Forgery Evidence Peaks Mid-Stack: Forensic Evidence Relay for Multimodal Forgery Detection

Yingxin Lai[1*] **Xinyuan Wang**[1*] **Yufei Liu**[1] **Jialin Guo**[1]
Kailin Lyu[3] **Zhiming Luo**[1,2†] **Shaozi Li**[1,2]
[1]Department of Artificial Intelligence, Xiamen University
[2]Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University, Xiamen, China
[3]Institute of Automation, Chinese Academy of Sciences
laiyingxin2@gmail.com

Abstract

Multimodal large language models (MLLMs) are increasingly used for explainable image forgery analysis, where a shared decoder is expected to predict authenticity, localize evidence, and generate natural-language explanations. Existing methods typically rely on final decoder states, leaving open whether these late, language-oriented representations preserve localized manipulation cues. We test this assumption with a token- and layer-wise probing diagnostic that uses pixel-level masks only for analysis to separate tampered-region and authentic-region vision tokens. Across three MLLM-based forgery backbones and multiple manipulation types, we identify a mid-layer forgery evidence peak: tampered-region tokens are most linearly separable at upper-middle decoder layers and become less separable afterward, whereas authentic-region tokens continue to strengthen. This region-conditioned asymmetry suggests competition between localized manipulation cues and late semantic aggregation. Motivated by this diagnostic, we propose Gated Forensic Memory (GFM), a lightweight evidence relay that reads from the diagnosed peak layer and writes re-encoded vision-token states back to late vision-token states through a zero-initialized gated residual. GFM keeps the LLM frozen, adds no extra vision tokens, and requires no pixel-level training supervision. Across three backbones and seven benchmarks, GFM improves authenticity prediction, explanation quality, and spatial alignment without mask supervision; the best read layer matches the diagnosed peak, turning the diagnostic into a practical layer-selection criterion. Code is available at https://github.com/laiyingxin2/GFM-Forensics.

1 Introduction

Multimodal large language models (MLLMs) [22, 21, 1, 51] are increasingly used for explainable image forgery analysis, where a shared decoder is expected to support authenticity prediction, spatial evidence alignment, and natural-language explanation. Prior work improves this setting through data construction, region-level modules, reasoning supervision, preference optimization, and forgery-specific rewards [41, 23, 33, 28, 50, 38, 10, 17, 14]. However, these efforts mainly change data, modules, or objectives, while leaving the decoder evidence-reading interface largely unchanged. Vision tokens are injected once, processed through the language decoder, and final decoder states are expected to support prediction, alignment, and explanation. This creates a potential bottleneck: sparse manipulation cues may become less readable at the final prediction stage.

*Equal contribution.
†Corresponding author: zhiming.luo@xmu.edu.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

This bottleneck is especially relevant to forgery analysis, where evidence is often sparse, local, and low-amplitude. Generic vision–language tasks often benefit from aggregating visual tokens into stable, language-aligned concepts. Forgery analysis instead depends on localized traces such as boundary discontinuities, texture and frequency artifacts, denoising traces, and inpainting artifacts [46, 2, 30, 8, 35, 49, 34, 6]. The late aggregation that improves semantic alignment may dilute or redistribute such pixel-grounded cues, making them less readable at their original token positions. Although recent layer-wise studies show that visual information varies across MLLM depth [48, 44, 18, 14, 19], the depth trajectory of forgery-discriminative evidence remains unclear.

To address this gap, we introduce a token- and layer-wise diagnostic that examines where forgery evidence is most readable within the MLLM. At each decoder layer, we probe vision-token states with a linear classifier, using pixel-level masks only for analysis to separate tampered-region and authentic-region tokens. This diagnostic yields three findings. First, the vision encoder is not forgery-specific: its attention concentrates on salient foregrounds rather than manipulation boundaries. Second, tampered-region tokens become most separable at upper-middle decoder layers and become less separable afterward, whereas authentic-region tokens strengthen toward the final layer. Third, logit-lens analysis shows that forgery-related vocabulary emerges in the same middle band during generation. We refer to this asymmetric trajectory as the *mid-layer forgery evidence peak*.

This asymmetry is difficult to explain by either global visual forgetting alone or monotonic evidence refinement. If late layers simply discarded visual detail, both tampered-region and authentic-region tokens would lose separability; if they monotonically refined forensic evidence, tampered-region separability would continue to rise. Instead, the two trajectories diverge, consistent with representational competition between localized manipulation cues and late semantic aggregation, possibly together with diffusion of local evidence across tokens. The diagnostic therefore suggests a targeted intervention: read evidence at the peak, preserve token-level specificity, and make the evidence available to late decision states.

We propose Gated Forensic Memory (GFM), a hidden-state evidence relay that implements this intervention while keeping the LLM frozen. GFM reads vision-token states at the diagnostic peak, routes and re-encodes them with lightweight MLPs, and writes them back to late vision-token states through a zero-initialized gated residual. This design differs from three simpler alternatives. Unlike final-layer classification heads, it modifies the hidden evidence path rather than only rescoring a late representation. Unlike output-side logit correction, it acts before late decoder states are used for generation. Unlike visual token pruning or retrieval-based memory, it does not remove tokens, add memory tokens, or retrieve external evidence; it keeps the token budget, decoder depth, and frozen LLM unchanged.

We instantiate GFM on FakeVLM and FakeShield as primary backbones, and on SIDA for cross-backbone validation. Across FakeClue, LOKI, DD-VQA, FF++, DMimage, MMTD-Set, and SID-Set, GFM improves authenticity prediction, explanation quality, and spatial evidence alignment. The best read layer matches the diagnostic peak, and the learned routing weights align with manipulation regions although pixel-level masks are never used for training. Our contributions are:

- We identify a *mid-layer forgery evidence peak*: tampered-region tokens peak at upper-middle decoder layers, while authentic-region tokens strengthen toward the final layer.
- We introduce GFM, a zero-initialized hidden-state evidence relay that keeps the LLM frozen, adds no vision tokens, and requires no pixel-level training supervision.
- We validate the diagnostic-to-design link: GFM performs best when reading at the diagnosed peak, improves late-layer tampered-region separability, and aligns routing weights with manipulation regions.
- Across three backbones and seven benchmarks, GFM improves authenticity prediction, explanation quality, spatial alignment, and cross-dataset transfer over representative forgery MLLM baselines.

