LANGUAGE PRIORS WITHOUT LANGUAGE: A FROZEN LLM LAYER FOR GENERATED IMAGE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generated image detection increasingly relies on frozen vision-language models such as CLIP, whose features generalize well across generators but are learned through global image-text alignment and remain limited in modeling semantic relationships among image regions and structural consistency. As modern generators suppress low-level artifacts, high-level cues such as semantic and structural anomalies become decisive. We observe that large language models (LLMs) acquire, through autoregressive pretraining, precisely the ability that CLIP lacks: organizing semantic relationships and contextual dependencies among tokens. We therefore propose to treat a single frozen decoder layer of a pretrained LLM as a semantic relation enhancement module for visual tokens. The layer re-encodes the CLIP tokens within the relational structure learned from language, without any language prompts or textual inputs, while both CLIP and the LLM remain frozen and only about 4.3M parameters are trained. Trained on a single generator, our detector reaches 91.55% accuracy and 99.38% average precision across 13 GAN and diffusion generators, surpassing the strongest prior detector by 3.05 accuracy points, and it generalizes to recently released diffusion and autoregressive generators with 98.43% accuracy.

1 INTRODUCTION

Modern generative models (Wu et al., 2025; Tian et al., 2024; Esser et al., 2024; Labs, 2024) now synthesize images that are hard to distinguish from real photographs, posing growing risks to content authenticity, copyright protection, and online information governance (Wen et al., 2026; Yuan et al., 2026). Reliable generated image detection has therefore become essential for curbing misinformation and supporting content moderation.

Since new generators emerge continuously, a practical detector must generalize to generators unseen during training, yet existing detectors degrade sharply in this setting (Wang et al., 2020; Liu et al., 2022; Tan et al., 2023; 2024; Zhu et al., 2024). Contrastive vision-language models such as CLIP (Radford et al., 2021) alleviate this problem and have become the dominant backbone of generalizable detectors (Ojha et al., 2023; Cozzolino et al., 2024; Khan & Dang-Nguyen, 2024; Tan et al., 2025; Zhang et al., 2025; Yan et al., 2024). However, CLIP is trained by global image–text alignment and tends to behave like a bag of words, with limited relational modeling among objects and regions (Thrush et al., 2022; Yuksekgonul et al., 2022; Tong et al., 2024) (Fig. 1(b)). This limitation matters increasingly: as modern generators (Esser et al., 2024; Labs, 2024; Chen et al., 2024b) suppress low-level artifacts, detection relies increasingly on high-level cues, such as semantic and structural anomalies (Sarkar et al., 2024; Zhou et al., 2025; Ye et al., 2025), e.g., the implausible reflection in Fig. 1(a), which CLIP features alone may fail to capture.

In contrast, large language models (LLMs) (Tom et al., 2020; Grattafiori et al., 2024) are pretrained by next-token prediction, an objective that, unlike the global alignment of CLIP, requires every token to be interpreted in relation to its context. LLMs thereby acquire strong relational modeling among tokens, precisely the ability that CLIP lacks. Moreover, recent studies show that frozen LLM layers transfer to visual recognition tasks (Pang et al., 2024; Lai et al., 2024; Tang et al., 2025). This raises a natural question: can the relational modeling learned from language be exploited to re-encode CLIP visual tokens and expose the high-level cues of generated images?

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.

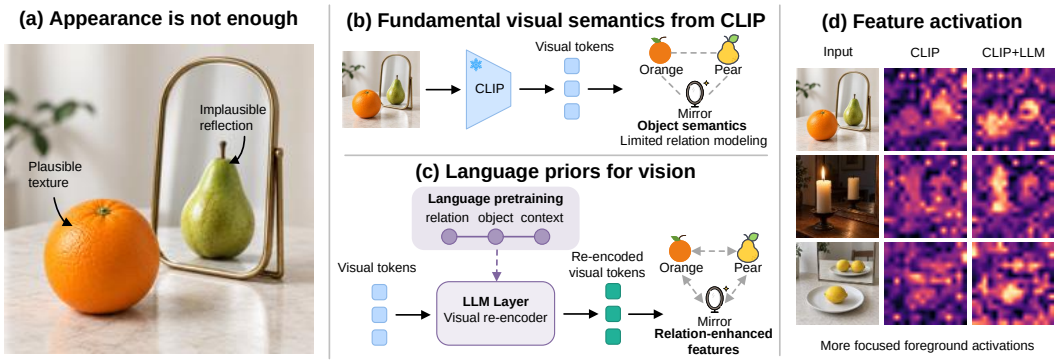


Figure 1: **Motivation.** CLIP captures what is in an image but models relations among regions poorly (a, b). Re-encoding its tokens with a frozen LLM layer, without any text, yields object-focused activations (c, d; see App. B.2).

To answer this question, we insert a single decoder layer of a pretrained LLM (hereafter, *LLM layer*) between a frozen CLIP encoder and a linear head as a semantic relation enhancement module (Fig. 1(c)). CLIP visual tokens are projected into the LLM hidden space, re-encoded by the LLM layer under bidirectional attention, and projected back. CLIP and the LLM layer stay frozen; only the two projections, a low-rank adapter (LoRA) (Hu et al., 2022), and the head are trained, about 4.3M parameters in total. Unlike MLLM-based detectors (Xu et al., 2025; Huang et al., 2025; Wen et al., 2026; Zhou et al., 2025), which use the LLM as a prompted text decoder, our detector takes no language input and predicts in one forward pass, and a single layer of the compact TinyLlama-1.1B (Zhang et al., 2024) already suffices.

Why should a layer pretrained only on text help a visual forensic task? Our controlled ablations show that the gain comes from the language-pretrained weights: neither a capacity-matched MLP nor a randomly initialized layer of identical architecture recovers it, even when the latter is fully trained. Moreover, a substantial part of the gain is lost when the LLM layer is restricted to the class token alone, which removes all token interaction, indicating the role of relational modeling among tokens. Visual analyses are consistent with this picture: after re-encoding, feature activations shift from a few outlier tokens to coherent object regions (Fig. 1(d)), and occlusion analysis shows that the decision relies on these regions rather than on scattered evidence. The benefit holds across four LLMs from 1.1B to 8B parameters. In summary, our contributions are as follows:

- We revisit generated image detection from the perspective of relational modeling. As low-level artifacts fade, detection increasingly hinges on high-level cues, which require modeling relations among visual tokens. This capability is left underdeveloped by the global alignment objective of CLIP, yet is exactly what LLMs acquire through next-token prediction.
- We propose to re-encode CLIP visual tokens with a single frozen LLM layer as a semantic relation enhancement module. It takes no language input and trains only about 4.3M parameters.
- Extensive experiments show that our method outperforms state-of-the-art detectors by 3.05, 3.02, and 4.33 accuracy points under three protocols. Controlled ablations and visual analyses further show that the gain stems from the language-pretrained weights and from relational modeling among tokens, rather than from added capacity or architecture.

2 Related Work

Generated image detection. Early detectors exploit low-level artifacts in the spatial, frequency, noise, gradient, and local-pixel domains (Wang et al., 2020; Frank et al., 2020; Liu et al., 2022; Tan et al., 2023; 2024) or diffusion reconstruction (Wang et al., 2023; Luo et al., 2024; Chen et al., 2024a), but such cues transfer poorly to unseen generators. Recent detectors therefore build on CLIP. UnivFD (Ojha et al., 2023) classifies frozen CLIP features, and Cozzolino et al. (2024) show that such detectors generalize because they rely on higher-level evidence. Later works adapt CLIP

with prompts or text features (Khan & Dang-Nguyen, 2024; Lin et al., 2024; Tan et al., 2025) or regularize its feature space against shortcuts (Zhang et al., 2025; Yan et al., 2024). All of them operate within the CLIP representation and inherit its limited relational modeling.

MLLMs for image forensics. MLLMs have been prompted zero-shot to judge authenticity (Jia et al., 2024), benchmarked on forgery-related tasks (Ye et al., 2025; Li et al., 2025), and instruction-tuned into explainable detectors that output textual rationales alongside predictions (Xu et al., 2025; Huang et al., 2025; Wen et al., 2026; Zhou et al., 2025). In all these works the LLM acts as a prompted text decoder: perception is still delegated to a CLIP-like encoder, and explanations come at a high training and inference cost. The relational modeling that LLMs acquire in pretraining has not been used to enhance the visual representation itself.

LLM layers as visual encoders. Standard multimodal architectures project visual tokens into an LLM, which then decodes text (Li et al., 2023; Liu et al., 2023; Alayrac et al., 2022). Departing from this design, Pang et al. (2024) show that inserting a frozen Transformer block of a pretrained LLM into a visual encoder improves downstream performance. Follow-up works apply the idea to biomedical image classification (Lai et al., 2024) and segmentation (Tang et al., 2025). These works use the LLM layer as a general-purpose booster for recognition, where the answer lies in *what* an image contains. Generated image detection poses a different demand: as low-level artifacts fade, the evidence lies increasingly in how the contents of an image relate to each other, and exposing such high-level cues, e.g., semantic and structural anomalies, calls for relational modeling among visual tokens. This is precisely what CLIP-based detectors lack and what a language-pretrained layer offers. Generated image detection is therefore a natural fit for this idea, yet the combination has remained unexplored. Our work fills this gap. An extended discussion is given in App. A.

3 METHOD

Fig. 2 gives an overview of our detector. A frozen CLIP encoder extracts visual tokens (Sec. 3.1), a single frozen LLM layer re-encodes them as a semantic relation enhancement module, and the re-encoded class token is classified by a linear head (Sec. 3.2, Fig. 2(a)). Sec. 3.2 also describes how the layer, originally designed for text, is applied to visual tokens (Fig. 2(c)), and Sec. 3.3 how it is adapted with LoRA and trained (Fig. 2(b)).

