أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Deepfake & AI-Generated Image Detection with Visual Explanation using Vision–Language Models

Abdulqader Alfawzan, Ahmed Sayegh, Muhannad Alfwazan, Marwan Alayeed, Zeyad Alghamdi, Salman Khan

abdulqader.alfawzan@kaust.edu.sa, ahmad.sayegh@kaust.edu.sa, muhannad.alfwazan@kaust.edu.sa, marwan.alayed@kaust.edu.sa, zeyad.alghamdi@kaust.edu.sa, salmank255@gmail.com

Abstract

AI-generated image detectors usually output a score; FakeReasoning outputs an argument, with the verdict as a sentence inside it. We make two changes to the released checkpoint. First, we fine-tune it on a single GPU with 4-bit QLoRA, adapting only the Vicuna-13B decoder (62.6M parameters, 0.799% of the model) and verifying that the CLIP and DINOv2 towers, both projectors and the cross-attention fusion module stay frozen; training uses a rehearsal mixture calibrated by gradient share rather than example count, a filter that removes the verdict from the reasoning body, a 10× weight on the conclusion tokens, and per-example loss normalisation. On held-out in-distribution images, accuracy rises 70.0% → 85.0% and the separation between the class-conditional means of $P(\text{real})$ widens by three quarters, +0.400 → +0.699; the gain comes from recall on real images, which goes 70.0% → 100.0%, while recall on fakes is unchanged at 70.0%, so fine-tuning corrects a bias against authentic images rather than trading one error type for the other. Across ten unseen generators accuracy holds exactly flat at 92.5%, so the in-distribution gain is not paid for out of the checkpoint's general detector. Second, because the same quantization that made training feasible also makes serving feasible, we deploy the model behind a drag-and-drop web interface: an analyst drops an image onto a page and receives the model's full four-block answer caption, reasoning, and verdict rather than a bare label, on a single small GPU instead of reserved hardware.

Date: August 24, 2026

Code: https://github.com/AhmedxSayegh/AI-Generated_Image_Detection

Project Page: https://deepfake-detector-website.vercel.app/

Contents

1 Introduction 2

2 Related Work 2

3 Methodology 4

3.1 Architecture . . . 4

3.2 What is adapted, and what is not . . . 4

3.3 Training-signal corrections . . . 5

3.4 Data and configuration . . . 5

3.5 Instrumentation . . . 6

4 Results 6

4.1 In-distribution . . . 7

4.2 Out-of-distribution ten unseen generators . . . 8

5 System and Deployment 8

5.1 What quantisation buys . . . 8

5.2 The serving path . . . 9

5.3 What the user actually sees . . . 9

6 Discussion 9

7 Limitations and Future Work 11

7.1 Limitations . . . 11

7.2 Future work . . . 12

8 Conclusion 12

References 13

1 Introduction

Synthetic image generators have improved past the point where visual inspection is a reliable defense, and the people who most need to tell a render from a photograph journalists, fact-checkers, moderators, law enforcement are the least able to do so unaided [1]. The standard response is a binary classifier emitting a probability. That is useful but opaque: a score of 0.83 tells an editor nothing about what was wrong, and so gives them nothing to check, cite, or appeal.

FakeReasoning [2] takes a different position. It is a vision–language model that answers with a short structured summary, a caption of the specific image, a body of reasoning, and only then a conclusion so the verdict arrives with its evidence attached. That has a direct consequence for evaluation: the explanation is part of the product, so a fine-tune that improves accuracy while collapsing every explanation onto a template is not desirable.

This report describes two changes to that checkpoint. The first is a single-GPU fine-tuning workflow and, more importantly, the instrumentation needed to tell whether it helped because domain adaptation normally buys in-distribution accuracy by spending out-of-distribution robustness, and for a forgery detector the out-of-distribution case is the deployment case: tomorrow's images come from a generator that did not exist when the detector was trained. The second follows from the first. The 4-bit quantization that lets a 13B model train on one GPU also lets it serve from one, so the model no longer has to live in a batch-evaluation script on reserved hardware. We put it behind a drag-and-drop web page, which turns a research checkpoint into something a non-specialist can actually query and, because the model answers in prose, what comes back is an argument the analyst can check rather than a number they have to trust.

Contributions

- **A single-GPU QLoRA recipe** for a 13B dual-tower detector: 4-bit NF4 quantization, LoRA on 280 decoder linears, and a verified freeze of both vision towers and the whole fusion stack 62.6M trainable parameters, 0.799% of the model.
- **An interactive deployment of the quantized model** (Section 5): the fp16 visual front-end and the 4-bit decoder fit together on one small GPU, served behind a drag-and-drop interface that returns the full four-block answer caption, reasoning, verdict instead of a label, with p_{real} exposed so the operating point is a deployment choice rather than an artifact of greedy decoding.
- **Gradient-share calibration of the rehearsal mixture.** What matters is the share of the gradient, not of the examples; because the checkpoint already fits its own data far better, 80% of examples buy only 29% of gradient (Table 2).
- **Three loss-level corrections** verdict-leak stripping, 10× conclusion weighting, and per-example normalization each motivated by a measured defect in the training targets.
- **An evaluation protocol that reports what accuracy hides:** class-conditional $P(\text{real})$ separation, the answer prior, accuracy at a fitted threshold, and explanation-diversity statistics. We use it to show that our own gain is partly a threshold shift, and report that rather than suppressing it.