2 Related Work

MLLM-based forgery analysis. Recent MLLM-based forgery analysis addresses authenticity prediction, evidence localization, and natural-language explanation. Existing work improves this setting through tampering-description data [41, 23, 50, 38, 20], mask-aware localization [23, 28],

open-world or face-forgery reasoning [33, 45, 11, 29, 32], and reward-based or retrieval-augmented reasoning [10, 9, 14, 13, 15, 36, 47, 17, 27, 40]. These methods mainly modify data, localization modules, or objectives. In contrast, we examine whether the decoder residual stream preserves forgery evidence, and modify the evidence-reading interface without changing the language head or requiring pixel-level training masks.

Layer-wise visual information flow in MLLMs. Interpretability and acceleration studies show that visual information varies across MLLM layers. Intermediate layers often mediate cross-modal transfer, whereas later layers become more language-oriented or visually redundant [48, 44, 4, 18, 39, 42]. Hallucination studies further link visual misalignment to intermediate and deep layers [19, 14]. We extend this line from generic visual content to sparse forgery cues, identifying a class-conditioned depth trajectory. The resulting diagnostic motivates hidden-state evidence relay rather than token pruning or decoding-time correction.

3 Layer-Wise Forgery Evidence Diagnostic

Probe setup. We consider explainable forgery analysis: given an image I and prompt q, the model generates an explanation R and an authenticity label $y \in \{0, 1\}$, where 1 denotes real and 0 fake. For annotated fake images, $M \in \{0, 1\}^{H \times W}$ denotes the manipulation mask. Masks are used only for diagnosis and evaluation, never for training GFM. A vision encoder and projector [21] embed I into N vision tokens, which are concatenated with text tokens to form the joint sequence $Z \in \mathbb{R}^{T \times D}$.

Let $\mathcal{V}(I) = \{v_1, \ldots, v_N\}$ denote the joint-sequence positions of vision tokens. The decoder has L layers indexed by $\ell \in \{0, \ldots, L-1\}$. We write $X^{(\ell)}=[x_1^{(\ell)}, \ldots, x_N^{(\ell)}] \in \mathbb{R}^{N \times D}$, where $x_i^{(\ell)}$ is the hidden state at position v_i. Explanation generation is trained on reference tokens $r_t^\star$ with

$$\mathcal{L}_{\mathrm{lm}} = -\sum_t \log p_\theta(r_t^\star \mid Z, r_{<t}^\star). \tag{1}$$

For fake images, we compute the fraction of manipulated pixels in each patch of the $h \times w$ grid aligned with the N vision tokens. We define token subsets $\mathcal{S}_f(I), \mathcal{S}_s(I) \subseteq [N]$: $i \in \mathcal{S}_f(I)$ if at least $\tau_{\mathrm{cov}}=0.5$ of token i's pixels are manipulated, and $i \in \mathcal{S}_s(I)$ otherwise.

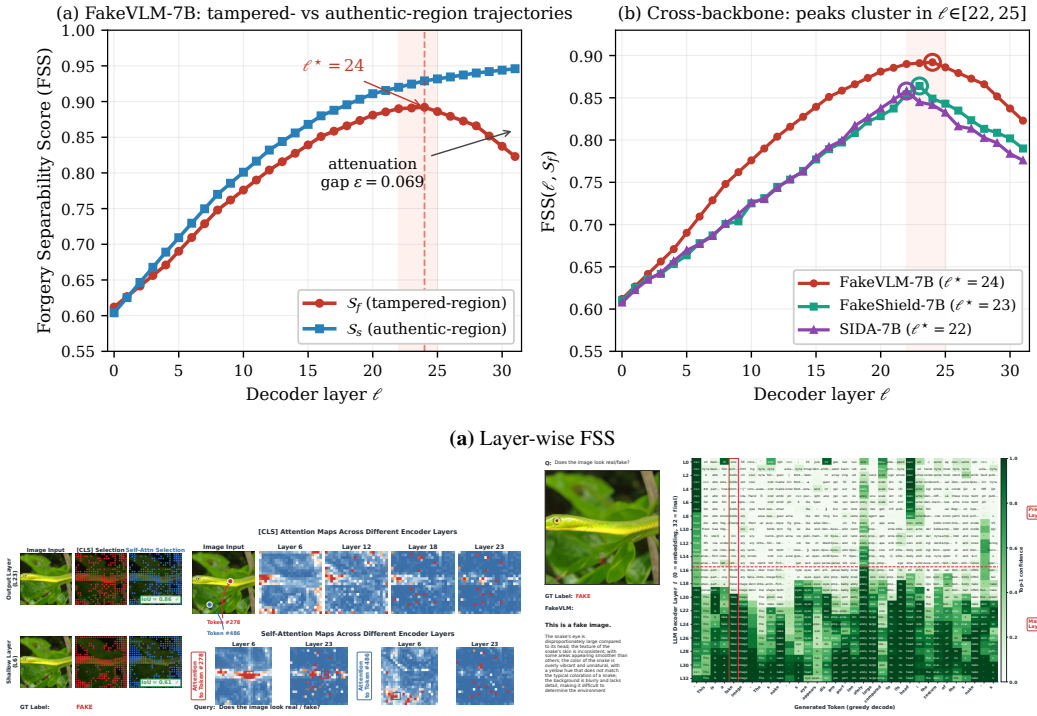
Layer-wise linear probe. At each decoder layer ℓ, we train an independent linear probe $\phi^{(\ell)} : \mathbb{R}^D \to \mathbb{R}$ on frozen hidden states with binary cross-entropy. During probe training, tampered-region tokens from fake images receive label 0, and tokens from real images receive label 1. Authentic-region tokens within fake images are excluded from probe training because their local content is genuine although their host image is fake. Each probe uses AdamW (learning rate 10^{-3}, weight decay 10^{-4}, 20 epochs, batch size 128); results are averaged over three seeds with image-level splits. Probes are offline diagnostics only: no probe parameter is shared with the main model, and no gradient flows back into the MLLM.