3.1 PRELIMINARIES

Generated image detection is a binary classification problem: given an image $\boldsymbol{x}$, predict $y \in \{0, 1\}$, where 1 denotes a generated image. Generalizable detectors (Ojha et al., 2023; Cozzolino et al., 2024) commonly use a frozen CLIP visual encoder $\mathcal{E}_v$ as the feature extractor and take the output tokens of its last layer,

$$\boldsymbol{T} = \mathcal{E}_v(\boldsymbol{x}) = [\boldsymbol{t}_{\text{cls}}; \boldsymbol{t}_1; \cdots; \boldsymbol{t}_N] \in \mathbb{R}^{(N+1)\times d_v}, \tag{1}$$

where $\boldsymbol{t}_{\text{cls}}$ is the class token, $\{\boldsymbol{t}_i\}_{i=1}^{N}$ are the patch tokens, and d_v is the token dimension (N=256 and d_v=1024 for ViT-L/14). These detectors typically train a linear classifier on $\boldsymbol{t}_{\text{cls}}$ and discard the patch tokens. A linear probe, however, only reweights the features that CLIP already provides and cannot add the relational modeling among tokens that high-level cues require (Sec. 1). We therefore keep all N+1 tokens and re-encode them with a pretrained LLM layer.

3.2 LLM-BASED SEMANTIC RELATION ENHANCEMENT

Projection and re-encoding. The CLIP token space and the LLM hidden space differ in dimension and statistics. We normalize the tokens with a trainable LayerNorm, map them into the LLM hidden space with a trainable linear layer $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{d_v \times d_l}$, re-encode them with a single decoder layer $\mathcal{F}_\ell$ taken from a pretrained LLM (the last layer by default), and map them back with $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{d_l \times d_v}$ (Fig. 2(a)):

$$\boldsymbol{H} = \text{LN}(\boldsymbol{T})\,\boldsymbol{W}_{\text{in}}, \qquad \hat{\boldsymbol{H}} = \mathcal{F}_\ell(\boldsymbol{H}), \qquad \boldsymbol{Z} = \hat{\boldsymbol{H}}\,\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{(N+1)\times d_v}, \tag{2}$$

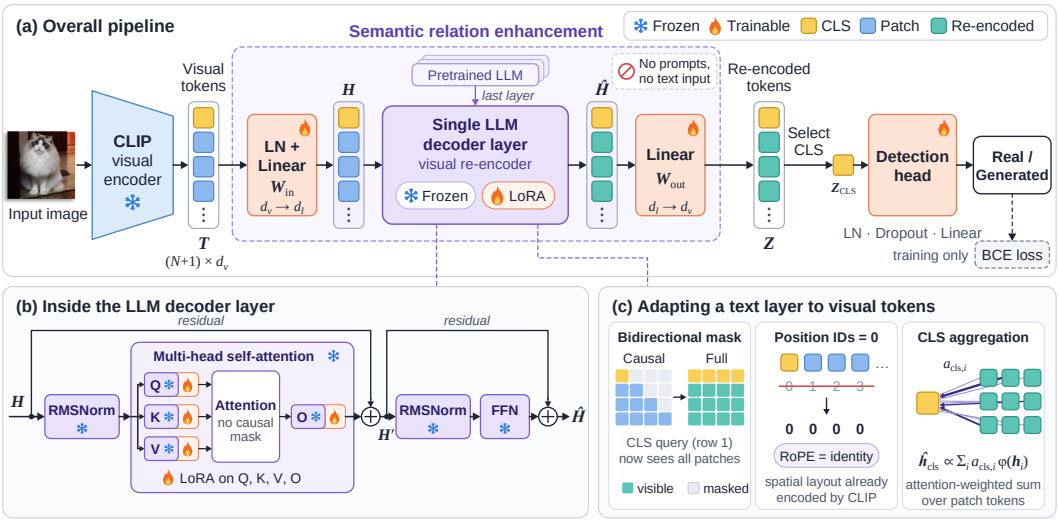


Figure 2: **Overview.** (a) Pipeline: CLIP visual tokens are projected into a single frozen LLM layer, re-encoded, and projected back; the re-encoded class token is classified. (b) Inside the layer: LoRA is applied to the attention projections only. (c) Applying a text layer to visual tokens: bidirectional attention, zero position indices, and class-token aggregation.

where d_l is the hidden dimension of the LLM. $\mathcal{F}_\ell$ is a standard pre-norm Transformer block (Fig. 2(b)),

$$\boldsymbol{H}' = \boldsymbol{H} + \mathrm{MHSA}\big(\mathrm{RMSNorm}(\boldsymbol{H})\big), \qquad \hat{\boldsymbol{H}} = \boldsymbol{H}' + \mathrm{FFN}\big(\mathrm{RMSNorm}(\boldsymbol{H}')\big), \tag{3}$$

whose weights carry the relational modeling acquired in language pretraining. Unlike the connectors of MLLMs (Liu et al., 2023; Li et al., 2023), $\boldsymbol{W}_{\text{in}}$ is not trained to align visual tokens with the word-embedding space. No text enters the layer, so $\boldsymbol{W}_{\text{in}}$ is free to learn the input representation that lets the frozen layer serve detection best.

Applying a text layer to visual tokens. A decoder layer is built for autoregressive text, so two of its components do not suit visual tokens. Following Pang et al. (2024), we change both (Fig. 2(c), left and middle). First, we remove the causal mask, since visual tokens arrive all at once and have no generation order. This is essential for our readout: CLIP places the class token first, so under a causal mask it could attend only to itself, and the model would reduce to the class-token-only variant of Tab. 6. Second, we set all position indices to zero, which turns the rotary positional embedding (RoPE) into the identity. RoPE encodes one-dimensional text order, whereas the two-dimensional layout of the patches is already contained in the CLIP tokens through the positional embeddings of CLIP.

Class-token readout. We use the re-encoded class token $\boldsymbol{z}_{\text{cls}} = \boldsymbol{Z}[0]$ as the image-level representation. With the causal mask removed, the attention sub-layer updates the class token as

$$\boldsymbol{h}'_{\text{cls}} = \boldsymbol{h}_{\text{cls}} + \sum_{i=0}^{N} a_{\text{cls},i}\, \boldsymbol{v}_i, \qquad a_{\text{cls},i} = \mathrm{softmax}_i\big(\boldsymbol{q}_{\text{cls}}^\top \boldsymbol{k}_i / \sqrt{d}\big), \tag{4}$$

where $\boldsymbol{q}$, $\boldsymbol{k}$, and $\boldsymbol{v}$ denote the query, key, and output-projected value of a token, and heads are omitted for clarity (Fig. 2(c), right). The class token, which carries the global semantics of the image, thus queries every patch token through the pretrained attention, and its update is dominated by the regions it relates to most strongly. This is the token interaction that the class-token-only variant in Sec. 4.3 removes. It is in line with the information filtering hypothesis of Pang et al. (2024), under which a frozen LLM layer amplifies informative tokens, and it suits a task in which a few anomalous regions can decide the authenticity of the whole image. The logit is $\hat{y} = \boldsymbol{w}_c^\top \mathrm{Dropout}\big(\mathrm{LN}(\boldsymbol{z}_{\text{cls}})\big) + b$.

Table 1: **ACC (%) under Protocol I.** Detectors are trained on SDv1.4 (†) and tested on the diffusion generators of GenImage and the GAN generators of ForenSynths, whose family is unseen in training. Baselines are grouped, from top to bottom, into artifact-based, reconstruction-based, and CLIP-based detectors. Best and second-best results are marked in red and blue.

Method	Venue	Diffusion generators (seen family)								GAN generators (unseen family)							Overall *ACC*
		SDv1.4†	SDv1.5	Wukong	GLIDE	ADM	VQDM	Midjourney	*Avg.*	ProGAN	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	*Avg.*	
Fusing	ICIP'22	99.90	99.91	99.90	57.50	51.30	64.20	52.30	75.00	51.20	50.20	53.50	52.40	58.20	49.32	52.47	64.60
LGrad	CVPR'23	99.12	99.05	97.53	64.24	53.00	50.93	76.34	77.17	61.61	61.43	48.82	60.74	50.17	49.70	55.41	67.13
NPR	CVPR'24	100.00	99.90	100.00	89.70	73.00	68.30	82.30	87.60	60.30	58.00	59.20	67.20	73.20	52.00	61.65	75.62
DIRE	ICCV'23	98.08	98.26	98.42	94.85	96.60	98.75	95.87	97.26	52.93	52.49	52.75	52.12	54.90	50.73	52.65	76.67
DRCT	ICML'24	95.01	94.41	94.68	89.18	79.42	90.03	88.50	90.18	76.83	71.58	82.68	94.51	63.13	78.39	77.85	84.49
UnivFD	CVPR'23	86.60	87.80	78.20	78.20	58.85	60.70	75.90	75.18	61.75	66.50	65.65	65.55	67.20	60.20	64.48	70.24
CLIPping	ICMR'24	96.07	95.48	88.86	85.00	70.14	79.35	77.66	84.65	88.78	80.69	89.57	88.48	90.82	85.94	87.38	85.91
CO-SPY	CVPR'25	88.88	94.25	88.93	80.78	68.67	78.53	78.59	82.66	77.84	68.88	76.75	69.65	67.48	77.57	73.03	78.21
VIB-Net	CVPR'25	99.55	99.20	98.25	74.25	73.85	89.35	88.05	88.93	89.70	74.10	91.20	88.60	80.70	87.15	85.24	87.23
Effort	ICML'25	100.00	100.00	99.50	85.00	72.00	89.50	78.00	89.14	94.00	75.00	93.50	88.50	95.50	80.00	87.75	88.50
Ours	–	100.00	99.95	99.80	86.05	74.35	96.05	82.00	91.17	94.50	76.85	96.00	98.10	99.05	87.45	91.99	91.55

Table 2: **AP (%) under Protocol I.** The setting, grouping, and marking follow Tab. 1. AP is threshold-free and complements ACC, which is measured at a fixed threshold of 0.5.