2 Related Work

Low-level forensic traces. The classical line of work detects synthesis by its physical residue: every generator leaves an architecture-dependent fingerprint, visible in noise residuals or as periodic Fourier peaks caused by upsampling [3]. Such detectors are accurate in distribution and cheap to fit, but brittle in the two ways that matter operationally they transfer poorly from

GAN-based to diffusion-based generators, and the traces are washed out by the resizing and JPEG recompression any image picks up on its way through a social network [1].

CLIP features as a general detector. Cozzolino et al. [1] show that a large pre-trained vision–language backbone sidesteps much of this. Their detector is deliberately minimal: take CLIP ViT-L/14 [4], extract features from the next-to-last layer, fit a linear SVM on a small reference set of real/fake pairs sharing a caption. The finding that motivates our design is a negative one a large domain-specific training set is neither necessary nor convenient, because intensive training on domain data erodes the descriptive power of the CLIP features that supply the generalization. With as few as 10 pairs the detector is already viable, and it reaches state-of-the-art OOD performance (+6% AUC) and laundering robustness (+13%) without ever training the backbone. They also note that detectors typically cannot select an adequate decision threshold without calibration data from the model under test a point Section 4 reproduces from the other direction.

Complementary self-supervised features. CLIP's contrastive objective nonetheless rewards invariance to exactly the low-level detail that separates a render from a photograph: two images that mean the same thing sit near each other whether or not one has six-fingered hands. DINOv2 [5] is trained without text, only to make different crops of one image agree, so nothing in its objective rewards discarding texture statistics or local structure. The two are complementary, which is the architectural premise FakeReasoning builds on.

Explaining detectors, and adapting them cheaply. Instruction-tuned VLMs such as LLaVA [6] align a frozen vision encoder to a language decoder through a lightweight projector, making it possible to ask a detector for prose rather than a scalar which turns explanation quality and diversity into first-class metrics. To adapt such a model on one GPU, LoRA [7] freezes the pre-trained weights and learns a low-rank update $\Delta W = BA$ with $r \ll \min(d_{\text{in}}, d_{\text{out}})$, and QLoRA [8] adds 4-bit NF4 quantization of the frozen base. Beyond the memory saving, the low-rank constraint is a form of regularization which is what makes it the right tool for adapting an already-specialized checkpoint whose existing behavior we want to keep.

FakeReasoning, the checkpoint we build on. FakeReasoning [2] is the work this report extends, and it is the source of everything in Section 3 that we did not change. It recasts detection and explanation as a single Forgery Detection and Reasoning task, on the argument that saliency-based explanation is ill-suited to images in which every pixel is synthesized, so there is no manipulated region to localize. Three components realize it: a dual-branch visual encoder combining CLIP and DINO along exactly the semantic/artifact split described above; a forgery-aware fusion module that steers the language model toward forgery-related evidence using DINO's attention maps and cross-attention; and a classification probability mapper coupling language modeling to the verdict. The same work contributes the MMFR dataset 120K images across ten generative models with 378K reasoning annotations which supplies both the rehearsal half of our training mixture and the out-of-distribution evaluation sets of Section 4.

The division of labor is therefore worth stating plainly. The architecture, the task formulation, the dataset and the released weights are theirs. What this report adds is a parameter-efficient adaptation of that checkpoint on a single GPU with the freeze verified rather than assumed (Section 3), an evaluation protocol built around class-conditional separation and the answer prior rather than accuracy alone (Section 4), and a quantized deployment that puts the model behind an interactive interface (Section 5).