We measure layer-ℓ readability with the Forgery Separability Score (FSS):

$$\begin{aligned}\mathrm{FSS}(\ell, \mathcal{S}) = \mathrm{AUC}\Big(&\underbrace{\{\phi^{(\ell)}(x_i^{(\ell)}(I)) : y(I)=1,\, i \in [N]\}}_{\text{positives: real-image tokens}}, \\ &\underbrace{\{\phi^{(\ell)}(x_i^{(\ell)}(I)) : y(I)=0,\, i \in \mathcal{S}(I)\}}_{\text{negatives: selected fake-image tokens}}\Big).\end{aligned} \tag{2}$$

Thus, $\mathrm{FSS}(\ell, \mathcal{S})$ measures how linearly distinguishable tokens in $\mathcal{S}$ are from real-image tokens at depth ℓ. For $\mathcal{S}_s$, FSS is evaluation-only: authentic-region tokens from fake images are scored but not used to train the probe. FSS is not a training objective for GFM.

Finding 1: the vision encoder is not forgery-specific. Figure 1(b) shows that both `[CLS]`-attention and self-attention concentrate on salient foregrounds rather than manipulation boundaries. Thus, we examine where forgery-discriminative evidence becomes readable inside the decoder.



(a) Layer-wise FSS

(b) Encoder attention **(c)** Logit-lens trajectory

Figure 1: Diagnostic evidence for the mid-layer forgery evidence peak. Encoder attention is not forgery-specific; tampered-region FSS peaks in the upper-middle decoder and attenuates later; and logit-lens trajectories show forgery-related vocabulary emerging in the same middle band.

Finding 2: forgery evidence peaks at upper-middle decoder layers. Applying the FSS probe across FakeVLM, FakeShield, and SIDA reveals asymmetric depth trajectories (Fig. 1(a)). We summarize this pattern by a peak layer $\ell^\star$ and attenuation gap

$$\underbrace{\mathrm{FSS}(L{-}1, \mathcal{S}_f) \;\le\; \mathrm{FSS}(\ell^\star, \mathcal{S}_f) - \Delta_{\mathrm{att}},}_{\text{tampered-region tokens attenuate after } \ell^\star} \qquad \underbrace{\mathrm{FSS}(L{-}1, \mathcal{S}_s) \;\ge\; \mathrm{FSS}(\ell^\star, \mathcal{S}_s) + \Delta_{\mathrm{gain}},}_{\text{authentic-region tokens strengthen after } \ell^\star} \tag{3}$$

Late layers therefore increase authentic-region separability but reduce tampered-region readability. This divergence is consistent with representational competition, possibly together with diffusion of local evidence across tokens. The strong final-layer $\mathcal{S}_s$ score suggests that visual evidence is not globally discarded; rather, local manipulation cues become less readable at their original token positions.

Finding 3: the peak is visible during generation. A per-token logit-lens analysis shows a consistent generation-time pattern (Fig. 1(c)): forgery-related vocabulary becomes prominent in the same middle band. This suggests that lexical forgery cues can emerge before the final answer, while their local visual support becomes less readable in late states. This motivates hidden-state write-back rather than output-only correction.

Design implications. Across all six backbone–manipulation configurations, the empirical peak $\ell^\star$ falls in layers 22–25 of 32, with $\Delta_{\mathrm{att}} \in [0.069, 0.085]$. These findings suggest three design requirements. **(D1) Read at the peak**, because final-state reading can miss tampered-region evidence. **(D2) Operate per token**, because tampered and authentic regions can coexist within one image. **(D3) Write into the hidden stream**, because output-side correction does not directly restore local visual evidence.

4 Gated Forensic Memory

GFM implements the diagnostic requirements as a lightweight hidden-state evidence relay. We use the notation from Section 3: N is the number of vision tokens, D is the hidden dimension, and

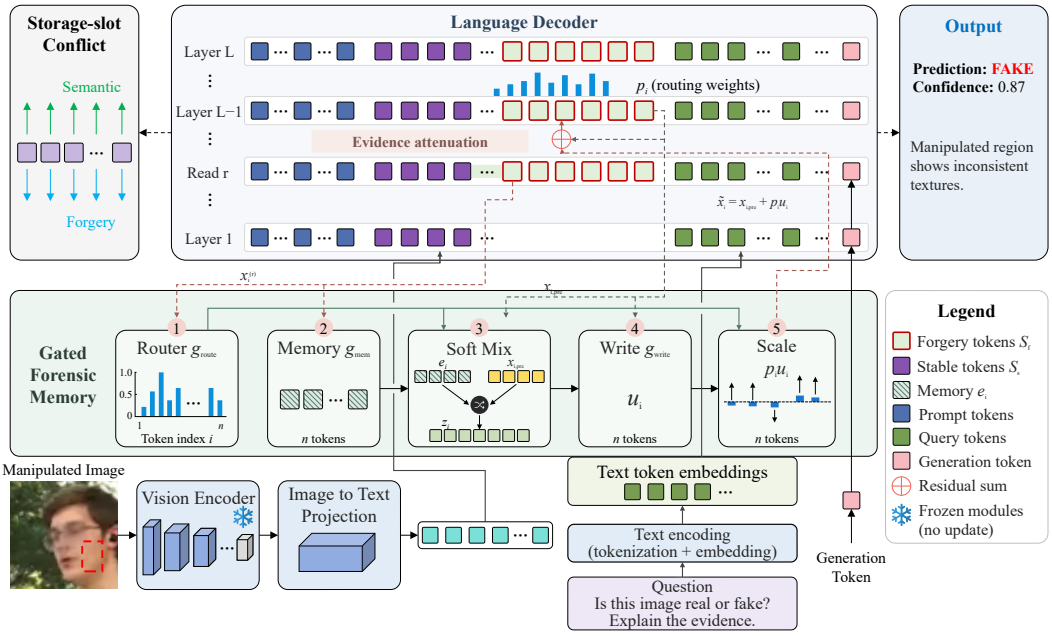


Figure 2: GFM overview. At $r = \ell^\star$, GFM reads vision-token states, routes and re-encodes them, and writes a gated residual before the final decoder block $\ell_w = L-1$. The LLM is frozen; only the projector, GFM modules, and auxiliary heads are trained.

$X^{(\ell)} = [x_1^{(\ell)}, \ldots, x_N^{(\ell)}] \in \mathbb{R}^{N \times D}$ denotes the vision-token states at decoder layer ℓ. GFM adds no memory tokens, uses no retrieval, and keeps the LLM frozen. Figure 2 shows the design. At read layer $r = \ell^\star$, GFM reads $X^{(r)}$, routes and re-encodes each vision token, and writes the result before the final decoder block $\ell_w = L-1$. Decoder depth, LLM parameters, vision-token count, and autoregressive generation are unchanged.