Method	Venue	Diffusion generators (seen family)								GAN generators (unseen family)							Overall *AP*
		SDv1.4†	SDv1.5	Wukong	GLIDE	ADM	VQDM	Midjourney	*Avg.*	ProGAN	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	*Avg.*	
Fusing	ICIP'22	99.80	97.98	99.90	94.20	69.30	84.60	81.20	89.57	67.63	67.90	69.37	87.79	91.20	43.08	71.16	81.07
LGrad	CVPR'23	99.94	99.92	99.72	58.52	58.52	56.34	91.06	80.57	83.59	82.74	47.51	90.24	99.19	49.25	75.42	78.20
NPR	CVPR'24	100.00	99.97	100.00	95.80	94.70	86.30	95.50	96.04	83.30	82.70	72.00	94.90	97.30	66.00	82.70	89.88
DIRE	ICCV'23	99.83	99.85	99.88	99.13	99.39	99.99	99.25	99.62	49.68	50.40	51.52	49.89	48.46	50.55	50.08	76.75
DRCT	ICML'24	99.65	99.49	99.37	94.64	88.96	96.54	97.03	96.53	91.03	79.44	93.51	98.68	96.29	86.62	90.93	93.94
UnivFD	CVPR'23	96.20	96.26	90.98	93.73	72.25	78.40	92.08	88.56	58.37	59.20	80.21	67.22	59.93	70.90	65.97	78.13
CLIPping	ICMR'24	93.97	93.10	86.52	87.44	68.00	77.17	77.34	83.36	88.54	77.23	85.33	88.44	89.82	81.56	85.15	84.19
CO-SPY	CVPR'25	95.34	98.63	99.93	91.63	82.42	88.65	97.81	93.49	82.26	69.95	78.41	68.28	71.40	78.74	74.84	84.88
VIB-Net	CVPR'25	100.00	99.97	99.93	97.13	95.49	97.00	97.81	98.19	96.59	84.31	97.17	98.44	97.60	96.94	95.18	96.80
Effort	ICML'25	100.00	100.00	100.00	97.38	91.00	99.71	98.68	98.11	98.58	95.61	97.56	92.60	98.43	89.93	95.45	96.88
Ours	–	100.00	100.00	99.99	99.37	98.03	99.84	98.99	99.46	99.83	98.13	99.29	99.95	99.97	98.52	99.28	99.38

3.3 Task Adaptation and Training

A fully frozen LLM layer keeps its pretrained priors intact but is never adapted to detection, whereas fine-tuning all of its weights on the comparatively small detection data generalizes worse (Tab. 7). We take a middle path and apply LoRA (Hu et al., 2022) to the four attention projections $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, $\boldsymbol{W}_v$, and $\boldsymbol{W}_o$ (Fig. 2(b)), leaving the feed-forward network untouched. Each projection becomes $\boldsymbol{W}_0 + \frac{\alpha}{r}\boldsymbol{B}\boldsymbol{A}$, where $\boldsymbol{W}_0$ is frozen and $\boldsymbol{B}$ is initialized to zero, so that training starts exactly from the pretrained layer and only adjusts how attention relates the tokens. With r=4, this adds 51.2K parameters, about 0.1% of the layer.

The trainable parameters are thus the input LayerNorm, $\boldsymbol{W}_{\text{in}}$, $\boldsymbol{W}_{\text{out}}$, the LoRA matrices, and the classification head, about 4.3M in total. They are optimized end-to-end with the binary cross-entropy loss on $\sigma(\hat{y})$. Overall, the detector adds 48M parameters (16%) to the 303M CLIP encoder and predicts in a single forward pass.

4 Experiments

4.1 Experimental Setup

Datasets and protocols. We evaluate generalization under three protocols. **Protocol I** (diffusion source): train on the SDv1.4 subset of GenImage (Zhu et al., 2024) and test on all GenImage subsets and on the GAN generators of ForenSynths (Wang et al., 2020), 13 generators in total. **Protocol II** (GAN source): following common practice (Tan et al., 2024; Jeong et al., 2022), train on four ProGAN categories of ForenSynths (car, cat, chair, horse) and test on the same 13 generators. **Protocol III** (emerging generators): train on the Holmes training split (Zhou et al., 2025) and test on three recent diffusion models, FLUX, PixArt-XL, and SD3.5 (Labs, 2024; Chen et al., 2024b; Esser et al., 2024), and six autoregressive generators, LlamaGen, Infinity, Janus, Janus-Pro, VAR, and Show-o (Sun et al., 2024; Han et al., 2025; Wu et al., 2025; Chen et al., 2025; Tian et al., 2024; Xie et al., 2024). No autoregressive generator appears in training, which makes this the most demanding protocol.

Table 3: **ACC (%) under Protocol II.** Detectors are trained on four ProGAN categories of ForenSynths (†) and tested on the same 13 generators as in Tab. 1, so that the GAN columns measure transfer within the training family and the diffusion columns transfer to an unseen family. Grouping and marking follow Tab. 1.

Method	Venue	GAN generators (seen family)							Diffusion generators (unseen family)								Overall *ACC*
		ProGAN†	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	*Avg.*	SDv1.4	SDv1.5	Wukong	GLIDE	ADM	VQDM	Midjourney	*Avg.*	
NPR	CVPR'24	99.60	90.80	73.00	81.20	99.00	62.00	84.27	79.00	86.00	81.00	82.00	71.00	61.00	82.00	77.43	80.58
UnivFD	CVPR'23	100.00	79.00	97.50	93.50	87.00	98.50	92.58	55.50	54.00	56.50	66.50	74.00	91.50	48.00	63.71	77.04
CLIPping	ICMR'24	99.77	89.50	94.27	87.26	98.09	95.63	94.09	71.29	72.48	70.01	84.29	75.45	83.26	57.27	73.44	82.96
VIB-Net	CVPR'25	99.50	83.00	92.50	93.50	93.00	100.00	93.58	70.00	70.00	74.00	62.00	69.00	86.00	60.50	70.21	81.00
Effort	ICML'25	100.00	87.50	98.50	92.50	96.50	100.00	95.83	72.00	74.00	76.00	83.50	65.50	86.50	54.00	73.07	83.58
Ours	–	99.80	91.95	94.85	95.60	96.35	95.15	95.62	79.55	79.25	79.90	88.30	78.50	88.40	58.25	78.88	86.60

Table 4: **AP (%) under Protocol II.** The setting follows Tab. 3, and grouping and marking follow Tab. 1.

Method	Venue	GAN generators (seen family)							Diffusion generators (unseen family)								Overall *AP*
		ProGAN†	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	*Avg.*	SDv1.4	SDv1.5	Wukong	GLIDE	ADM	VQDM	Midjourney	*Avg.*	
NPR	CVPR'24	100.00	99.30	76.10	98.20	100.00	70.10	90.62	97.50	85.20	90.50	62.70	77.10	66.10	86.30	80.77	85.32
UnivFD	CVPR'23	100.00	96.20	99.53	99.99	99.87	100.00	99.27	64.32	57.17	69.82	81.97	84.97	98.10	42.48	71.26	84.19
CLIPping	ICMR'24	99.76	89.27	90.48	96.68	97.16	91.96	94.22	77.01	76.47	81.02	82.30	78.73	87.44	55.86	76.98	84.93
VIB-Net	CVPR'25	99.81	89.59	98.06	99.32	96.45	100.00	97.21	76.98	77.92	83.34	72.98	78.25	91.87	66.91	78.32	87.04
Effort	ICML'25	100.00	98.22	99.99	95.46	99.96	100.00	98.94	91.01	92.25	90.44	94.69	75.42	94.94	65.48	86.32	92.14
Ours	–	100.00	98.24	99.29	99.85	99.80	99.98	99.53	93.21	93.02	93.44	96.33	92.52	96.76	72.68	91.14	95.01

Implementation details. We use the frozen CLIP ViT-L/14 encoder with 224×224 inputs and, unless otherwise specified, the last decoder layer of TinyLlama-1.1B (Zhang et al., 2024). LoRA uses rank r=4, α=16, and dropout 0.05. We train for 50 epochs with AdamW (learning rate 10^{-4}, weight decay 10^{-2}), batch size 256, fp16 mixed precision, and gradient clipping at norm 1.0, on a single NVIDIA RTX A6000 GPU.

Metrics and baselines. Following common practice (Ojha et al., 2023; Tan et al., 2024), we report accuracy (ACC) at a fixed threshold of 0.5 and the threshold-free average precision (AP). We compare with detectors based on low-level artifacts (Ju et al., 2022; Tan et al., 2023; 2024), diffusion reconstruction (Wang et al., 2023; Chen et al., 2024a), and CLIP features (Ojha et al., 2023; Khan & Dang-Nguyen, 2024; Cheng et al., 2025; Zhang et al., 2025; Yan et al., 2024).

4.2 Comparison with State-of-the-Art Detectors

Protocol I: diffusion source. Protocol I investigates whether knowledge acquired from a single diffusion generator can transfer to other generative architectures. Trained on SDv1.4 alone, our detector reaches 91.55% average ACC and 99.38% average AP over the 13 generators (Tabs. 1 and 2), exceeding the two strongest CLIP-based detectors, Effort and VIB-Net, by 3.05 and 4.32 ACC points. The margin over both is larger on the GAN generators, whose family is absent from training. Reconstruction-based DIRE is the most accurate on several diffusion subsets but falls to chance level on every GAN, whereas our detector stays above 87% on five of the six. Our weakest subsets are ADM, StyleGAN, and Midjourney, on which some baselines are more accurate.

Protocol II: GAN source. Protocol II examines whether a detector learned from a single GAN source remains effective when the underlying synthesis mechanism changes. Trained on four ProGAN categories, our detector again obtains the best averages, 86.60% ACC and 95.01% AP, ahead of Effort by 3.02 and 2.87 points (Tabs. 3 and 4). On the GAN targets it is on par with Effort (95.6% vs. 95.8% on average), and the margin comes from the diffusion targets (78.9% vs. 73.1%), where it has the highest ACC on SDv1.4, GLIDE, and ADM and the highest AP on SDv1.5, GLIDE, ADM, and Wukong. Midjourney remains difficult for every detector except NPR.

Protocol III: emerging generators. On recent diffusion and autoregressive generators excluded from training, our detector reaches 98.43% mean ACC and 99.82% mean AP (Tab. 5), 4.33 ACC points above Effort and 0.52 AP points above LaRE, the strongest baselines on the respective metrics. It ranks first in ACC on eight of the ten generator configurations and in AP on seven. The main exception is VAR, where LaRE is more accurate. This broad advantage indicates that the semantic

Table 5: Generalization to emerging generative architectures under Protocol III. The evaluation covers autoregressive and diffusion generators excluded from training.