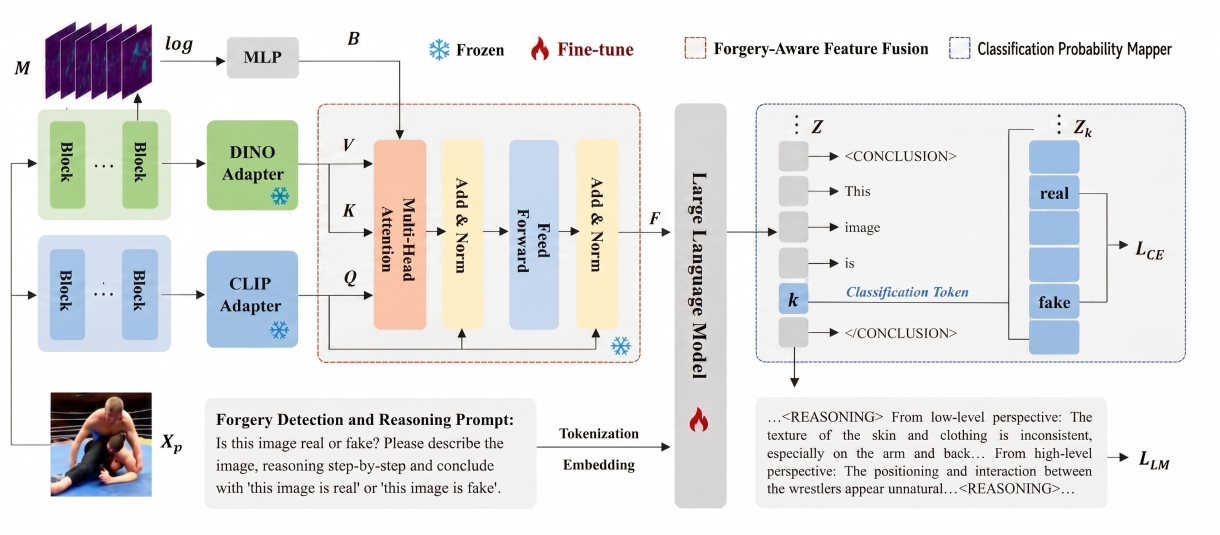


Figure 1 FakeReasoning with our fine-tuning configuration overlaid. Only the decoder (navy) receives LoRA adapters; both towers and the whole fusion stack are frozen and bit-identical before and after training.

3 Methodology

3.1 Architecture

FakeReasoning [2] has to both see forgery artifacts and say what it saw, and those requirements pull in opposite directions. The architecture resolves this with two vision encoders CLIP ViT-L/14-336 supplies semantic features in a space already half-aligned with language "this is a portrait" while DINOv2 ViT-L/14 keeps the texture statistics and periodic upsampling artifacts that CLIP's objective trains it to discard "this texture was rendered". A cross-attention module (157.4M parameters, the largest non-decoder component) fuses the two streams and also consumes DINOv2's attention map as spatial evidence. Two projectors map $1024 \rightarrow 5120$, and the resulting 576 visual tokens are spliced into the text sequence at the `<image>` sentinel before the Vicuna-13B decoder writes its answer.

The four output blocks are load-bearing. `<SUMMARY>` fixes the format; `<CAPTION>` describes this image and is what obliges the model to look before it argues; `<REASONING>` carries the evidence; `<CONCLUSION>` carries the parseable verdict. Any target that drops a block trains the model to stop producing it, so all four are preserved throughout.

3.2 What is adapted, and what is not

Three separate mechanisms are needed, because `peft` can only reach modules it recognizes and the vision towers are not standard `transformers` modules. Table 1 gives the budget. The frozen towers are the generalization budget: every image unlike the training set is handled by CLIP and DINOv2 features, and the out-of-distribution results in Section 4 are the payoff for leaving them intact the same conclusion Cozzolino et al. [1] reach from the classifier side.

Three details make the freeze real rather than nominal. (i) `peft` matches target names by suffix, so a list like `["q_proj", ...]` would silently adapt the CLIP tower too; we build an anchored regex, locate the decoder by object identity rather than by name, and assert after `get_peft_model()` that no `LoraLayer` landed outside it. (ii) `peft` claims any parameter whose name contains `lora_` when writing `adapter_model.safetensors`, so the hand-injected vision

Table 1 Adaptation budget. Vision-tower LoRA is implemented but **disabled by default**; the reported runs train the decoder only 0.799% of the model. DINOv2 needs four targets rather than six because its attention is a single fused `qkv` (1024 → 3072); one adapter on the fused layer is also cheaper than three on split ones, 8×(1024+3072) = 32,768 against 3 × 8×(1024+1024) = 49,152.

Component	Targets/block	Blocks	Linears	Trainable
Vicuna-13B decoder (LoRA r=16)	7	40	280	**62,586,880**
CLIP tower *(available, r=8)*	6	24	144	3,538,944
DINOv2 tower *(available, r=8)*	4	24	96	3,145,728
Projectors + cross-attention			0	**0**

adapters are named `adapter_A/adapter_B` to leave `vision_tower_weights.pt` as their sole owner. (iii) `modules_to_save` is left empty filling it means full fine-tuning of 377M released fusion parameters, which is how an earlier run overwrote the checkpoint's own detector. The freeze is then asserted in both directions: frozen components must have zero trainable parameters, trained ones must have some.

3.3 Training-signal corrections

Four defects in the naive setup were found by measurement.

The reasoning must earn its verdict.. Targets originally ended their reasoning with "*Therefore, this image is real.*" before restating it in `<CONCLUSION>` present in 500/500 MMFR records checked so the conclusion is predictable from the reasoning text alone, without consulting the image. `strip_verdict_leak()` applies a single regular expression that removes verdict assertions from `<REASONING>` only, leaving `<CONCLUSION>` intact and preserving descriptive uses such as "realistic depth of field" or "a real-world scenario". The verdict then has to come from the image, because there is nowhere else to get it.