The design follows three constraints from Section 3. First, because tampered-region FSS peaks before the final layer, GFM reads from $r = \ell^\star$ rather than from the final hidden state. Second, because forged and authentic regions can coexist in one image, routing is token-level rather than image-level. Third, because late states remain the interface used for final prediction and explanation, the selected evidence is written back into late vision-token states instead of being used only by an auxiliary classifier.

Token-level evidence routing. Because the evidence peak is token-local, GFM routes each vision token independently. A two-layer router $g_{\text{route}} : \mathbb{R}^D \to \mathbb{R}$ with LayerNorm and GELU produces a routing probability

$$p_i = \sigma\left(g_{\text{route}}(x_i^{(r)})/\tau_g\right), \qquad \tau_g = 1, \tag{4}$$

where τ_g is the gate temperature. The scalar $p_i \in (0, 1)$ controls how strongly token i uses the mid-layer evidence path. We use continuous routing rather than discrete top-k selection because forgery cues can be weak, spatially diffuse, or partially mixed with authentic content. A hard threshold may discard borderline evidence, whereas a soft gate preserves uncertainty and enables the model to learn graded evidence strength.

Forensic memory encoding. Mid-layer states are mapped into a writable form because decoder depths can use different representational bases. A two-layer memory encoder $g_{\text{mem}} : \mathbb{R}^D \to \mathbb{R}^D$ maps each read-layer state to an evidence state

$$e_i = g_{\text{mem}}(x_i^{(r)}) \in \mathbb{R}^D. \tag{5}$$

The router and memory encoder have disjoint parameters: p_i controls how strongly the evidence path is used, and e_i determines what content is written. This separation keeps routing decisions distinct from representation conversion and prevents the gate from serving only as a generic feature-strength score.

Residual write-back. Let $x_{i,\mathrm{pre}}^{(\ell_\mathrm{w})}$ denote the vision-token state at position i immediately before the write layer $\ell_\mathrm{w} = L-1$. For each token, GFM forms a gate-weighted mixture, predicts a residual update, and writes the post-injection state:

$$z_i = p_i e_i + (1-p_i) x_{i,\mathrm{pre}}^{(\ell_\mathrm{w})}, \qquad u_i = g_\mathrm{write}\Big([z_i;\, x_{i,\mathrm{pre}}^{(\ell_\mathrm{w})}]\Big), \qquad \tilde{x}_i = x_{i,\mathrm{pre}}^{(\ell_\mathrm{w})} + p_i u_i, \tag{6}$$

where $g_\mathrm{write} : \mathbb{R}^{2D} \to \mathbb{R}^D$ and $\widetilde{X}^{(\ell_\mathrm{w})} = [\tilde{x}_1, \ldots, \tilde{x}_N]$ denotes the post-injection vision-token states. Only vision-token states are updated; text-token states are unchanged. The mixture z_i preserves late-layer context when p_i is small and emphasizes mid-layer evidence when p_i is large. The residual form avoids directly replacing late states, which can remove semantic context needed for explanation generation.

The final linear layer of g_write is zero-initialized, so $u_i = 0$ and $\tilde{x}_i = x_{i,\mathrm{pre}}^{(\ell_\mathrm{w})}$ at initialization. Thus, GFM starts as an identity map on the frozen LLM. This initialization stabilizes optimization: before learning a reliable route, the model behaves like the original backbone; during training, the gated residual learns where and how much mid-layer evidence should be injected.

Training signals. The LM loss weakly supervises routing, so we add two binary image-level heads $c_\mathrm{early}, c_\mathrm{final} : \mathbb{R}^D \to \mathbb{R}^2$, together with two routing regularizers. Let $y \in \{0, 1\}$ denote the image-level authenticity label, where 1 denotes real and 0 fake. The early head supervises gated mid-layer evidence, while the final head supervises post-injection states:

$$h_\mathrm{early} = \frac{\sum_i p_i e_i}{\sum_i p_i + \epsilon_\mathrm{pool}}, \qquad \mathcal{L}_\mathrm{early} = \mathrm{CE}(c_\mathrm{early}(h_\mathrm{early}), y)\,, \tag{7}$$

$$h_\mathrm{final} = \frac{1}{N}\sum_i \tilde{x}_i, \qquad \mathcal{L}_\mathrm{final} = \mathrm{CE}(c_\mathrm{final}(h_\mathrm{final}), y)\,, \tag{8}$$

where $\epsilon_\mathrm{pool} = 10^{-6}$. The early head encourages the router and memory encoder to preserve image-level authenticity evidence before write-back, while the final head supervises the post-injection states before the final decoder block. Gradients from $\mathcal{L}_\mathrm{final}$ flow through $\tilde{x}_i$ to all GFM modules.

We minimize the gate-entropy penalty

$$\mathcal{L}_\mathrm{conf} = \frac{1}{N}\sum_i -\big[p_i \log p_i + (1-p_i)\log(1-p_i)\big] \tag{9}$$

to discourage diffuse routing, and use a label-conditional gate-rate prior

$$\mathcal{L}_\mathrm{ratio} = (\bar{p} - \rho(y))^2, \qquad \bar{p} = \frac{1}{N}\sum_i p_i, \qquad \rho(y) = \rho_\mathrm{fake}\mathbf{1}[y=0]. \tag{10}$$

This prior encourages a lower average gate on real images and allows a higher average gate on fake images, where localized manipulated regions may require mid-layer evidence relay. We set $\rho_\mathrm{fake} = 0.3$ by default.

The auxiliary loss is

$$\mathcal{L}_\mathrm{aux} = \lambda\left(\mathcal{L}_\mathrm{early} + \mathcal{L}_\mathrm{final} + \beta_\mathrm{conf}\mathcal{L}_\mathrm{conf} + \beta_\mathrm{ratio}\mathcal{L}_\mathrm{ratio}\right), \tag{11}$$

with $\lambda = 0.1$ and $\beta_\mathrm{conf} = \beta_\mathrm{ratio} = 0.1$. We optimize GFM modules and auxiliary heads with AdamW at 2×10^{-4} and the projector at 5×10^{-5}, using an effective batch size of 64.

Training and inference. The total objective is

$$\mathcal{L} = \mathcal{L}_\mathrm{lm} + \mathcal{L}_\mathrm{aux}, \tag{12}$$

computed on post-injection hidden states. Samples without image-level authenticity labels contribute only $\mathcal{L}_\mathrm{lm}$. Pixel-level masks never enter training; any alignment between routing weights p_i and tampered regions is evaluated only in Section 5.