Method	Autoregressive Generators														Diffusion Generators						Aggregate	
	Janus		Janus-Pro-1B		Janus-Pro-7B		Show-o		LlamaGen		Infinity		VAR		PixArt-XL		SD3.5-L		FLUX		Mean	
	ACC	*AP*	*ACC*	*AP*	*ACC*	*AP*	*ACC*	*AP*	*ACC*	*AP*	*ACC*	*AP*	*ACC*	*AP*	*ACC*	*AP*	*ACC*	*AP*	*ACC*	*AP*	*ACC*	*AP*
CNNSpot	70.00	86.00	70.90	85.80	85.00	93.60	72.20	86.00	61.90	71.40	86.80	94.60	59.90	75.00	78.20	90.10	63.80	81.10	81.10	92.00	72.90	85.60
FreDect	66.68	63.90	78.37	75.57	78.53	76.15	71.50	72.32	76.69	84.25	79.83	84.65	63.79	65.62	72.49	74.60	68.07	69.25	75.76	90.46	73.17	75.68
Fusing	92.58	96.22	95.56	97.52	95.95	97.77	96.11	97.97	96.08	97.97	86.33	87.99	86.13	87.97	95.86	97.85	90.80	92.80	95.72	97.84	93.11	95.19
UnivFD	87.60	97.80	96.90	99.50	96.40	99.50	85.90	97.40	93.10	98.60	79.20	96.20	64.30	85.90	75.70	94.40	87.80	97.80	69.60	91.40	83.60	95.90
CLIPping	84.62	84.29	97.03	96.66	95.94	95.68	92.21	91.89	96.30	96.02	90.47	90.21	67.59	67.31	88.85	88.53	96.57	96.36	84.22	84.06	89.38	89.10
NPR	51.20	55.90	69.50	75.10	73.90	77.90	93.70	99.60	93.50	99.40	93.80	99.90	85.90	91.20	93.40	99.10	91.60	97.70	93.60	99.50	84.00	89.50
LaRE	70.80	99.30	74.70	97.50	95.60	99.70	80.00	99.00	91.60	99.60	77.90	99.60	98.80	100.00	82.20	99.70	94.10	99.50	84.30	99.00	85.00	99.30
VIB-Net	92.50	99.61	97.00	99.80	97.50	99.92	94.50	99.73	99.50	99.99	97.50	99.90	71.50	93.21	91.00	98.75	98.00	99.87	88.50	98.79	92.75	98.96
Effort	96.00	99.50	97.50	99.76	96.50	99.73	93.50	99.42	98.50	99.95	97.50	99.99	80.50	96.33	95.00	99.74	98.00	99.97	88.00	98.42	94.10	99.28
Ours	99.80	99.94	99.30	99.99	99.35	99.96	99.40	99.99	99.40	99.97	98.60	99.92	94.30	98.94	98.15	99.92	99.50	99.98	96.50	99.59	98.43	99.82

Table 6: **Source of the improvement** (Protocol I, mean over 13 generators). Rows 2–4 have identical trainable parameter counts and each changes one factor relative to the row above, and Δ is the change from the previous row. Row 5 re-initializes the LLM layer of row 4 and trains all of its weights. Params: total / trainable (M).

#	Variant	Re-encoder	Adapt.	Tokens	Params (M)	*ACC*	Δ*ACC*	*AP*	Δ*AP*
1	CLIP + linear probe	–	–	$\boldsymbol{t}_{\text{cls}}$	303.18 / 0.00	84.75	–	95.43	–
2	+ capacity-matched MLP	MLP (random)	full	$\boldsymbol{t}_{\text{cls}}$	307.43 / 4.25	85.92	+1.17	96.10	+0.67
3	+ pretrained layer, class token only	$\mathcal{F}_{21}$ (pretrained)	LoRA	$\boldsymbol{t}_{\text{cls}}$	351.47 / 4.25	88.95	+3.03	98.54	+2.44
4	+ all patch tokens (**Ours**)	$\mathcal{F}_{21}$ (pretrained)	LoRA	all $N{+}1$	351.47 / 4.25	**91.55**	+2.60	**99.38**	+0.84
5	Random layer, fully trained	$\mathcal{F}$ (random)	full	all $N{+}1$	351.42 / 48.24	86.72	–	96.93	–

and structural relations captured through LLM-based visual re-encoding remain discriminative even when the target generation paradigm is absent from the training data.

4.3 Ablation Study

All ablations follow Protocol I and report *ACC* and *AP* averaged over the 13 test generators. Each variant changes a single factor of the default configuration.

Where does the improvement come from? The gain over CLIP could stem from the added trainable parameters, from inserting a Transformer block of any kind, or from the pretrained LLM weights. Tab. 6 separates these factors one at a time. Row 3 feeds only $\boldsymbol{t}_{\text{cls}}$ into $\mathcal{F}_\ell$. With a sequence of length one, the attention weight in Eq. 4 is identically one and no token interaction can occur, so its only difference from the full model (row 4) is the number of tokens entering the layer. Three observations follow. (i) Capacity is a minor factor: a capacity-matched MLP adds only $+1.17$ *ACC*. (ii) The pretrained weights help even without token interaction: at the same number of trainable parameters, replacing the random MLP by the pretrained layer adds $+3.03$ *ACC* (row 2$\rightarrow$3). (iii) Admitting the patch tokens adds a further $+2.60$ *ACC* at no parameter cost (row 3$\rightarrow$4). In relative error reduction, this step (23.5%) is as large as the previous one (21.5%). It cannot be explained by access to patch information alone, since a randomly initialized layer receives the same tokens and trains $11\times$ more parameters, yet reaches only 86.72 (row 5). The gain thus comes from the pretrained weights, and a substantial part of it from the relational modeling among tokens that they perform.

How strongly should the LLM layer be adapted? Tab. 7 varies the trainable parameters inside $\mathcal{F}_\ell$ from none to all 44.0M. With the layer fully frozen and only the projections and the head trained, the model already reaches 90.87 *ACC*, $+6.12$ over linear probing, so the pretrained weights are effective without any adaptation. LoRA with $r{=}4$ adds only 51.2K parameters and performs best, while larger ranks bring no further gain, which suggests that the adaptation needed for detection is low-dimensional. Full fine-tuning is harmful: with $11\times$ more trainable parameters, *ACC* drops to 88.99, the lowest among all settings, consistent with overfitting on the limited detection data.

Does the benefit depend on which layer is used? Tab. 8 replaces the last decoder layer of TinyLlama with layers 0, 5, 10, and 15. All layers have identical shapes, so the five models share architec-

Table 7: **Adaptation strength.** Only the number of trainable parameters inside $\mathcal{F}_\ell$ varies, from fully frozen to fully fine-tuned.

Adaptation of $\mathcal{F}_\ell$	In $\mathcal{F}_\ell$	Train. (M)	*ACC*	*AP*
Fully frozen	0	4.20	90.87	98.88
LoRA, r=4 (**Ours**)	51.2 K	4.25	**91.55**	**99.38**
LoRA, r=8	102.4 K	4.30	90.99	98.65
LoRA, r=16	204.8 K	4.40	91.19	99.31
Full fine-tuning	44.0 M	48.24	88.99	99.00

Table 8: **Depth of the LLM layer.** Only the layer index ℓ of TinyLlama changes.

Layer ℓ	*ACC*	*AP*	Params (M)
CLIP + linear (ref.)	84.75	95.43	303.18 / 0.00
0 (shallowest)	85.10	96.12	351.47 / 4.25
5	88.33	97.11	351.47 / 4.25
10	86.38	96.89	351.47 / 4.25
15	89.25	98.44	351.47 / 4.25
21 (last, **Ours**)	**91.55**	**99.38**	351.47 / 4.25

Table 9: **Number of stacked LLM layers.** The last k decoder layers of TinyLlama are stacked, each with its own LoRA (r=4). Params: total / trainable (M). The default configuration is shaded.

k	Layers	*ACC*	*AP*	Params (M)
1	21	91.55	**99.38**	351.47 / 4.25
2	20–21	90.12	99.16	395.57 / 4.30
3	19–21	**91.61**	99.16	439.66 / 4.35
4	18–21	90.92	99.05	483.76 / 4.40

Table 10: **Different LLMs.** The last decoder layer of each LLM replaces that of TinyLlama, with LoRA (r=4) and all other settings unchanged. d_l is the hidden dimension of the LLM. The default configuration is shaded.

LLM	d_l	*ACC*	*AP*
TinyLlama-1.1B	2048	**91.55**	99.38
Qwen2.5-7B	3584	88.18	99.25
Mistral-7B-v0.3	4096	90.57	99.42
Llama-3.1-8B	4096	91.22	**99.47**

ture and parameter counts and differ only in what their pretrained weights encode. The shallowest layer is nearly ineffective ($+0.35$ *ACC* over linear probing, below the capacity-matched MLP). Accuracy then rises with depth, though not monotonically (ℓ=10 falls below ℓ=5), and peaks at the last layer. One plausible explanation is that the later layers of a language model capture more abstract and global relations among tokens, which suits detection cues that are themselves high-level.

Are multiple LLM layers necessary? Tab. 9 stacks the last k pretrained layers, each with its own LoRA. Stacking brings no consistent benefit: *ACC* varies between 90.12 and 91.61 without a trend, and *AP* decreases slightly with k. A single layer already lets the class token interact with every patch token. Deeper stacks additionally let the patch tokens interact with each other before the readout, which does not help further. We therefore use a single layer, which is also the cheapest configuration.

Does the benefit generalize across LLMs? Tab. 10 replaces TinyLlama-1.1B with the last decoder layer of Qwen2.5-7B (Hui et al., 2024), Mistral-7B-v0.3, and Llama-3.1-8B (Grattafiori et al., 2024). Every LLM improves markedly over linear probing ($+3.43$ to $+6.80$ *ACC*, $+3.82$ to $+4.04$ *AP*) and surpasses both the capacity-matched MLP and the randomly initialized layer of Tab. 6, so the benefit is not specific to one model. Scale is not decisive: the 1.1B TinyLlama attains the highest *ACC*, although a layer of the 7–8B models is about five times larger. *AP* varies by only 0.22 points across the four LLMs while *ACC* at the fixed threshold varies more, which suggests that they rank images almost equally well and differ mainly in score calibration. A single layer of a compact LLM is therefore sufficient.