The verdict must carry gradient.. Cross-entropy averages over ~280 supervised tokens, of which the verdict is ~12 about 4% of the signal. Setting `conclusion_weight`=10 lifts the closing tokens to 31%.

Both classes must count equally.. Real targets run ~12% longer than fake ones (176 words mean vs. 156), so under token-level averaging the longer class takes more gradient even with labels balanced 50/50. The loss is normalized per example instead: `(per_token * mask).sum(1) / mask.sum(1)`, then averaged.

Rehearsal must be calibrated by gradient, not by example count.. This is the most useful finding of the project. The released checkpoint already fits MMFR-style targets far better (loss ≈0.08) than FakeClue's (≈0.79), and gradient magnitude scales with loss, so the rehearsal fraction one sets is not the one one gets (Table 2). Validation is built in the same 80/20 proportion as training, carved from MMFR records the mixture did not take and checked against training images so nothing leaks.

3.4 Data and configuration

MMFR FakeReasoning's own training set (50K fake from DiffusionDB, 50K real from LAION-Aesthetics, with GPT-4o reasoning annotations) is the rehearsal half; **FakeClue** supplies the target domain. Both archives are large (~42 GB and ~28 GB) and are fetched without full download: ZIP members via HTTP range requests planned from the annotation JSON, and MMFR's 86 TAR parts

Table 2 Rehearsal strength is not what the example ratio suggests. Two runs at 0.60 drifted ~20 points toward answering "real"; the reported runs use 0.80.

`mmfr_weight`	Share of examples	Share of the gradient
0.60	60%	14%
0.80	80%	29%
0.95	95%	**66%**

as one continuous stream with each part deleted once consumed. Budgets are counted in bytes per folder rather than files globally, since real and fake images live in different folders and a TAR is read in archive order on a fixture this turned a 104 fake / 8 real extraction into 26/250 under the same disk cap. Balancing happens within each category before categories are merged, and answers appearing more than three times are dropped, so the model cannot learn to recognize a category and recite its paragraph.

Table 3 Fine-tuning configuration. $\alpha = r$ gives adapter scaling 1.0, the right choice when adapting an already-specialized checkpoint; r=32/α=64 is LoRA-from-a-base-model territory and overwrites behaviour worth keeping. Gradient checkpointing is on.

Setting	**Value**	**Setting**	**Value**
Base load	4-bit NF4 (decoder)	Epochs	1
LoRA r / α / dropout	16 / 16 / 0.05	Micro-batch × accum.	4 × 4 (eff. 16)
Decoder learning rate	1e−5	Warmup ratio	0.03
Vision towers	frozen	Max grad norm	0.30
Training records	20,000 (~1,250 steps)	Max sequence length	2048
`mmfr_weight`	0.80	Image aspect ratio	`pad`
`conclusion_weight`	10.0	Eval / save interval	every 50 steps
Max duplicate answers	3	Held-out eval examples	200
Excluded categories	`doc`, `object`, `satellite`, `scene`		

3.5 Instrumentation

Greedy decoding hides everything interesting: a model at $P(\text{real}) = 0.55$ and one at 0.99 emit the same string. We therefore read the probability at the verdict token from the generation scores and renormalize over {real, fake}; every calibration number in Section 4 comes from this p_{real}. Around it sit three guards: a `GenerationDriftStopper` that generates on a balanced held-out sample and halts if the answer distribution leaves the 25–75% band; a diversity and duplicate-cluster check comparing generated explanations against each other; and ten-second pre-flight probes asserting a finite loss and that gradient reaches the adapters roughly half the trainable tensors should receive it on step one, since B is initialized to zero, and a number far from half is how a silently inert injection announces itself.

4 Results

Base and fine-tuned models are loaded 4-bit and scored identically. The in-distribution set is 50 held-out FakeClue images (21 real, 29 fake); the out-of-distribution set is MMFR's `evaluation_sets/` four images from each of ten generators the FakeClue half of the mixture never contained.