At inference time, GFM performs a single additional route–encode–write operation over the existing vision tokens. It does not change the prompt, add visual tokens, retrieve external memory, or alter the decoding procedure. This keeps the intervention compatible with existing MLLM-based forgery detectors: one selects the read layer r from the offline diagnostic, inserts write-back before ℓ_w, and runs the original autoregressive generation path.

5 Experiments

Setup. We instantiate GFM on FakeVLM-7B [38] and FakeShield-7B [41] as primary backbones, and on SIDA-7B [12] and SIDA-13B for cross-scale validation. The LLM is frozen for all backbones; only the projector, GFM modules, and auxiliary heads are trained. We evaluate detection and explanation on FakeClue [38] and LOKI [43], spatial alignment (without mask supervision during training) on six distributions including FF++ [31], multi-type tampering on MMTD-Set [41], and SID-Set [12] splits with SIDA.

Metrics and baselines. Detection uses accuracy (Acc) and F1, explanation uses ROUGE-L (R-L) and CSS (CLIP-based semantic similarity), and spatial alignment uses Routing-Mask IoU. Routing-Mask IoU thresholds routing probabilities or proxy evidence maps at $\tau_{\text{bin}} = 0.5$, upsamples them to image resolution, and computes IoU against the ground-truth manipulation mask, which we use only at evaluation time. Non-routing baselines use normalized vision-token evidence scores as proxy maps under the same protocol. We compare with generic MLLMs, specialist forgery classifiers, and MLLM-based forgery detectors.

Intervention baselines and hyperparameters. To isolate the read-and-write mechanism, we compare with three matched-budget interventions. Final-layer logit correction trains a two-layer MLP on the final language state. Late-layer detection head trains a classifier on mean-pooled final vision-token states. Attention rerouting rescales late-layer attention toward tokens predicted as forgery-relevant by a lightweight image-level scorer trained on the same labels as GFM, without adding decoder parameters. All interventions share data, schedule, and inference budget with GFM. Unless otherwise specified, $r = \ell^\star$, $\ell_{\text{w}} = L-1$, $\lambda = 0.1$, $\beta_{\text{conf}} = \beta_{\text{ratio}} = 0.1$, and $\rho_{\text{fake}} = 0.3$.

Read-layer validation and robustness. Figure 3 sweeps the read layer r on FakeVLM-7B / LOKI. Acc, F1, ROUGE-L, and Routing-Mask IoU all peak at $r = 24$, which coincides with the diagnostic peak $\ell^\star = 24$ for this backbone. This supports using the offline FSS probe as a practical criterion for selecting the read layer. Figure 4 evaluates robustness under JPEG compression, reduced input resolution, and additive Gaussian noise. GFM remains above FakeVLM across all tested perturbations, with the gap widening as corruption severity increases — consistent with mid-layer evidence relay being most useful when late-layer evidence is less reliable.

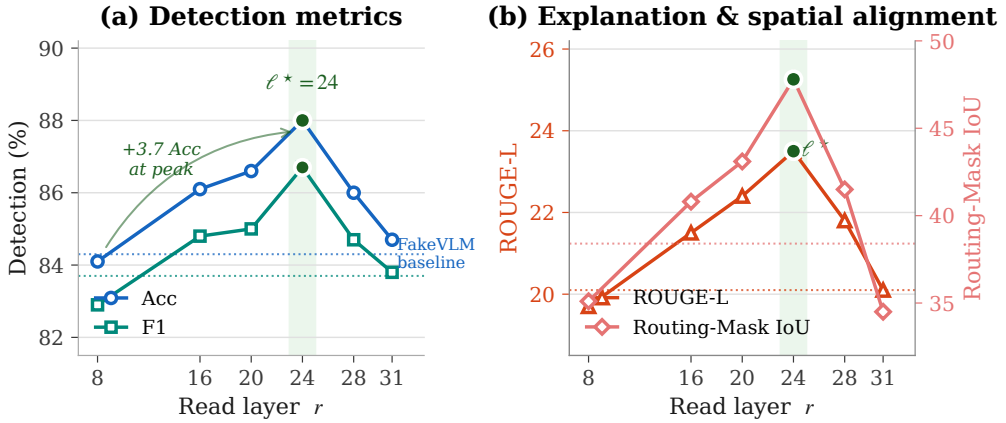


Figure 3: Read-layer validation on FakeVLM-7B / LOKI. Detection, explanation, and routing-alignment metrics peak at $r = 24$.

Main results: FakeVLM backbone. Table 1 reports detection and explanation results on FakeClue and LOKI. GFM improves the FakeVLM backbone on both in-domain and cross-generator transfer settings, with larger gains on the cross-generator LOKI split (+3.7 Acc, +4.2 CSS) than on the in-domain FakeClue split (+0.8 Acc, +1.4 CSS). The larger transfer gain suggests that mid-layer evidence relay is most beneficial when final-layer representations are not yet adapted to the target generator distribution.

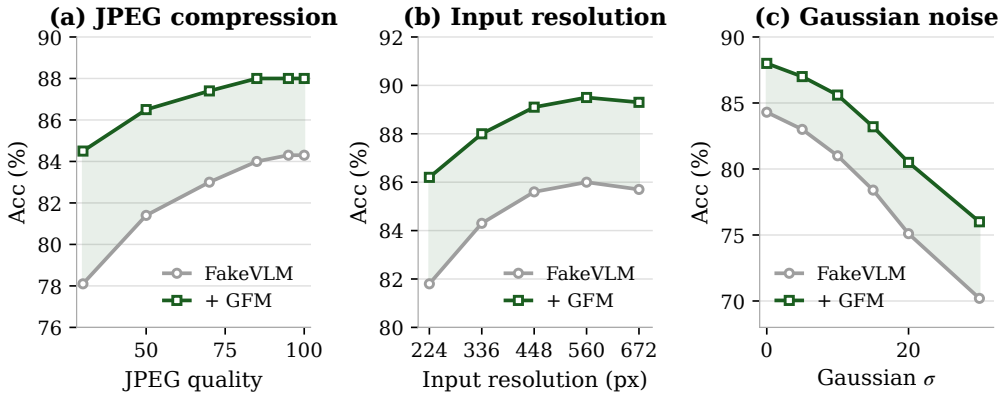


Figure 4: Robustness on FakeVLM-7B / LOKI. GFM remains above FakeVLM under input corruptions.