4.4 Visual Analysis

What does the LLM layer change? We ask two separate questions. How the patch-token representation changes inside the layer is examined with the label-free feature activation, and what the decision depends on is examined with occlusion sensitivity (Zeiler & Fergus, 2014): each of the 256 patches is occluded in turn and the change of the correct-class logit is recorded. Fig. 3(a) shows that the CLIP activation is dominated by a few high-norm outlier tokens and has little spatial structure, whereas after the attention and FFN sub-layers it concentrates on coherent object regions, under both the magnitude and the frequency measure. Fig. 3(b) shows that the decision depends on similar regions. For our model, the patches that support the correct class form a contiguous region over the subject, and hardly any patch argues against it. For the capacity-matched MLP baseline (Tab. 6,

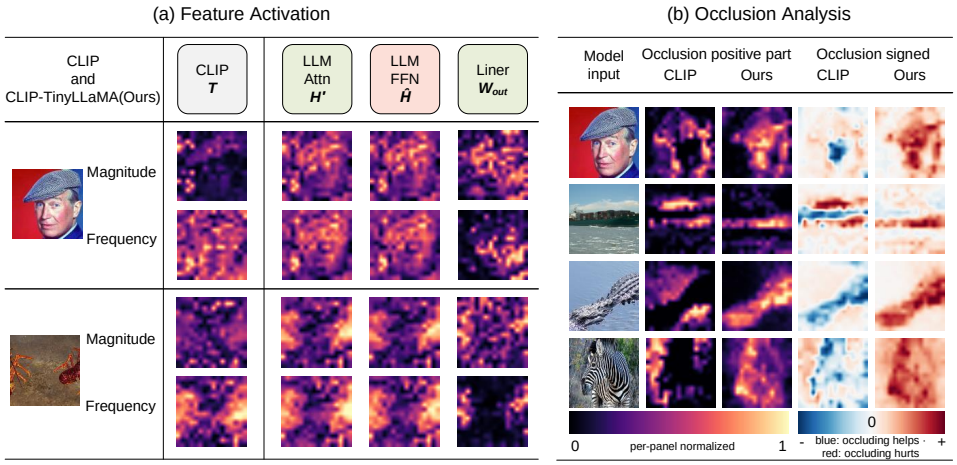


Figure 3: **Visual analysis.** (a) Feature activation of the patch tokens at the CLIP output $\boldsymbol{T}$, after the attention ($\boldsymbol{H}'$) and FFN ($\hat{\boldsymbol{H}}$) sub-layers, and after the output projection ($\boldsymbol{Z}$). (b) Occlusion sensitivity of the MLP baseline and of our model. Positive maps show the drop of the correct-class logit when a patch is occluded; signed maps also show patches whose occlusion raises it (blue).

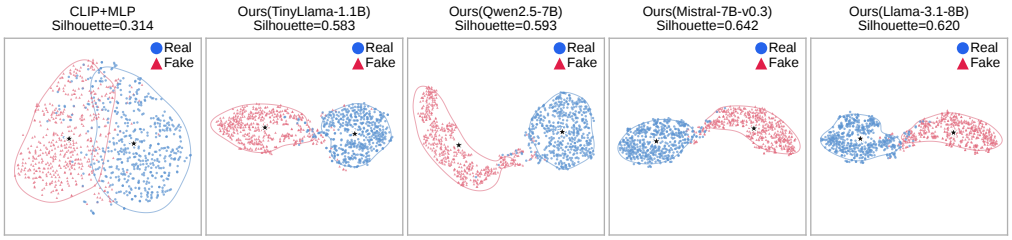


Figure 4: **t-SNE of the features fed to the final classifier** on VQDM, an unseen generator under Protocol I, with 500 real and 500 generated images per panel. Contours enclose about 90% of each class. Silhouette scores are computed on the embedding and measure geometric separation, not accuracy.

row 2), supporting evidence is scattered, and on images that it misclassifies, large parts of the subject push the prediction toward the wrong class. These observations are qualitative and based on selected samples; the selection protocol, statistics, and further examples are given in App. B.

Feature separability. Fig. 4 embeds the features fed to the final classifier for 500 real and 500 generated VQDM images, a generator unseen in training. The MLP baseline yields two overlapping clouds, whereas a single LLM layer yields two compact clusters for all four LLMs. The silhouette score in the original feature space rises from 0.14 to 0.53–0.64 (App. B.4).

5 Conclusion

As low-level artifacts fade, generalizable detection needs a representation that captures how the regions of an image relate, not only what they contain, so that high-level cues such as semantic and structural anomalies can surface. We showed that this capability can be borrowed from language pretraining rather than learned from forensic data: a single frozen LLM layer, given no text, adds relational modeling to CLIP tokens and improves detection by 3.0 to 4.3 accuracy points across three protocols. Controlled ablations trace the gain to the language-pretrained weights and the token interaction they perform, not to capacity or architecture. LLMs can thus serve image forensics through their weights alone, without prompts or text decoding. Limitations and future directions are discussed in App. C.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and language editing, including the preparation of the disclosure statements. We did not use these tools to develop the conceptual framework, formulate hypotheses or mathematical claims, design experiments, implement the proposed method, generate or process datasets. Theorem proving and proof writing are not applicable to this work. All AI-assisted text was reviewed and revised by the authors to ensure consistency with the methods, experimental results, and cited sources. The authors take full responsibility for the final manuscript, including all claims, results, references, and AI-assisted content.

ETHICS STATEMENT

This work aims to support image authenticity assessment through generated image detection. Our experiments use the existing benchmark datasets and pretrained models described in Sec. 4.1. We recognize that detecting whether an image is generated does not establish whether its content is truthful or whether its use is deceptive. False positives could unfairly discredit authentic images and their creators, while false negatives could allow misleading synthetic content to evade detection. Biases in the datasets and pretrained models may also lead to uneven performance across image domains and demographic groups, and detection methods could be misused to develop more effective evasion strategies. We therefore recommend treating predictions as supporting evidence rather than definitive judgments, and combining them with human review and provenance information. As discussed in App. C, robustness to post-processing has not been evaluated; deployment would require additional robustness and fairness assessments under the intended operating conditions. Any downstream use should also respect the privacy interests of depicted individuals and the usage conditions of the underlying datasets and pretrained models.

REPRODUCIBILITY STATEMENT

The model architecture and training objective are described in Sec. 3, including visual token extraction, the projection layers, modifications to attention and positional encoding, LoRA adaptation, and the classification head. Section 4.1 specifies the benchmark datasets, three training and evaluation protocols, default model configuration, optimization hyperparameters, input resolution, hardware, and evaluation metrics. The controlled variants used to examine the effects of pretrained weights, token interaction, and adaptation are described in Sec. 4.3. Appendix B documents the visual analysis procedures, including deterministic subsampling and sample identifiers, feature-activation measures, occlusion sensitivity, and rendering choices. Appendix B.4 further specifies the feature extraction, dimensionality reduction, visualization parameters, and random seed used for the feature-space analysis.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

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.

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

Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, pp. 74–91. Springer, 2024b.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

Siyuan Cheng, Lingjuan Lyu, Zhenting Wang, Xiangyu Zhang, and Vikash Sehwag. Co-spy: Combining semantic and pixel features to detect synthetic images by ai. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13455–13465, 2025.

Davide Cozzolino, Giovanni Poggi, Riccardo Corvi, Matthias Nießner, and Luisa Verdoliva. Raising the bar of ai-generated image detection with clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4356–4366, 2024.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International conference on learning representations*, volume 2024, pp. 2632–2652, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Fabrizio Guillaro, Giada Zingarini, Ben Usman, Avneesh Sud, Davide Cozzolino, and Luisa Verdoliva. A bias-free training paradigm for more general ai-generated image detection. *arXiv preprint arXiv:2412.17671*, 2024.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15733–15744, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

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

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Yonghyun Jeong, Doyeon Kim, Youngmin Ro, and Jongwon Choi. Frepgan: robust deepfake detection using frequency-level perturbations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 1060–1068, 2022.

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

Yan Ju, Shan Jia, Lipeng Ke, Hongfei Xue, Koki Nagano, and Siwei Lyu. Fusing global and local features for generalized ai-synthesized image detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 3465–3469. IEEE, 2022.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9161–9175, 2023.

Mamadou Keita, Wassim Hamidouche, Hessen Bougueffa Eutamene, Abdelmalik Taleb-Ahmed, and Abdenour Hadid. Fidavl: Fake image detection and attribution using vision-language model. In *International Conference on Pattern Recognition*, pp. 160–176. Springer, 2025.

Sohail Ahmed Khan and Duc-Tien Dang-Nguyen. Clipping the deception: Adapting vision-language models for universal deepfake detection. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*, pp. 1006–1015, 2024.

Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

Zhixin Lai, Jing Wu, Suiyao Chen, Yucheng Zhou, and Naira Hovakimyan. Residual-based language models are free boosters for biomedical imaging tasks, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

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

Li Lin, Irene Amerini, Xin Wang, Shu Hu, et al. Robust clip-based detector for exposing diffusion model-generated images. *arXiv preprint arXiv:2404.12908*, 2024.

Bo Liu, Fan Yang, Xiuli Bi, Bin Xiao, Weisheng Li, and Xinbo Gao. Detecting generated images by real images. In *European Conference on Computer Vision*, pp. 95–110. Springer, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Yunpeng Luo, Junlong Du, Ke Yan, and Shouhong Ding. Lareˆ 2: Latent reconstruction error based method for diffusion-generated image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17006–17015, 2024.

Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24480–24489, 2023.

Ziqi Pang, Ziyang Xie, Yunze Man, and Yu-Xiong Wang. Frozen transformers in language models are effective visual encoder layers. In *International Conference on Learning Representations*, 2024.

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

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

Ayush Sarkar, Hanlin Mai, Amitabh Mahapatra, Svetlana Lazebnik, David A Forsyth, and Anand Bhattad. Shadows don't lie and lines can't bend! generative models don't know projective geometry... for now. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28140–28149. IEEE, 2024.

Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image generation models. In *Proceedings of the 2023 ACM SIGSAC conference on computer and communications security*, pp. 3418–3432, 2023.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12105–12114, 2023.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28130–28139, 2024.

Chuangchuang Tan, Renshuai Tao, Huan Liu, Guanghua Gu, Baoyuan Wu, Yao Zhao, and Yunchao Wei. C2p-clip: Injecting category common prompt in clip to enhance generalization in deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7184–7192, 2025.