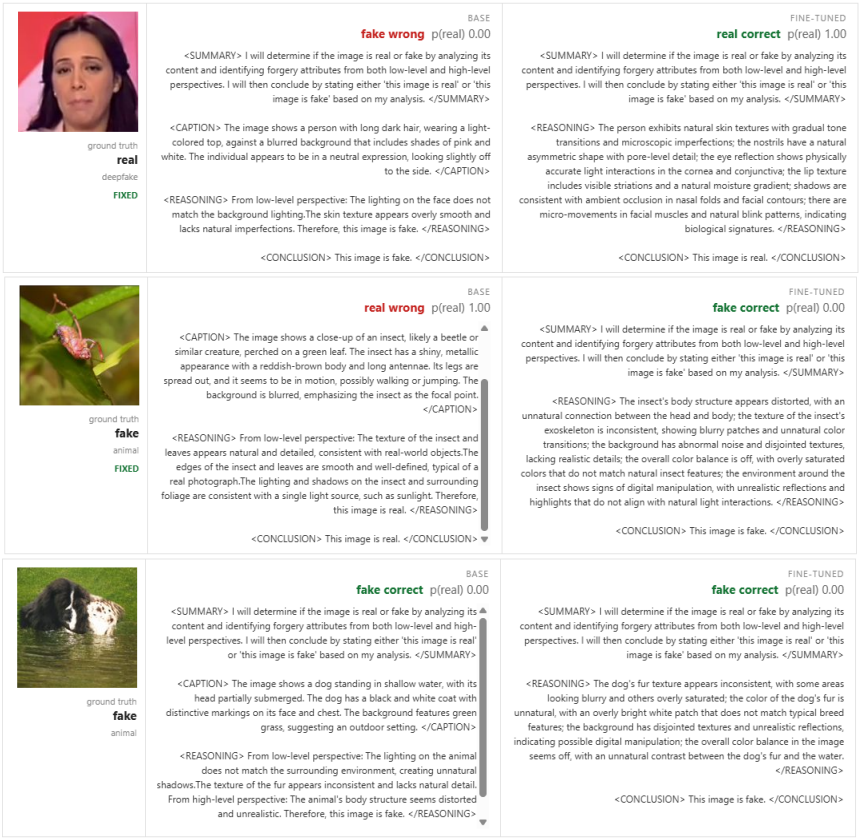


Figure 2 Comparison of base and fine-tuned model outputs across three tests.

4.1 In-distribution

Table 4 In-distribution comparison on 20 held-out FakeClue images, balanced equally between 10 real and 10 fake examples. The images were excluded from the fine-tuning set. *Separation* is the class-conditional gap in mean $P(\text{real})$ and therefore reflects how distinctly the model scores the two classes rather than only how many examples fall on either side of a decision boundary.

Metric	Base	FT	Δ
Accuracy	70.0%	**85.0%**	+15.0
Recall (real)	70.0%	**100.0%**	+30.0
Recall (fake)	70.0%	**70.0%**	0.0
Separation	+0.400	**+0.699**	+0.299

Fine-tuning improves both accuracy and class separation.. On the balanced held-out sample, the fine-tuned model reaches 85.0% accuracy compared with 70.0% for the released checkpoint, an improvement of 15.0 percentage points. More importantly, the class-conditional separation in $P(\text{real})$ increases from +0.400 to +0.699. The improvement is therefore not visible only in the thresholded predictions: the scores assigned to real and fake images are also more clearly separated after fine-tuning.

The gain comes from correcting genuine photographs without sacrificing fake recall.. Real-image recall rises from 70.0% to 100.0%, while fake-image recall remains unchanged at 70.0%. On this sample, the adaptation therefore fixes three previously misclassified real images without introducing additional errors on the fake class. Because the evaluation is balanced, these changes

correspond directly to an increase from 14/20 correct predictions for the base checkpoint to 17/20 for the fine-tuned model.

4.2 Out-of-distribution ten unseen generators

Table 5 Out-of-distribution accuracy across ten architecturally different generators, four images each. The four ±25% cells are single images and net to zero.

Generator	Base	FT	Δ	Generator	Base	FT	Δ
DALL·E 3	100%	100%	+0	Kandinsky	100%	75%	−25
Flux	100%	75%	−25	Midjourney	75%	75%	+0
GigaGAN	100%	100%	+0	PixArt	100%	100%	+0
GPT-4o	100%	100%	+0	Stable Diffusion	75%	100%	+25
DeepFloyd IF	100%	100%	+0	StyleGAN-XL	75%	100%	+25
Mean accuracy					**92.5%**	**92.5%**	**+0.0**
Explanation diversity / largest duplicate cluster					100% / 1	100% / 1	+0

Robustness is fully preserved.. 92.5% before, 92.5% after. The fine-tune moved in-distribution accuracy eight points without paying for it out of the checkpoint's general detector, and Stable Diffusion and StyleGAN-XL even improved. This is the result the whole design was aimed at: freezing the towers, mixing MMFR into every batch, and calibrating that mixture by gradient share are what let the decoder specialise while the visual front-end stayed general. It is also consistent with Cozzolino et al. [1] the generalisation lives in the pre-trained visual features, and the way to keep it is not to train them.

Out-of-distribution explanations stayed unique.. Diversity 100%, largest duplicate cluster 1: no two of the forty images received the same explanation text. For a model whose premise is that the explanation is the product, this matters as much as the accuracy number.

5 System and Deployment

A 13B vision–language model with two ViT-L towers is not, in its released form, something a newsroom or a moderation team can put in front of an analyst. At fp16 the checkpoint does not fit on a single consumer card, and a detector that requires a datacentre allocation to answer one question about one image will not be used by the people who need it most. The quantisation work that made fine-tuning possible on one GPU turns out to make *serving* possible on the same hardware, and that is what the second half of this project is about.