Table 1: FakeVLM results on FakeClue and LOKI. Arrows report changes over FakeVLM. * denotes training on FakeClue.

Method	FakeClue (in-domain)				LOKI (cross-generator transfer)			
	Acc	F1	R-L	CSS	Acc	F1	R-L	CSS
Qwen2-VL-72B [37]	57.8	56.5	17.5	54.4	55.4	40.9	17.3	43.2
NPR* [35]	90.2	91.6	–	–	77.4	80.0	–	–
FakeVLM [38]	98.6	98.1	58.0	87.7	84.3	83.7	20.1	58.2
FakeVLM + GFM	**99.4** ↑0.8	**99.0** ↑0.9	**60.5** ↑2.5	**89.1** ↑1.4	**88.0** ↑3.7	**86.7** ↑3.0	**23.5** ↑3.4	**62.4** ↑4.2

Main results: FakeShield backbone. Table 2 reports detection results on MMTD-Set across seven manipulation subsets. GFM improves Acc over FakeShield on all subsets and F1 on five subsets. The largest gain appears on IMD2020 ($+0.04$ Acc, $+0.02$ F1), with $+0.02$ Acc gains also on CASIA1+ and DeepFake.

Table 2: FakeShield results on MMTD-Set. Arrows on the second line report ΔAcc / ΔF1.

Method	CASIA1+ Acc / F1	IMD2020 Acc / F1	Columbia Acc / F1	Coverage Acc / F1	DSO Acc / F1	DeepFake Acc / F1	AIGC-Edit Acc / F1
PSCC-Net [24]	0.90/0.89	0.67/0.78	0.78/0.87	0.84/0.91	0.66/0.80	0.48/0.58	0.49/0.65
CAT-Net [16]	0.88/0.87	0.68/0.79	0.89/0.94	0.23/0.37	0.86/0.92	0.85/0.84	0.82/0.81
MVSS-Net [7]	0.62/0.76	0.75/0.85	0.94/0.97	0.65/0.79	0.96/0.98	0.84/0.91	0.44/0.24
FakeShield [41]	0.95/0.95	0.83/0.90	0.98/0.99	0.97/0.98	0.97/0.98	0.93/0.93	0.98/0.99
FakeShield + GFM	**0.97/0.97**	**0.87/0.92**	**0.99/0.99**	**0.98/0.99**	**0.98/0.99**	**0.95/0.95**	**0.99/0.99**
	↑0.02/0.02	↑**0.04/0.02**	↑0.01/0.00	↑0.01/0.01	↑0.01/0.01	↑0.02/0.02	↑0.01/0.00

Main results: SIDA backbone. Table 3 reports results on SID-Set across the Real, Fully synthetic, and Tampered splits. The largest gain appears on the Tampered split ($+2.3$ Acc on SIDA-7B), while the Fully synthetic split shows a smaller gain ($+0.4$ Acc); the same trend holds on SIDA-13B. This pattern is consistent with our diagnostic: localized manipulations benefit most from mid-layer evidence relay, whereas globally synthetic images already give late-layer states a strong signal.

Additional transfer and overhead. Table 4 reports additional transfer results and inference overhead on FakeVLM-7B. GFM improves DD-VQA detection and explanation, and improves DMimage cross-generator detection. It introduces no extra vision tokens and adds only a small inference overhead in our implementation. Together, these results indicate that GFM is a lightweight and transferable evidence relay.

Spatial evidence alignment. Figure 5 and Table 5 evaluate whether the learned routing weights p_i align with pixel-level manipulation regions; pixel-level masks are used only for evaluation, never during training. The GFM router exceeds the non-routing baselines by $8.8-10.2$ IoU points across six distributions, indicating that routing captures spatially localized manipulation cues without mask supervision.

Table 3: SIDA results on SID-Set. Arrows report ΔAcc / ΔF1.

Method	Real Acc / F1	Fully synthetic Acc / F1	Tampered Acc / F1	Overall Acc / F1
AntifakePrompt [3]	64.8 / 78.6	93.8 / 96.8	30.8 / 47.2	63.1 / 69.3
UnivFD [26]	68.0 / 67.4	62.1 / 87.5	64.0 / 85.3	64.0 / 85.3
Gram-Net [25]	70.1 / 82.4	93.5 / 96.6	0.8 / 1.6	55.0 / 58.0
SIDA-7B [12]	89.1 / 91.0	98.7 / 98.6	91.2 / 91.0	93.5 / 93.5
SIDA-7B + GFM	**90.4 / 92.1** ↑1.3 / 1.1	**99.1 / 99.0** ↑0.4 / 0.4	**93.5 / 93.2** ↑2.3 / 2.2	**94.6 / 94.7** ↑1.1 / 1.2
SIDA-13B [12]	89.6 / 91.1	98.5 / 98.7	92.9 / 91.2	93.6 / 93.5
SIDA-13B + GFM	**91.0 / 92.4** ↑1.4 / 1.3	**99.0 / 99.2** ↑0.5 / 0.5	**94.8 / 93.7** ↑1.9 / 2.5	**94.9 / 95.0** ↑1.3 / 1.5

Table 4: Plug-in transfer and inference overhead on FakeVLM-7B. Arrows in (a) denote absolute change versus the FakeVLM baseline. Resource cost in (b) is measured on a single NVIDIA H100 GPU with FP16 weights and a 448×448 input, averaged over 1,000 inference passes.