Fenghe Tang, Wenxin Ma, Zhiyang He, Xiaodong Tao, Zihang Jiang, and S. Kevin Zhou. Pre-trained llm is a semantic-aware and generalizable segmentation booster. In *Medical Image Computing and Computer Assisted Intervention*, 2025.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5228–5238. IEEE, 2022.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Brown Tom, Mann Benjamin, Ryder Nick, Subbiah Melanie, Dhariwal Prafulla, Neelakantan Arvind, Shyam Pranav, Sastry Girish, Askell Amanda, Agarwal Sandhini, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33(1):1877–1901, 2020.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9568–9578. IEEE, 2024.

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

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

Siwei Wen, Peilin Feng, Hengrui Kang, Zichen Wen, Yize Chen, Jiang Wu, Conghui He, Weijia Li, et al. Spot the fake: Large multimodal model-based synthetic image detection with artifact explanation. *Advances in Neural Information Processing Systems*, 38:58972–59005, 2026.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12966–12977, 2025.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

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. In *International Conference on Learning Representations*, volume 2025, pp. 31186–31216, 2025.

Zhiyuan Yan, Jiangming Wang, Peng Jin, Ke-Yue Zhang, Chengchun Liu, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Orthogonal subspace decomposition for generalizable ai-generated image detection. *arXiv preprint arXiv:2411.15633*, 2024.

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. In *International Conference on Learning Representations*, volume 2025, pp. 70440–70522, 2025.

Lin Yuan, Xiaowan Li, Yan Zhang, Jiawei Zhang, Hongbo Li, and Xinbo Gao. Mlep: Multi-granularity local entropy patterns for generalized ai-generated image detection. *Advances in Neural Information Processing Systems*, 38:68981–69000, 2026.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? *arXiv preprint arXiv:2210.01936*, 2022.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Haifeng Zhang, Qinghui He, Xiuli Bi, Weisheng Li, Bo Liu, and Bin Xiao. Towards universal ai-generated image detection by variational information bottleneck network. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23828–23837, 2025.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model, 2024.

Chende Zheng, Chenhao Lin, Zhengyu Zhao, Hang Wang, Xu Guo, Shuai Liu, and Chao Shen. Breaking semantic artifacts for generalized ai-generated image detection. *Advances in Neural Information Processing Systems*, 37:59570–59596, 2024.

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. *arXiv preprint arXiv:2507.02664*, 2025.

Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. Genimage: A million-scale benchmark for detecting ai-generated image. *Advances in Neural Information Processing Systems*, 36, 2024.

A Extended Related Work

Artifact-based detection. Early detectors learn low-level traces left by the generation process. CNNSpot (Wang et al., 2020) trains a CNN classifier on ProGAN images with blur and JPEG augmentation and transfers to other CNN-based generators. FreDect (Frank et al., 2020) exposes up-sampling artifacts in the frequency domain, Liu et al. (2022) characterize the noise patterns of real images, Fusing (Ju et al., 2022) combines global and local patch features, LGrad (Tan et al., 2023) uses the gradients of a pretrained CNN as a generalized artifact representation, and NPR (Tan et al., 2024) models the local pixel relationships induced by up-sampling. For diffusion models, DIRE (Wang et al., 2023) and LaRE^2 (Luo et al., 2024) measure reconstruction errors in the image and latent space, respectively, and DRCT (Chen et al., 2024a) uses diffusion reconstructions as hard samples for contrastive training. These detectors are accurate on the generators seen in training but degrade under distribution shift, because the traces they rely on are generator-specific and become fainter as generators improve.

Detection with pretrained vision-language features. UnivFD (Ojha et al., 2023) shows that nearest-neighbor or linear classification in the frozen CLIP feature space generalizes across generator families, and Cozzolino et al. (2024) show that a lightweight classifier on CLIP features, trained with few examples, is both general and robust, and appears to rely on cues other than the

low-level traces exploited by earlier detectors. Subsequent works adapt CLIP to the task. CLIPping (Khan & Dang-Nguyen, 2024) compares prompt tuning, adapters, and fine-tuning, Lin et al. (2024) combine CLIP image and text features with a lightweight MLP, DE-FAKE (Sha et al., 2023) additionally exploits the prompt associated with an image, and C2P-CLIP (Tan et al., 2025) injects category-common prompts into the image encoder. CO-SPY (Cheng et al., 2025) fuses CLIP semantic features with pixel-level artifact features. Another group regularizes the pretrained feature space: VIB-Net (Zhang et al., 2025) applies a variational information bottleneck to discard task-irrelevant information, and Effort (Yan et al., 2024) decomposes the weights into orthogonal subspaces to preserve pretrained knowledge while learning forgery cues. Orthogonally, some works improve the training data, by scaling the number of generators (Park & Owens, 2025) or by removing content bias between real and generated images (Guillaro et al., 2024), and Zheng et al. (2024) break image-level semantics by patch shuffling to avoid scene-specific shortcuts. The shortcut these works guard against is the semantic *content* of an image, such as its category or scene. Our concern is different: how the regions of an image relate to each other, which is independent of what the image depicts. All of the above methods operate within the CLIP representation or within features learned from pixels, and none introduces a module whose prior is relational modeling among tokens.

MLLMs for image forensics. Jia et al. (2024) prompt off-the-shelf MLLMs to judge image authenticity, and LOKI (Ye et al., 2025) and FakeBench (Li et al., 2025) benchmark MLLMs on synthetic-content detection and explanation, finding non-trivial zero-shot ability but limited reliability. AntifakePrompt (Chang et al., 2023) casts detection as visual question answering and tunes a soft prompt for an instruction-following VLM, and FIDAVL (Keita et al., 2025) addresses detection and attribution with a vision-language model. Instruction-tuned detectors pair the prediction with text: FakeShield (Xu et al., 2025) and SIDA (Huang et al., 2025) explain and localize manipulations, FakeVLM (Wen et al., 2026) describes artifacts in natural language, and AIGI-Holmes (Zhou et al., 2025) produces human-verifiable explanations. In all these systems the LLM is a text decoder driven by prompts and instruction data, visual perception is delegated to a CLIP-like encoder, and a model with billions of parameters is run autoregressively at inference. We use the LLM in a different role: a single layer, no text on either the input or the output side, and a single forward pass.

LLM layers as visual encoders. Multimodal architectures connect a visual encoder to an LLM through a Q-Former (Li et al., 2023), a linear projection (Liu et al., 2023), or gated cross-attention (Alayrac et al., 2022), and the LLM decodes text. Pang et al. (2024) depart from this design. They insert a single frozen Transformer block of an LLM between a visual encoder and its task head, with one linear layer on each side, remove the causal mask and the rotary positional embedding, and train the rest of the network from scratch. The block improves image, point-cloud, and video recognition, motion forecasting, and vision-language tasks, for several LLMs and block indices, and the authors explain the effect with the information filtering hypothesis: the pretrained block identifies informative tokens and amplifies their contribution. Lai et al. (2024) add a residual connection around the frozen block for biomedical image classification, and LLM4Seg (Tang et al., 2025) inserts frozen LLM layers into CNN encoder–decoder segmentation networks.

Our method differs from these works in the following four respects. (i) *Motivation.* Prior works use the LLM layer as a general-purpose module for improving recognition performance, whereas we introduce it to provide a specific capability required by the task but lacking in the CLIP backbone, namely relational modeling among visual tokens. (ii) *Setting.* Pang et al. (2024) always train the visual backbone jointly with the frozen LLM layer, either from scratch or by fine-tuning a pretrained backbone, while leaving open how the training process encourages the visual features to cooperate with the language-model layer. In contrast, we also keep the pretrained CLIP encoder frozen, so that only the projection layers, LoRA parameters, and classification head are trained (about 4.3M parameters). This rules out backbone co-adaptation as a possible source of the performance gain. (iii) *Adaptation.* We systematically study how strongly the layer should be adapted, ranging from fully frozen to fully fine-tuned, and find that a rank-4 LoRA applied only to the attention projections is sufficient (Tab. 7). (iv) *Analysis.* In addition to capacity-matched and randomly initialized controls, we isolate the contribution of token interaction by restricting the layer to process only the class token (Tab. 6). Moreover, beyond label-free feature activation analysis, we further employ occlusion sensitivity, which provides an interventional analysis (App. B).

Relational understanding in CLIP and high-level cues in generated images. Several studies document that contrastively trained vision-language models represent images much like bags of concepts. Winoground (Thrush et al., 2022) and ARO (Yuksekgonul et al., 2022) report near-chance performance when captions differ only in how the same objects and attributes are composed, Kamath et al. (2023) report failures on basic spatial relations, and Tong et al. (2024) trace systematic visual failures of MLLMs back to the CLIP encoder. On the generation side, Sarkar et al. (2024) show that images from modern generators, although photorealistic, violate projective geometry in their shadows and perspective, and that these violations alone suffice to distinguish them from real photographs. Together, these findings motivate our premise that high-level cues persist in generated images and that CLIP features alone are not well suited to capture them.

B Details of the Visual Analysis

This appendix documents the setup, the two measurements, and the rendering choices behind Fig. 3, and gives further examples (Figs. 5 and 6).

Why two measurements. The two analyses answer different questions. Feature activation is computed from intermediate features alone, without labels or gradients, so it shows which tokens stand out within an image but not which tokens support the decision. Occlusion sensitivity (Zeiler & Fergus, 2014) is interventional: it records how the output changes when a patch is removed. Keeping the two apart matters more for detection than for recognition. In classification or segmentation the semantic foreground is the answer, so salient tokens and useful tokens largely coincide, whereas the evidence that an image is generated may lie in the background, along object boundaries, or in texture, and a label-free map alone cannot show that the detector relies on the regions it highlights.

B.1 Setup and Sample Selection

Models. We compare two detectors that share the frozen CLIP ViT-L/14 encoder, 224×224 inputs with patch size 14, the SDv1.4 training split of GenImage, fp16 inference, and a decision threshold of 0.5. The *MLP baseline* is the capacity-matched MLP of Tab. 6 (row 2): it feeds the CLIP class token into a trainable MLP and contains no LLM layer. It appears as "CLIP" or "Baseline" in the figures. *Our model* is the default configuration of Sec. 3, with the last decoder layer of TinyLlama-1.1B and LoRA. CLIP outputs 257 tokens, namely one class token and 256 patch tokens that form a 16×16 grid. All heat maps are defined on this grid and can be compared position by position.