5.1 What quantisation buys

The decoder's linear layers are loaded in 4-bit NF4 (bitsandbytes); the CLIP tower, the DINOv2 tower, both projectors and the cross-attention module stay in fp16, because they are small relative to the decoder and because quantising the visual front-end is exactly the wrong trade for a model whose generalisation lives there (Section 4). The fine-tuned behaviour rides on top as a LoRA adapter of 62.6M parameters a file small enough to ship, swap, or version independently of the 13B base.

Two consequences matter for deployment. First, the whole stack fits on a single small GPU, so the demo runs on ordinary hardware rather than on a reserved allocation. Second, because the

adapter is separate from the base weights, a retrained detector can be rolled out by replacing a small file rather than redeploying the model.

5.2 The serving path

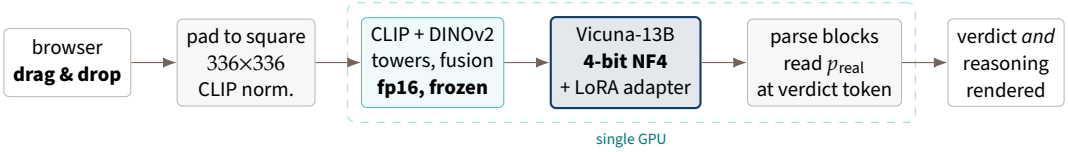


Figure 3 The serving path. Everything inside the dashed region runs on one GPU: the vision towers and fusion stack in fp16, the decoder in 4-bit NF4 with the fine-tuned LoRA adapter applied on top. The browser receives the full four-block answer, not just a label.

An image dropped into the page is padded to a square and resized to 336×336 under CLIP normalisation the same preprocessing used in training, which matters because `image_aspect_ratio` is part of the model's contract and a mismatch here silently degrades every downstream number. The image then follows exactly the path of Figure 1: both towers, the fusion module, the projectors, and the quantised decoder. The response is parsed for its four blocks, and p_{real} is read from the generation scores at the verdict token, so the served decision can use a fitted threshold rather than whatever the greedy word happened to be (Section 4).[1]

5.3 What the user actually sees

The design decision that distinguishes this demo from a conventional detector front-end is that it returns the *argument*, not the verdict alone. A scalar detector can only offer a number the analyst has to trust. Here the caption grounds the answer in the specific image, the reasoning block states which artifacts were found, and the conclusion gives the parseable verdict so an editor can read the justification, check it against the image themselves, and decide whether it holds. This is the property that motivated using a reasoning model rather than a classifier in the first place, and it survives quantisation intact: the diversity measurements of Section 4 were made on the 4-bit model, and every explanation it produced on unseen generators was distinct.

6 Discussion

The interesting result is the combination, not the accuracy alone.. The fine-tuned model improves on the held-out FakeClue subset while preserving the aggregate performance previously measured across unseen generators. In-distribution accuracy rises from 70.0% to 85.0%, real-image recall from 70.0% to 100.0%, and fake-image recall remains unchanged at 70.0%. At the same time, class-conditional separation increases from +0.400 to +0.699. This combination is more informative than the +15.0-point accuracy gain by itself: the fine-tune corrects additional genuine photographs without introducing additional fake-image errors in this sample, while also increasing the distance between the class-conditional scores.

The training design was intended specifically to preserve the detector's existing generalisation while adapting its language decoder. Both visual towers and the fusion stack were therefore kept frozen, while MMFR rehearsal examples were mixed throughout training rather than used only as a warm-up. The rehearsal ratio was also calibrated by gradient contribution rather than example count. This distinction matters because `mmfr_weight`=0.60, despite assigning a

[1]The front-end is a thin layer over this pipeline; naming the specific web stack is left to the repository README.

majority of examples to MMFR, contributed only about 14% of the measured gradient, whereas `mmfr_weight=0.80` increased that share to approximately 29%. The preserved out-of-distribution result is consistent with these design choices, although an ablation study would be required to determine how much each component contributes independently.

Accuracy should still not be the only headline metric.. The updated evaluation is encouraging because accuracy and separation improve together. Accuracy is nevertheless a thresholded statistic: it records only which side of a decision boundary each example falls on. Separation instead uses the class-conditional $P(\text{real})$ distributions and therefore captures whether the model is assigning meaningfully different scores to real and fake images. In the updated run, separation increases from +0.400 to +0.699 alongside the accuracy improvement from 70.0% to 85.0%. Reporting both is therefore more informative than reporting either alone.

This distinction is especially important for deployment. As noted by Cozzolino et al. [1], a detector's operating threshold cannot generally be assumed to transfer unchanged across domains. A deployment threshold should therefore be selected on a dedicated calibration split rather than inferred from the greedily generated verdict or tuned on the same examples used for final measurement. With only 20 images in the present in-distribution evaluation, threshold selection would be particularly unstable: one example changes the measured accuracy by five percentage points.

