated with `<conf>`. Both types of confidence-score pairs differ in one digit, but the orange curve represents a large confidence gap ($\Delta = 0.40$) and the blue curve a small gap ($\Delta = 0.01$). The larger-gap curve grows more rapidly with depth and separates clearly from the smaller-gap curve in the final layers, consistent with confidence-sensitive hidden representations.

System:
You are a helpful assistant.
User:
<image> Is this image real, entirely generated, or tampered? If it has been tampered, what method was used, and what are the content and bounding box coordinates of the tampered text? Output the thinking process in <think> </think>, \n final answer (number) in <answer> </answer> tags and your confidence from 0 to 1 in <conf> </conf>.
Here is an example answer for a real image: <answer> This image is real. </answer> <conf></conf> Here is an example answer for an entirely generated image: <answer> This image is entirely generated. </answer> <conf></conf> Here is an example answer for a locally tampered image: <answer> This image is tampered. It was tampered by copy-paste. The tampered text reads \"small\" in the text line \"a small yellow flower\", and it is located at ... </answer> <conf></conf>

Figure 6: Prompt used for forensic analysis, specifying the authenticity categories, tampering details, and structured reasoning, answer, and confidence outputs.

Results and interpretation. Hidden-state distances generally increase with layer depth for both types of pairs. More importantly, the orange curve increases more rapidly than the blue curve, with a pronounced separation in the final layers. The relevant evidence is this difference between the curves, rather than the increase in absolute distance alone: a one-digit change that denotes a much larger change in certainty produces a larger representational separation in deeper layers. This pattern supports the interpretation that the trained confidence representation encodes distinctions in confidence magnitude beyond superficial similarity between numerical strings.

Connection to CGAM. CGAM uses $\mathbf{h}_{\text{conf}}$ as the query for aggregating selected reasoning and answer representations, and applies auxiliary authenticity supervision to the aggregated output. The observed separation provides qualitative support for this design: the query can carry information about the expressed level of certainty, rather than merely the surface form of the confidence score. Together with the query ablation in Table 4, this analysis supports using the confidence-carrying hidden state to connect self-assessment with forensic representation learning. Encoding confidence magnitude is distinct from establishing its calibration; the latter is evaluated separately against verdict correctness.

A.2.3 Prompt and Qualitative Response Examples

Prompt and output format. Figure 6 presents the prompt used for text-image forensic analysis. It asks the model to distinguish real, entirely generated, and locally tampered images and, for tampered images, identify the manipulation method, affected text, and bounding box. The response separates forensic reasoning, the final answer, and confidence into `<think>`, `<answer>`, and `<conf>` spans, respectively. Confidence is requested on a scale from 0 to 1, making the authenticity verdict and its self-assessed reliability separately inspectable.