Sample selection. The candidate pool covers the 13 test generators of Protocol I (SDv1.4, SDv1.5, ADM, GLIDE, Midjourney, VQDM, Wukong, ProGAN, BigGAN, CycleGAN, GauGAN, StarGAN, and StyleGAN) with at most $1{,}000$ images per class and generator, obtained by seeded deterministic subsampling. Each sample carries an identifier derived from the SHA-1 hash of its generator, label, and file path; it is printed next to the generator name in Figs. 5 and 6, so the selection is reproducible.

B.2 Feature Activation

Captured stages. We record the token sequence of our model at the five positions listed in Tab. 11, each of which corresponds to one tensor of Sec. 3. Figs. 3(a) and 5 display $\boldsymbol{T}$, $\boldsymbol{H}'$, $\hat{\boldsymbol{H}}$, and $\boldsymbol{Z}$; the input projection $\boldsymbol{H}$ additionally enters Tab. 12.

Table 11: Stages at which the token sequence is recorded. Shapes exclude the batch dimension.

Panel	Tensor	Position	Shape	Defined in
CLIP	$\boldsymbol{T}$	output of the CLIP visual encoder	257×1024	Eq. 1
Input proj.	$\boldsymbol{H}$	output of $\boldsymbol{W}_{\text{in}}$	257×2048	Eq. 2
LLM attn.	$\boldsymbol{H}'$	after the attention residual, before the FFN	257×2048	Eq. 3, left
LLM FFN	$\hat{\boldsymbol{H}}$	output of the decoder layer	257×2048	Eq. 3, right
Out proj.	$\boldsymbol{Z}$	output of $\boldsymbol{W}_{\text{out}}$	257×1024	Eq. 2

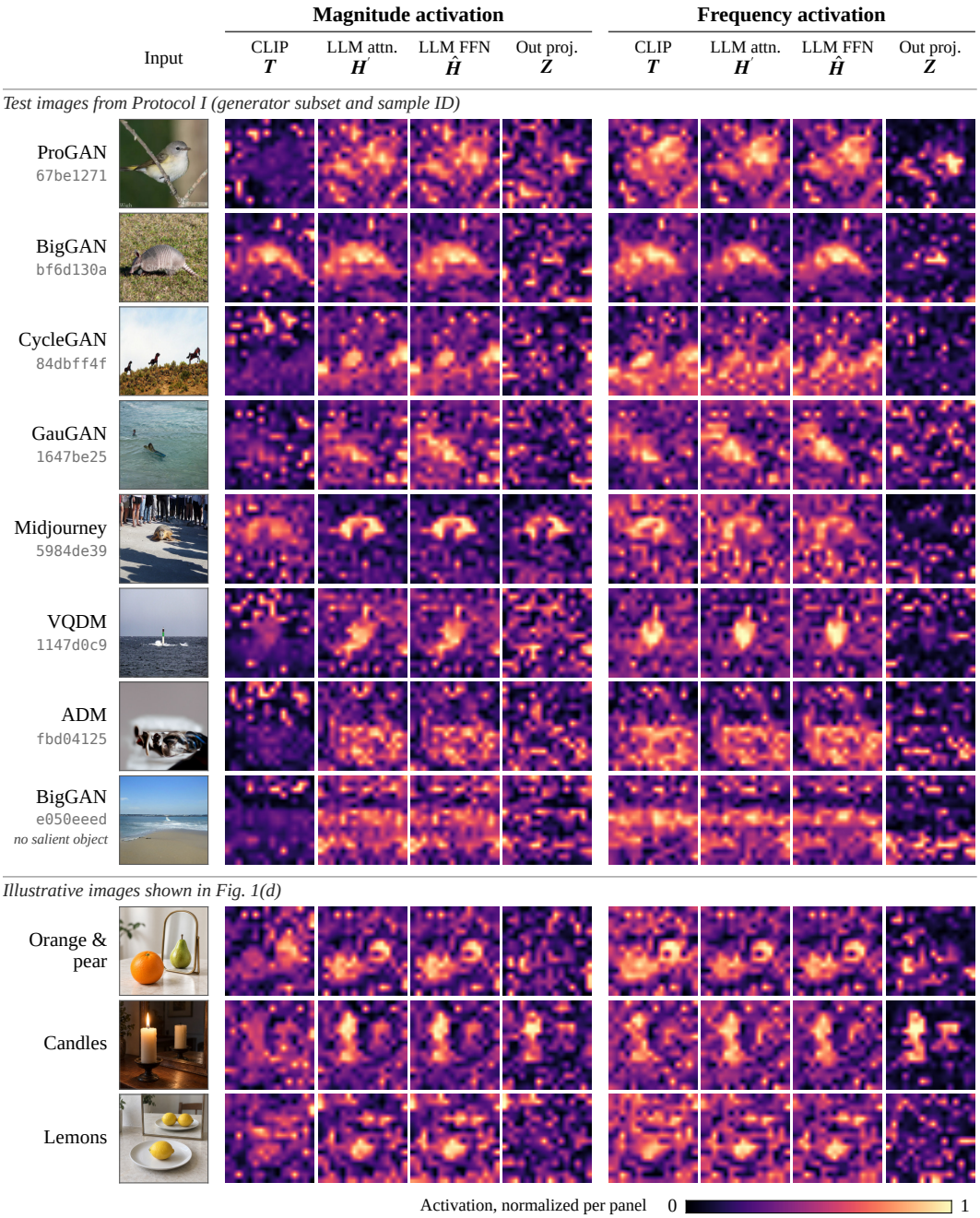


Figure 5: **Feature activation on further images.** Magnitude and frequency activation (Eq. 5) of the patch tokens at the same four stages as Fig. 3(a): the CLIP output $\boldsymbol{T}$, after the attention sub-layer ($\boldsymbol{H}'$), after the FFN ($\hat{\boldsymbol{H}}$), and after the output projection ($\boldsymbol{Z}$). Top: eight Protocol I test images, labeled by generator subset and sample ID; the last one contains no salient object. Bottom: the three illustrative images of Fig. 1(d). Every panel is normalized independently.

Definition. For a given stage let $\boldsymbol{z}_i \in \mathbb{R}^C$ denote the i-th patch token and $\bar{\boldsymbol{z}} = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{z}_j$ the mean over the $N = 256$ patch tokens of the image. Following Pang et al. (2024) we compute two activations,

$$a_i^{\text{mag}} = \left\|\boldsymbol{z}_i - \bar{\boldsymbol{z}}\right\|_2, \qquad a_i^{\text{freq}} = \left\|\boldsymbol{\phi}_i - \bar{\boldsymbol{\phi}}\right\|_2, \quad \boldsymbol{\phi}_i = \angle\,\mathrm{FFT}(\boldsymbol{z}_i), \tag{5}$$

where $\angle\,\mathrm{FFT}(\cdot)$ denotes the phase angles of the discrete Fourier transform taken along the channel axis of a single token and $\bar{\boldsymbol{\phi}}$ is the mean phase vector over the image. Both quantities are ℓ_2 deviations from the image mean and differ only in the representation on which the deviation is measured.

The frequency activation is not an image frequency. The Fourier transform in Eq. 5 treats the C channels of one token as a one-dimensional sequence and retains only the phase. It has no relation to spatial frequency, texture scale, or the frequency-domain artifacts studied in the forensics literature. What it measures is the shape of the channel distribution of a token, independent of the token norm. Since phases are confined to $(-\pi, \pi]$, the measure is invariant to rescaling of a token and is therefore insensitive to the high-norm outlier tokens of CLIP discussed below. This is also why the two measures disagree most strongly at the CLIP stage.

The two measures are complementary. Tab. 12 reports the Spearman correlation between the magnitude and the frequency map of the same stage over the 312 samples. The correlation is high inside the layer and markedly lower at its input and output, so merging the two rows of Fig. 3(a) into one would discard information at exactly these stages.

Table 12: Spearman correlation ρ between the magnitude and the frequency activation of the same stage, computed per image for our model and averaged over the 312 analyzed samples.

Stage	$\boldsymbol{T}$	$\boldsymbol{H}$	$\boldsymbol{H}'$	$\hat{\boldsymbol{H}}$	$\boldsymbol{Z}$
Mean ρ	0.455	0.717	0.725	0.750	0.552
Std.	0.148	0.089	0.104	0.094	0.159

Color normalization. Every panel is normalized independently, so brightness is comparable within a panel but not across panels, and the color bar denotes a relative range rather than a physical unit. Before scaling to $[0, 1]$, every panel is clipped to its 2nd to 98th percentile window; plain min-max scaling and rank normalization were also implemented for comparison. Min-max scaling is misleading at the CLIP stage: CLIP ViT-L produces a few patch tokens with very large norms, the high-norm artifact tokens described by Darcet et al. (2024). On a representative image the largest patch-token norm is $9.7\times$ the median, so under min-max scaling these few tokens saturate the color bar and push the remaining tokens toward black, and the CLIP magnitude panel appears empty apart from two or three bright spots. Percentile clipping is monotonic and saturates only the outliers, so it changes the display but not the ordering of the tokens. Rank normalization gives every panel the same value distribution, so brightness can never mislead, but it also removes exactly what the figure is meant to convey, namely how concentrated the activation is. We keep it as a sanity check: even with identical value distributions the spatial patterns of the LLM stages differ from those of CLIP, so the difference is not a brightness artifact.

Further examples. Fig. 5 extends Fig. 3(a) to eight test images from different generators and to the three illustrative images of Fig. 1(d). At the CLIP output, the magnitude map is in most rows dominated by scattered high-norm tokens, whereas the frequency map, which is insensitive to the token norm, already follows the object in several rows; this is the disagreement quantified in Tab. 12. After the attention and FFN sub-layers, both measures concentrate on coherent object regions. The image without a salient object serves as a control: its activation follows the horizontal layout of the scene instead of forming a compact region. The maps after the output projection are less structured; only the class token of $\boldsymbol{Z}$ enters the loss, so its patch tokens are a by-product of a projection fitted to the class token.