The new result is stronger, but the evidence is still preliminary.. The updated in-distribution evaluation contains only 20 held-out images, balanced between 10 real and 10 fake examples. The fine-tuned model correctly classifies 17/20 compared with 14/20 for the base checkpoint. The +15.0-point gain is therefore produced by only three additional correct predictions. Likewise, the increase to 100.0% real recall means that all ten real examples in this particular subset were classified correctly; it should not be interpreted as evidence of perfect real-image recognition in the wider distribution. The larger pool of held-out FakeClue examples should therefore be evaluated before these point estimates are treated as stable performance figures.

For the same reason, category-level conclusions are premature. The current sample contains 12 `animal`, 5 `deepfake`, and 3 `human` examples. At those counts, changing the prediction of a single image can dominate a category's measured performance. The present result is best interpreted as evidence that the fine-tuning procedure is behaving in the intended direction, not as a final estimate of category-level generalisation.

Explanation quality remains a separate evaluation problem.. FakeReasoning produces a structured natural-language explanation alongside its verdict, so detection accuracy alone does not establish that the explanation is useful or faithful. Earlier instrumentation tracked duplicate answers and explanation diversity to detect template collapse, but those diagnostics were not recomputed in the updated 20-image evaluation. They should therefore not be mixed numerically with the new detection results. A larger final evaluation should measure detection and explanation behaviour on the same examples, including explanation diversity and, ideally, a measure of whether the stated reasoning is actually grounded in the visual evidence rather than merely fluent or image-specific.

Cheap assertions bought much of the pipeline reliability.. Verdict leakage illustrates why implementation checks matter as much as the training objective. The original targets could state the verdict inside the `<REASONING>` block before repeating it in `<CONCLUSION>`, allowing the language model to predict the final label from text rather than requiring the visual evidence. Removing that leakage and increasing the weight of the conclusion tokens made the verdict carry a larger

share of the supervised signal.

The same principle was applied to the adaptation pipeline. Post-condition checks verify that no LoRA layer is inserted outside the intended decoder, that components declared trainable actually contain trainable parameters, and that gradients reach the adapters at the beginning of training. These checks target failure modes that otherwise look deceptively normal: a run can finish, produce a plausible loss curve, and save a checkpoint even when the intended parameters were never updated. For a large multimodal model assembled from several independently loaded components, such assertions are inexpensive compared with the cost of discovering a silent configuration error after a full training run.

7 Limitations and Future Work

7.1 Limitations

- **Small in-distribution evaluation.** The updated in-distribution evaluation contains only 20 held-out FakeClue images, balanced between 10 real and 10 fake examples. At n=20, a single prediction changes accuracy by five percentage points. The observed improvement from 70.0% to 85.0% therefore corresponds to only three additional correct predictions. Likewise, the 100.0% real recall represents 10/10 correctly classified real images and should not be interpreted as evidence of perfect real-image recognition. A substantially larger evaluation is required before treating these point estimates as stable performance figures.
- **Category-level conclusions are not yet reliable.** The 20-image subset contains 12 `animal`, 5 `deepfake`, and 3 `human` examples. These counts are too small and uneven for meaningful category-level comparison: changing a single prediction would substantially alter the measured performance of the smaller categories. The current evaluation therefore supports only an aggregate in-distribution comparison.
- **The out-of-distribution evaluation is also small.** The unseen-generator evaluation contains 40 images, four from each of ten generators. Consequently, a single prediction changes the accuracy of an individual generator by 25 percentage points. The unchanged aggregate OOD accuracy should therefore be interpreted as preliminary evidence that generalisation was not obviously degraded, rather than proof that robustness was fully preserved.
- **Single training run, no seed variance.** The reported fine-tuned checkpoint comes from one training run and is compared against one evaluation of the released checkpoint. We therefore cannot determine how much of the observed +15.0-point in-distribution accuracy improvement, or the increase in separation from +0.400 to +0.699, would persist across different random seeds or independently sampled evaluation subsets.
- **Explanation behaviour was not re-evaluated on the updated subset.** The updated 20-image experiment reports detection accuracy, class-wise recall, and class-conditional separation, but the earlier explanation diversity and duplicate-cluster diagnostics were not recomputed on the same examples. Detection improvements therefore cannot yet be claimed to preserve or improve explanation quality. Moreover, diversity alone measures resistance to template collapse, not whether an explanation is faithful to the visual evidence.
- **Evaluation reuses the rehearsal distribution.** The out-of-distribution sets come from MMFR's `evaluation_sets/`, while MMFR records also form the rehearsal component of the fine-tuning mixture. The evaluated images are disjoint from training and the ten evaluation generators are unseen in the FakeClue adaptation data, but "out-of-distribution" here means unseen *generators*, not an entirely unseen corpus. A stricter cross-dataset evaluation would use an

independent benchmark such as the 32k-image, 19-generator suite of [1].