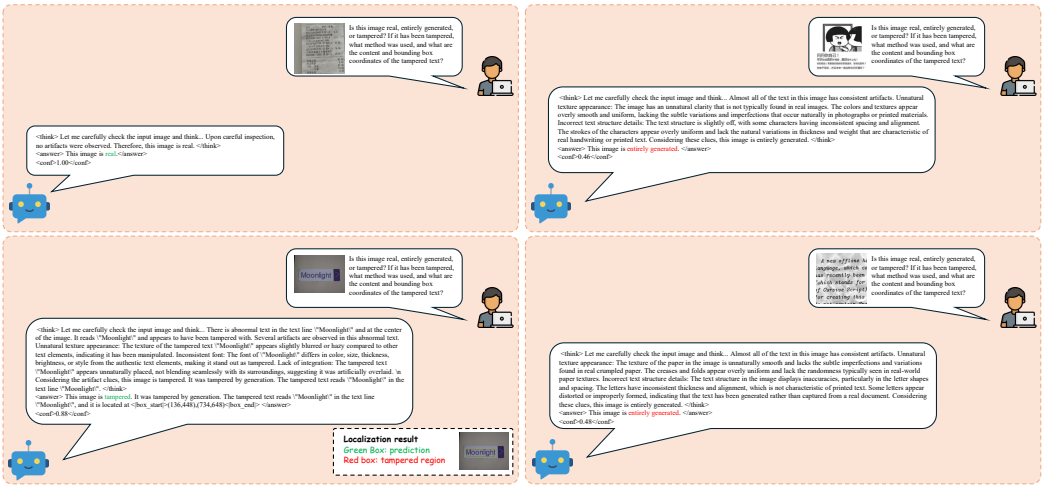


Figure 7: Qualitative response examples with verdict and confidence annotations. Green text highlights correct verdicts; red text flags incorrect verdicts. In the localization inset, green boxes indicate predictions and red boxes indicate ground-truth tampered regions. The test samples are from TFR (Qu et al., 2026).

Qualitative observations. In Figure 7, the real-image example receives a correct verdict with confidence 1.00, while the localized-tampering example identifies the affected text and its region with confidence 0.88. The entirely generated example receives a wrong verdict and confidence lower than 0.50. These examples motivate jointly learning forensic prediction and confidence assessment. More detection and localization results are in Figure 8.

A.2.4 Localization Evaluation without OCR Rectification

Evaluation rationale. We report localization IoU for TextShield-R1 (Qu et al., 2026) and all other models without OCR Rectification. This post-processing step adjusts model-predicted regions using OCR-generated candidate boxes. In TextShield-R1 (Qu et al., 2026), the OCR tool used for



Figure 8: Tampering detection and localization examples of our confidence-calibrated framework. The test samples are from TFR (Qu et al., 2026).

rectification may be the same as that used to obtain the dataset's annotation boxes. Consequently, an increase in IoU can reflect closer agreement with the annotation tool's boundaries rather than an improvement in the model's ability to identify and localize tampered text. Disabling this step consistently across models separates their localization outputs from this additional source of coordinate alignment.

Qualitative evidence. Figure 9 illustrates two situations. Red boxes denote annotated tampered regions, green boxes the model's original predictions, and blue boxes the outputs after OCR Rectification. In the two examples on the left, the model already identifies the correct tampered text, but boundary noise from the annotation tool prevents the original prediction from attaining an IoU of 1. Rectification reproduces the annotation boundaries and raises IoU to 1, without changing which text the model has identified as tampered. In the two examples on the right, noise in the OCR candidate coordinates can cause rectification to output a region spatially inconsistent with the original prediction yet coincident with the ground truth. Such a match should not be interpreted as evidence that the model itself predicted the correct location.



Figure 9: Effects of OCR Rectification on localization evaluation. Red: annotated tampered regions; green: original model predictions; blue: rectified predictions. Left two examples: boundary alignment raises IoU to 1 although the model already identifies the tampered text. Right two examples: rectification can map a spatially inconsistent prediction to the annotated region.

Quantitative evidence. Table 8 summarizes the effect of OCR Rectification on TextShield-R1 across the four evaluation sets. For each set, we count tampered samples whose IoU strictly increases after rectification, report their fraction among all tampered samples, and compute the mean *post-rectification* IoU on that improved subset.

The improved subsets account for 32.84%–54.95% of tampered samples, and their mean rectified IoU ranges from 0.9861 to 0.9995. These near-perfect values are consistent with the illustrated alignment to OCR-derived annotation boxes. These statistics indicate that the gains from OCR Rec-

Table 8: OCR Rectification statistics for TextShield-R1. "Improved" counts tampered samples with higher IoU after rectification; "Share" uses all tampered samples as the denominator. The last column is the mean rectified IoU on the improved subset only, on a $[0, 1]$ scale, not the mean IoU over the full test set.

Dataset	Tampered samples	Improved	Share	Mean rectified IoU (improved subset)
Test Set	4,417	1,610	36.45%	0.9952
CIS Set	4,099	1,346	32.84%	0.9994
CTM Set	384	211	54.95%	0.9861
CL Set	3,001	1,029	34.29%	0.9995

tification partly arise from using the same OCR tool for rectification and dataset annotation. Together with the qualitative analysis above, they suggest that these gains do not entirely reflect improvements in the model's localization ability. Moreover, noise in the OCR tool also affects the reliability of the resulting IoU metric. We therefore compare all models without OCR Rectification, measuring IoU directly from their predicted regions. This protocol evaluates model-level localization under a common post-processing setting rather than the performance of a combined model-and-OCR pipeline.