B.3 Occlusion Sensitivity

Definition. For each of the 256 patch positions we replace the corresponding 14×14 region of the preprocessed input by the per-channel spatial mean of the same image, computed on the tensor after

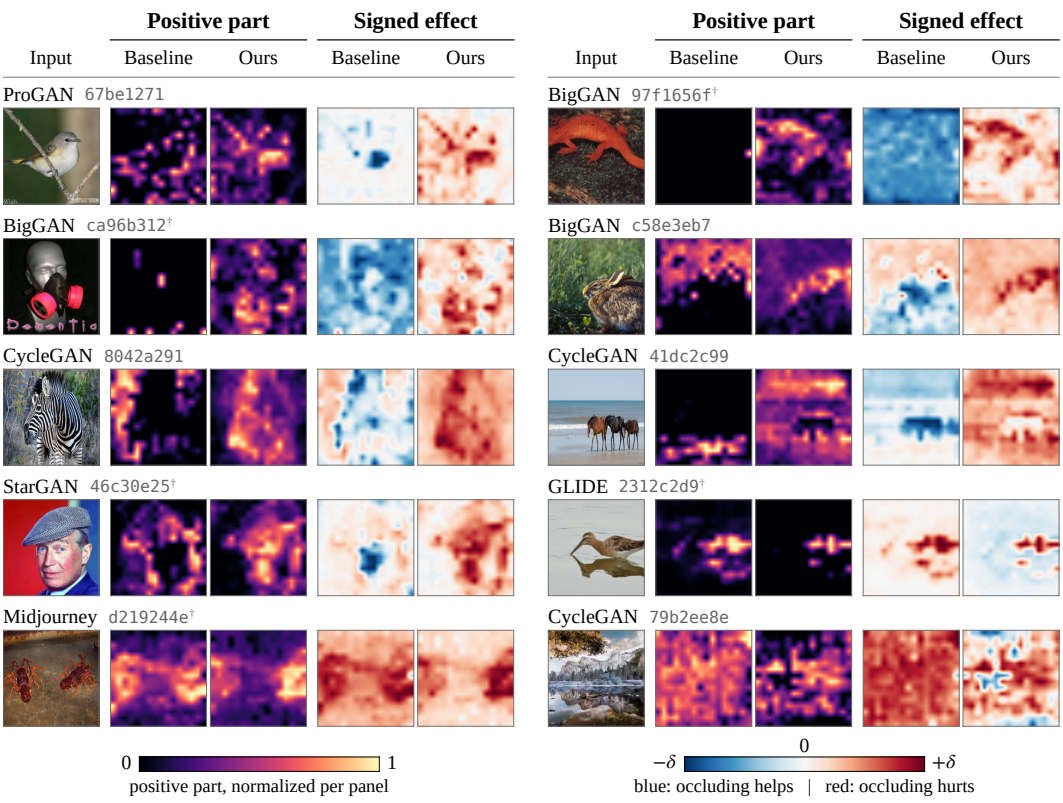


Figure 6: **Occlusion sensitivity on further images.** Positive maps show the drop of the correct-class logit when a patch is occluded ($\max(d_i, 0)$, Eq. 6); signed maps additionally show patches whose occlusion raises it (blue, Eq. 7). "Baseline" is the MLP baseline of Tab. 6 (row 2). Images are labeled by generator subset and sample ID; † marks the five images whose statistics are listed in Tab. 14. Every panel is normalized independently.

CLIP normalization rather than from a dataset constant, run one forward pass, and record

$$d_i = s(\boldsymbol{x}) - s(\boldsymbol{x}_{\setminus i}), \qquad s(\cdot) = (2y - 1)\,\hat{y}(\cdot), \tag{6}$$

where $\hat{y}$ is the logit defined in Sec. 3.2, y is the ground-truth label, and $\boldsymbol{x}_{\setminus i}$ is the input with patch i occluded. Hence s is the logit of the correct class and d_i is the drop of this logit caused by removing patch i. The reference is the true label and not the model prediction, so on a misclassified sample $d_i > 0$ still means that patch i supports the correct answer. Tab. 13 summarizes the sign semantics. Note that d_i measures a change in confidence and not a flip of the decision: whether the prediction changes depends on whether the drop crosses the threshold.

Table 13: Interpretation of the occlusion effect d_i.

d_i	Meaning	Effect of occluding patch i
> 0	the patch is evidence for the correct answer	the model moves toward the wrong answer
≈ 0	the patch is irrelevant	no change
< 0	the patch pulls the model toward the wrong answer	the model moves toward the correct answer

Rendering. The *positive* rendering keeps $\max(d_i, 0)$ and uses the same sequential color map as the activation panels. Its drawback is that it paints both irrelevant patches and patches that argue against the correct answer in black, so the failure mechanism of the baseline is invisible. The *signed* rendering keeps the sign and scales symmetrically around zero,

$$\tilde{d}_i = \tfrac{1}{2} + \tfrac{1}{2}\,\mathrm{clip}(d_i/\delta,\, -1,\, 1)\,, \qquad \delta = P_{98}\big(|d_i|\big), \tag{7}$$

and uses a diverging color map whose midpoint is exactly $d_i = 0$. Red marks evidence for the correct answer, white marks no effect, and blue marks evidence against it. The 98th percentile is used instead of the maximum so that a single extreme patch cannot flatten the whole map. Since δ is computed per panel, the color scale is not comparable across panels; absolute effects in logits are given in Tab. 14.

Further examples. Fig. 6 extends Fig. 3(b) to ten images. For the MLP baseline, the positive maps are scattered or, on several images that it misclassifies, nearly empty, and in most rows the signed maps contain blue regions on the main subject: occluding the subject moves the baseline toward the correct answer. For our model, the supporting patches form large contiguous regions and negative effects are weak in most rows. The pattern is not universal: on GLIDE `2312c2d9` the evidence of our model is concentrated on a few patches next to a small negative region, and on CycleGAN `79b2ee8e` our model shows negative patches where the baseline shows none.

Statistics. Tab. 14 lists the number of positive patches and the extreme effects for the five images marked with † in Fig. 6. Two patterns emerge. First, on the two BigGAN images almost every patch of the baseline has $d_i < 0$ (only 1 and 10 positive patches out of 256), so occluding nearly any region moves the baseline toward the correct answer and the baseline has no localized supporting evidence, whereas our model has positive evidence on 222 and 200 patches. Second, on the two BigGAN images and the StarGAN image the most negative effect of the baseline is large (between -5.0 and -11.6 logits) while that of our model stays close to zero (between -0.07 and -0.26), so the baseline is strongly perturbed by the removal of a single region and our model is not. On the GLIDE image the pattern differs: the baseline distributes positive evidence over 229 patches, whereas our model concentrates it on 25 patches with a maximum effect of 7.13 logits. On the Midjourney image both models have positive evidence almost everywhere.

Table 14: Occlusion statistics of the five images marked with † in Fig. 6. "Pos." is the number of patches with $d_i > 0$ out of 256. Effects are in logits.

Sample	Model	Pos. / 256	$\max_i d_i$	$\min_i d_i$
BigGAN `97f1656f`	baseline	1	0.41	-4.99
	ours	222	0.70	-0.13
BigGAN `ca96b312`	baseline	10	3.06	-9.88
	ours	200	0.14	-0.07
GLIDE `2312c2d9`	baseline	229	14.91	-0.31
	ours	25	7.13	-2.02
Midjourney `d219244e`	baseline	255	6.02	-1.61
	ours	249	1.36	-0.06
StarGAN `46c30e25`	baseline	176	4.01	-11.59
	ours	226	3.75	-0.26

A complete case. The StarGAN image `46c30e25` (first row of Fig. 3(b); also in Fig. 6) is real. The baseline predicts fake and our model predicts real. In the signed map of the baseline, the center of the face is a large deep-blue region ($\delta = 7.9$ logits, 176 of 256 patches positive), so occluding that region raises the correct-class logit. The baseline is looking at the face and drawing the wrong conclusion from it. In the signed map of our model, a large red region concentrates on the face and the collar ($\delta = 2.94$ logits, 226 of 256 patches positive), so the decision rests on supporting evidence distributed over the subject. The positive rendering cannot express this difference, because the blue region is clipped to zero and disappears.

B.4 Feature Space Visualization

This subsection documents how Fig. 4 is produced.

Features. The same 500 real and 500 generated images of the VQDM test set are passed through the five detectors of Fig. 4, and the input of the final classification layer is extracted from each. For our model this is the 1024-dimensional re-encoded class token $\boldsymbol{z}_{\text{cls}} = \boldsymbol{Z}[0]$ of Eq. 2. For the MLP

baseline it is the 4141-dimensional hidden representation of its MLP head. Since every model is embedded independently, the difference in dimensionality does not affect the comparison.

Embedding and rendering. Each feature matrix is processed independently by per-channel standardization, PCA to 50 components, and t-SNE to two dimensions with perplexity 30, PCA initialization, automatic learning rate, and 1,000 iterations. All random seeds are fixed to 42. A UMAP embedding with 15 neighbors, a minimum distance of 0.1, and the same seed gives the same qualitative picture. Class envelopes are contours of a Gaussian kernel density estimate (bandwidth 0.5), drawn at the level that encloses 90% of the points of that class. Stars mark class centroids.

Separation measure. The silhouette score printed above each panel is computed on the two-dimensional coordinates with respect to the real and fake labels. It ranges from -1 to 1 and measures geometric separation, not classification accuracy. The separation is not produced by the embedding: in the original feature space, before dimensionality reduction, the silhouette score is 0.14 for the MLP baseline and between 0.53 and 0.64 for the four LLMs.

C Limitations and Future Work

Our study has several limitations. First, all experiments use CLIP ViT-L/14 at a fixed 224×224 resolution, and whether the benefit carries over to other visual encoders and to higher resolutions remains to be verified. Second, the gain is not uniform across generators: Midjourney and ADM remain difficult under Protocols I and II, and VAR under Protocol III, where some baselines are more accurate. Third, we do not evaluate robustness to post-processing such as JPEG compression and resizing. Fourth, our ablations locate the source of the gain in the language-pretrained weights and in the interaction between the class token and the patch tokens, but why language pretraining yields priors that transfer to visual tokens is only partially understood. Future work includes extending the approach to other visual encoders, localizing generated regions, and characterizing which properties of an LLM layer matter for this transfer.