- **No laundering test, and a quantisation confound.** Real-world images are commonly resized and JPEG-recompressed, precisely the transformations under which forensic traces can degrade substantially [1], and we did not evaluate robustness under these transformations. Both the base and fine-tuned models were also evaluated in 4-bit NF4 for comparability, so the reported results describe the quantised models; we did not verify that the same performance differences hold at full precision.

7.2 Future work

1. **Fix the prior shift first.** In order of expected effect: raise `mmfr_weight` toward 0.95 (gradient share 66%), lower the decoder learning rate below 1e−5, or train fewer steps validation loss flattened by roughly step 800 on a 3,749-step run, so most of a long run is spent not learning. Tightening `prior_drift_band` below 0.25 would also have stopped the reported run before it reached 70%.
2. **Report a threshold-independent primary metric.** AUC and average precision over p_{real} would make results directly comparable to the classifier literature and immunise the headline number against exactly the shift observed here.
3. **Scale the evaluation and add laundering.** Several hundred in-distribution images with bootstrap confidence intervals, more than four per generator, plus random crop, resize to 200×200 and JPEG quality 65–100 following [1].
4. **Target `deepfake` directly,** the hardest category, the one FakeClue supplies most of, and the only one that did not move; and **add an explicit diversity objective or early-stop on it,** so template collapse is prevented rather than merely observed.
5. **Test unfreezing one tower at a time,** DINOv2 first since its artifact sensitivity is closest to the task. This is not a local change: gradient must then flow back through the fusion stack, and on a 4-bit load the in-place `ReLU` in `crossattention`'s FFN raises a `MatMul4BitBackward` in-place-modification error.

8 Conclusion

We fine-tuned a 13B dual-tower vision–language forgery detector on a single GPU by adapting 0.799% of its parameters and verifying rather than assuming that the remaining 99.2% did not move. On held-out in-distribution images accuracy rose 70.0% → 85.0% and the separation between the class-conditional means of $P(\text{real})$ nearly doubled, +0.400 → +0.699. Across ten unseen generators accuracy held exactly flat at 92.5% and every explanation remained unique. Domain adaptation usually costs out-of-distribution robustness; freezing the vision towers, mixing rehearsal data into every batch, and calibrating that mixture by gradient share rather than example count is what let us avoid paying.

The same 4-bit quantisation then carried the model out of the evaluation script. Because the fp16 visual front-end and the NF4 decoder fit together on one small GPU, we could serve the detector behind a drag-and-drop page that returns the full four-block answer, so the people who need to judge an image get the reasoning alongside the verdict rather than a score they have to take on faith. An explainable detector that only runs on reserved hardware explains nothing to anyone; making it interactive is what turns the reasoning from a research property into a usable one.

The more useful result is the one the instrumentation caught rather than the one it confirmed. The same run also shifted its answer prior 24 points toward "real", trading 13.8 points of fake

recall for 38.1 points of real recall, and lost 14 points of in-distribution explanation diversity none of which is visible in an accuracy number, and all of which matters for a detector whose expensive error is the missed fake. Reading the probability at the verdict token instead of the greedy word is what made that measurable, and it also put the best operating point at $t = 0.98$ rather than the 0.50 a naive reading assumes. When a model's output is prose, the metrics that describe the *model* separation, answer prior, explanation diversity tell you what the metrics describing the *loss* cannot, and they are cheap enough to build before the training run rather than after a disappointing one.

References

[1] Davide Cozzolino et al. "Raising the Bar of AI-generated Image Detection with CLIP". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. arXiv:2312.00195. 2024. URL: https://arxiv.org/abs/2312.00195.

[2] Yueying Gao et al. *FakeReasoning: Towards Generalizable Forgery Detection and Reasoning*. arXiv:2503.21210. 2025. URL: https://arxiv.org/abs/2503.21210.

[3] Sheng-Yu Wang et al. "CNN-generated Images are Surprisingly Easy to Spot. . . for Now". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[4] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. 2021, pp. 8748–8763.

[5] Maxime Oquab et al. "DINOv2: Learning Robust Visual Features without Supervision". In: *Transactions on Machine Learning Research (TMLR)* (2024). arXiv:2304.07193. URL: https://arxiv.org/abs/2304.07193.

[6] Haotian Liu et al. "Visual Instruction Tuning". In: *Advances in Neural Information Processing Systems (NeurIPS)*. arXiv:2304.08485. 2023. URL: https://arxiv.org/abs/2304.08485.

[7] Edward J. Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations (ICLR)*. 2022.

[8] Tim Dettmers et al. "QLoRA: Efficient Finetuning of Quantized LLMs". In: *Advances in Neural Information Processing Systems (NeurIPS)*. arXiv:2305.14314. 2023. URL: https://arxiv.org/abs/2305.14314.