(a) Transfer to held-out forgery benchmarks

Dataset	Method	Acc	F1	R-L
DD-VQA [47]	FakeVLM	86.4	85.7	27.2
	+ GFM	**88.5** ↑2.1	**87.3** ↑1.6	**29.0** ↑1.8
DMimage [5]	FakeVLM	83.9	83.0	–
	+ GFM	**87.2** ↑3.3	**86.0** ↑3.0	–
LOKI [43]	FakeVLM	84.3	83.7	20.1
	+ GFM	**88.0** ↑3.7	**86.7** ↑3.0	**23.5** ↑3.4

(b) Modeling and inference cost

	Absolute	Rel.
Trainable parameters		
Router g_{route}	4.2 M	0.06%
Memory encoder g_{mem}	8.1 M	0.11%
Write-back g_{write}	6.7 M	0.09%
Aux. heads + projector	2.0 M	0.03%
total / 7 B backbone	**21.0 M**	**0.29%**
Per-image overhead		
Forward FLOPs	+0.31 T	+1.9%
Latency (H100, FP16)	+11.7 ms	+2.7%
Peak memory	+0.42 GB	+1.6%
Extra vision tokens	0	—

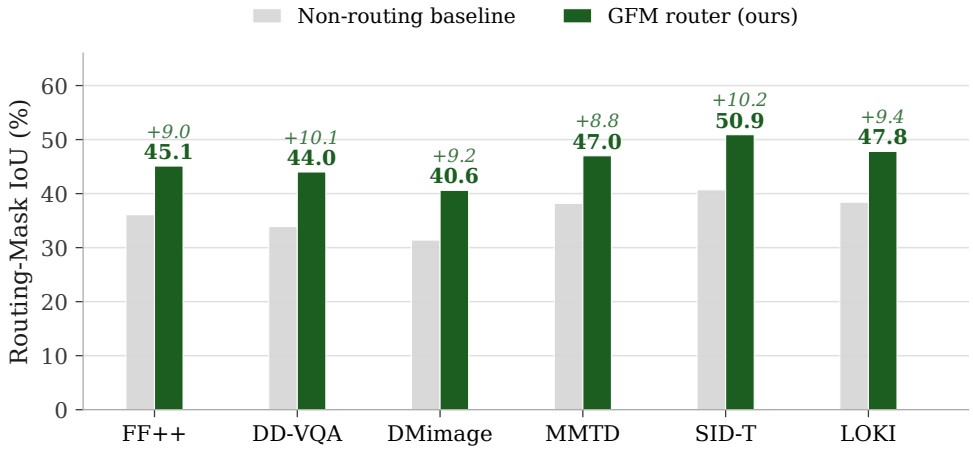


Figure 5: Routing-mask alignment across six distributions. Results from Table 5; pixel-level masks are used only for evaluation.

Comparison to mechanism baselines. Table 6 compares GFM with the matched-budget intervention baselines. GFM improves all reported metrics and yields the largest gain on Routing-Mask IoU. Logit correction and the late-layer detection head produce smaller gains, likely because they rescore late representations without modifying the hidden-state evidence path. Attention rerouting outperforms these output-side baselines but lacks a learned encoder for mapping mid-layer evidence into the late residual stream. This comparison supports the necessity of both reading from the diagnostic peak and writing evidence back into late hidden states.

Table 5: Per-distribution Routing-Mask IoU at $\tau_{\text{bin}} = 0.5$. The LOKI baseline is FakeVLM without intervention (Table 6).

Method	FF++	DD-VQA	DMimage	MMTD	SID-T	LOKI
Non-routing baseline	36.1	33.9	31.4	38.2	40.7	38.4
GFM router	**45.1** ↑9.0	**44.0** ↑10.1	**40.6** ↑9.2	**47.0** ↑8.8	**50.9** ↑10.2	**47.8** ↑9.4

Table 6: Mechanism comparison on FakeVLM-7B / LOKI. Arrows report changes over FakeVLM.

Intervention	Acc	F1	ROUGE-L	Routing-Mask IoU
FakeVLM (no intervention)	84.3	83.7	20.1	38.4
+ Final-layer logit correction	85.1 ↑0.8	84.0 ↑0.3	20.4 ↑0.3	38.9 ↑0.5
+ Late-layer detection head	85.6 ↑1.3	84.5 ↑0.8	20.6 ↑0.5	39.2 ↑0.8
+ Attention rerouting	86.4 ↑2.1	85.2 ↑1.5	21.7 ↑1.6	40.1 ↑1.7
+ GFM	**88.0** ↑3.7	**86.7** ↑3.0	**23.5** ↑3.4	**47.8** ↑9.4

Component ablation. Table 7 ablates GFM's main design choices on FakeVLM-7B / LOKI. Removing token-level routing, the memory encoder, the soft residual write-back, or zero-initialization degrades performance. Hard write-back $\tilde{x}_i = e_i$ causes the largest drop, indicating that directly overwriting late states erases useful late-layer context. Auxiliary losses contribute: removing $\mathcal{L}_{\text{early}}$ reduces Routing-Mask IoU most, indicating that supervision on the routed evidence pool most strongly influences routing alignment.

Table 7: Component ablation on FakeVLM-7B / LOKI. Arrows report changes relative to full GFM.

Variant	Acc	F1	ROUGE-L	Routing-Mask IoU
Image-level scalar routing	86.2 ↓1.8	84.9 ↓1.8	22.2 ↓1.3	41.2 ↓6.6
Discrete top-k routing	85.9 ↓2.1	84.4 ↓2.3	21.7 ↓1.8	38.5 ↓9.3
No memory encoder	85.1 ↓2.9	83.6 ↓3.1	20.8 ↓2.7	35.8 ↓12.0
Hard write-back $\tilde{x}_i = e_i$	84.0 ↓4.0	82.5 ↓4.2	19.6 ↓3.9	32.4 ↓15.4
No g_{write} zero-init	86.4 ↓1.6	85.2 ↓1.5	22.6 ↓0.9	45.1 ↓2.7
No $\mathcal{L}_{\text{early}}$	85.7 ↓2.3	84.3 ↓2.4	21.9 ↓1.6	40.6 ↓7.2
No $\mathcal{L}_{\text{final}}$	86.7 ↓1.3	85.3 ↓1.4	22.8 ↓0.7	46.3 ↓1.5
No $\mathcal{L}_{\text{conf}}$	87.0 ↓1.0	85.7 ↓1.0	23.0 ↓0.5	41.9 ↓5.9
No $\mathcal{L}_{\text{ratio}}$	87.1 ↓0.9	85.8 ↓0.9	23.1 ↓0.4	44.5 ↓3.3
Full GFM	**88.0**	**86.7**	**23.5**	**47.8**

Post-intervention diagnostic. Finally, we re-train the layer-wise FSS probe on the post-intervention model with GFM added to FakeVLM (the original probe cannot be reused, since hidden states have changed). Final-layer tampered FSS rises from 0.823 to 0.882 ($+0.059$), the attenuation gap shrinks from Δ_{att}=0.069 to 0.010, and final-layer authentic FSS remains nearly unchanged ($0.946 \rightarrow 0.943$). These changes indicate that GFM improves late-layer readability of tampered-region evidence while preserving authentic-region separability.