A.2.5 Analysis of Representation Alignment Properties

Table 9: Layer-wise averaged cosine similarity across samples (representative layers, $N = 450$). $\bar{c}_{\text{pl}}$: mean of w/o CGAM; $\bar{c}_{\text{ca}}$: mean of CGAM; $\Delta\bar{c} = \bar{c}_{\text{ca}} - \bar{c}_{\text{pl}}$. Bold entries denote large effect sizes or notable sign shifts.

Pair	Layer	$\bar{c}_{\text{pl}}$	$\bar{c}_{\text{ca}}$	$\Delta\bar{c}$
h_{ts}–h_{conf}	6	0.617	0.653	+0.036
h_{ts}–h_{conf}	12	0.518	0.572	+0.054
h_{ts}–h_{conf}	18	0.585	0.626	+0.041
h_{ts}–h_{conf}	24	0.331	0.382	+0.051
h_{ts}–h_{conf}	30	0.243	0.341	**+0.098**
h_{ans}–h_{conf}	6	0.642	0.661	+0.019
h_{ans}–h_{conf}	12	0.556	0.565	+0.009
h_{ans}–h_{conf}	18	0.661	0.663	+0.002
h_{ans}–h_{conf}	24	0.568	0.512	-0.057
h_{ans}–h_{conf}	30	0.555	0.623	**+0.068**
h_{ts}–h_{ans}	6	0.515	0.556	+0.042
h_{ts}–h_{ans}	12	0.447	0.486	+0.039
h_{ts}–h_{ans}	18	0.559	0.591	+0.032
h_{ts}–h_{ans}	24	0.337	0.341	+0.004
h_{ts}–h_{ans}	30	0.284	0.342	**+0.059**

Experimental Setup: To explore the influence of the CGAM on the directional alignment of representations at three critical token positions, this paper conducts representation probing experiments. Two model checkpoints fine-tuned based on Qwen3-VL-8B are compared in the experiments: (1) w/o CGAM, which is fine-tuned via online confidence supervision on pre-trained weights without additional attention modules; (2) w/ CGAM, which is jointly trained with the CGAM incorporated during the SFT stage. The CTM test set is adopted for evaluation with a stratified sampling strategy. Specifically, 150 samples are randomly selected from each ground-truth category, forming a total of 450 paired samples for analyzing the correlations among h_{ts}, h_{ans}, and h_{conf}. The token-level positioning rules for the three key positions remain consistent with those applied in the training phase.

Teacher-forced forward propagation is performed on the model-generated responses to record the token-level hidden states of all 37 network layers, where Layer 0 denotes the embedding layer and Layers 1 to 36 correspond to the outputs of Transformer blocks. Mean pooling is implemented on all

tokens within each span to obtain 4096-dimensional positional representation vectors for each layer. The cosine similarity between the pooled representations of two spans in each layer for every sample is utilized as the measurement of representation directional alignment, and the average results over all 450 samples are reported.

Results and Discussion: As illustrated in Table 1, the cosine similarities of both h_{ts}–h_{conf} and h_{ts}–h_{ans} increase synchronously across most of the layers. This reveals that joint training equipped with CGAM systematically enhances the directional alignment between the starting chain-of-thought representation, confidence output, and classification decision across most network depths. The average difference $\Delta\bar{c}$ generally ranges from $+0.04$ to $+0.10$ in the middle and late network layers.

In contrast, the representation pair of h_{ans}–h_{conf} presents a more complex variation pattern. Its cosine similarity rises steadily from shallow to middle layers, yet the correlation weakens at Layer 24. This may arise because the confidence semantics within h_{conf} are not well-formed in shallow layers, preventing stable correlation between h_{ans} and h_{conf}.