6 Conclusion

We identified a mid-layer forgery evidence peak in MLLM-based forgery analysis: tampered-region vision tokens are most linearly separable from real-image tokens at upper-middle decoder layers and become less separable afterward, while authentic-region separability improves toward the final layer. This class-conditioned asymmetry appears across logit-lens analyses and three MLLM-based forgery backbones, supporting an interpretation based on representational competition between localized forgery cues and language-oriented semantic aggregation.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China under Grant Nos. 62276221 and 62376232.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

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

[3] You-Ming Chang, Chen Yeh, Wei-Chen Chiu, and Ning Yu. Antifakeprompt: Prompt-tuned vision-language models are fake image detectors. *arXiv preprint arXiv:2310.17419*, 2023.

[4] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, 2024.

[5] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

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

[7] Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao, and Xirong Li. MVSS-Net: Multi-view multi-scale supervised networks for image manipulation detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3539–3553, 2022.

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

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

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

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

[12] Zhenglin Huang, Jinwei Hu, Xiangtai Li, Yiwei He, Xingyu Zhao, Bei Peng, Baoyuan Wu, Xiaowei Huang, and Guangliang Cheng. Sida: Social media image deepfake detection, localization and explanation with large multimodal model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28831–28841, 2025.

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

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

[15] Hengrui Kang, Siwei Wen, Zichen Wen, Junyan Ye, Weijia Li, Peilin Feng, Baichuan Zhou, Bin Wang, Dahua Lin, Linfeng Zhang, et al. Legion: Learning to ground and explain for synthetic image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18937–18947, 2025.

[16] Myung-Joon Kwon, In-Jae Yu, Seung-Hun Nam, and Heung-Kyu Lee. CAT-Net: Compression artifact tracing network for detection and localization of image splicing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

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

[18] Yi Li, Hualiang Wang, Xinpeng Ding, Haonan Wang, and Xiaomeng Li. Token activation map to visually explain multimodal llms. *ArXiv*, abs/2506.23270, 2025.

[19] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji rong Wen. Evaluating object hallucination in large vision-language models. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

[20] Yixuan Li, Yu Tian, Yipo Huang, Wei Lu, Shiqi Wang, Weisi Lin, and Anderson Rocha. Fakescope: Large multimodal expert model for transparent ai-generated image forensics. *arXiv preprint arXiv:2503.24267*, 2025.

[21] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. arXiv:2310.03744, 2023.

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[23] Jiawei Liu, Fanrui Zhang, Jiaying Zhu, Esther Sun, Qiang Zhang, and Zheng-Jun Zha. Forgerygpt: Multimodal large language model for explainable image forgery detection and localization. *arXiv preprint arXiv:2410.10238*, 2024.

[24] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. PSCC-Net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7505–7517, 2022.

[25] Zhengzhe Liu et al. Global texture enhancement for fake face detection in the wild. In *CVPR*, pages 8060–8069, 2020.

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

[27] Kyoungjun Park, Yifan Yang, Juheon Yi, Shicheng Zheng, Yifei Shen, Dongqi Han, Caihua Shan, Muhammad Muaz, and Lili Qiu. Vidguard-r1: Ai-generated video detection and explanation via reasoning mllms and rl. *arXiv preprint arXiv:2510.02282*, 2025.

[28] Rongxuan Peng, Shunquan Tan, Chenqi Kong, Anwei Luo, Alex C Kot, and Jiwu Huang. Forensicssam: Toward robust and unified image forgery detection and localization resisting to adversarial attack. *arXiv preprint arXiv:2508.07402*, 2025.

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

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

[31] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *International Conference on Computer Vision (ICCV)*, 2019.

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

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

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

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

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

[37] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

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

[39] Long Xing, Qidong Huang, Xiao wen Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *ArXiv*, abs/2410.17247, 2024.

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

[41] Zhipei Xu, Xuanyu Zhang, Runyi Li, Zecheng Tang, Qing Huang, and Jian Zhang. Fakeshield: Explainable image forgery detection and localization via multi-modal large language models. *arXiv preprint arXiv:2410.02761*, 2024.

[42] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19792–19802, 2025.

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

[44] Hao Yin, Guangzong Si, and Zilei Wang. Lifting the veil on visual information flow in mllms: Unlocking pathways to faster inference. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9382–9391, 2025.

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

[46] Lingzhi Zhang, Zhengjie Xu, Connelly Barnes, Yuqian Zhou, Qing Liu, He Zhang, Sohrab Amirghodsi, Zhe Lin, Eli Shechtman, and Jianbo Shi. Perceptual artifacts localization for image synthesis tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7579–7590, 2023.

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

[48] Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-modal information flow in multimodal large language models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19781–19791, 2025.

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

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

[51] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the diagnostic findings and GFM contribution supported by Sections 3–5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [No]

   Justification: The submitted main paper does not contain a dedicated discussion of limitations.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: This paper focuses on empirical diagnostics and architectural interventions for MLLMs rather than deriving new theoretical proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [No]

Justification: Sections 3–5 describe the diagnostic, GFM architecture, objectives, and main hyperparameters, but a complete experiment-specific reproduction configuration is not provided.

Guidelines:

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

5. **Open access to data and code**

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

   Answer: [No]

   Justification: The public repository linked in the abstract provides code, dependencies, model and data links, and launch commands. Complete reproduction instructions for all evaluated backbones and benchmarks are not provided.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [No]

   Justification: Sections 4 and 5 specify the core GFM settings, optimizer, learning rates, and batch size, but not complete per-backbone training schedules.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The paper reports point estimates without error bars, confidence intervals, or statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: Table 4 reports inference hardware and overhead; per-experiment training resources and total compute are not reported.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research aims to improve the detection and explanation of digital forgeries, directly supporting media integrity and conforming entirely to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

    Justification: The paper does not include a discussion of potential negative societal impacts.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The work introduces a forgery-detection module, not a new image generator or scraped dataset.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [No]

    Justification: The backbones and datasets are cited, but their license names and terms of use are not explicitly listed in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The public repository documents the GFM and probe code, dependencies, data format, and probe, training, and evaluation entry points.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: Our research relies entirely on existing datasets and does not involve any human subject studies or crowdsourcing.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve human subjects research, hence IRB approval is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: Sections 3–5 describe the MLLM backbones, layer-wise probing, and GFM hidden-state intervention used in the core method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.