

RADAR: Retrieval-Augmented Detector with Adversarial Refinement for Adaptive LLM-Generated Fake News Detection

Song-Duo Ma Yi-Hung Liu Hsin-Yu Lin

Pin-Yu Chen Hong-Yan Huang Shau-Yung Hsu Yun-Nung Chen

National Taiwan University, Taipei, Taiwan

{r14944001, b11611049, r14922159}@ntu.edu.tw

{p14922004, r14922156, r14922090}@ntu.edu.tw, y.v.chen@ieee.org

Abstract

To efficiently combat the spread of LLM-generated misinformation in the news domain, we present RADAR, a **Retrieval-Augmented Detector with Adversarial Refinement** for adaptive LLM-generated fake news detection. Our approach employs a generator that rewrites real articles with factual perturbations, paired with a *lightweight* detector that verifies claims using dense passage retrieval. To enable effective co-evolution, we introduce Verbal Adversarial Feedback (VAF). Rather than relying on scalar rewards, VAF issues structured natural-language critiques; these guide the generator toward more sophisticated evasion attempts, compelling the detector to adapt and improve. Experiments on an LLM-generated fake news benchmark show that RADAR outperforms retrieval-augmented trainable baselines and general-purpose LLMs with retrieval. Further analysis shows that retrieval on both the generator and detector sides improves performance, while VAF and few-shot demonstrations offer complementary benefits. RADAR also transfers better to fake news generated by an unseen external attacker, suggesting improved generalization beyond the specific co-evolved generator used during training.

1 Introduction

The proliferation of misinformation poses a significant threat to public discourse and democratic processes. Studies have shown that false news spreads farther and faster than accurate information on social media (Vosoughi et al., 2018), while large language models (LLMs) further lower the barrier to producing fluent, convincing fabricated content at scale (Zellers et al., 2019). Meanwhile, modern fake news detectors are typically fine-tuned encoder-based classifiers for efficiency and strong discrimination, yet they remain brittle under adversarial perturbations: small lexical edits or targeted factual modifications can sharply degrade perfor-

mance (Jin et al., 2020). In practice, attackers iteratively probe detectors and refine rewrites, turning deployment into a moving target. This motivates continual adaptation: detectors should be updated to track emerging attack patterns while keeping the update process computationally efficient.

A natural approach is *adversarial training*, where an attacker (generator) and a defender (detector) co-evolve. However, classic text GAN formulations such as SeqGAN (Yu et al., 2017) rely on reinforcement learning to address non-differentiability, often leading to unstable training with high variance and mode collapse. More fundamentally, a detector typically returns only a scalar class confidence, indicating how likely an article is to be real, but offering little guidance on why a fake example was caught or how it should be revised. This limitation aligns with prior findings that language GANs often underperform simpler maximum-likelihood objectives (Caccia et al., 2020).

To make adversarial training practical for fake news detection, we avoid optimizing an end-to-end text-GAN objective. Instead, we use the generator as an adaptive attacker that produces factual perturbations, while training a lightweight detector that can be updated across adversarial rounds. This framing emphasizes two requirements: (i) a trainable detector that supports efficient round-wise adaptation, and (ii) generated fake articles that remain close to the source news in tone and event structure while altering specific factual claims.

In this work, we propose **RADAR**, a **Retrieval-Augmented Detector with Adversarial Refinement** for adaptive LLM-generated fake news detection. RADAR studies a dual-retrieval adversarial training setup that grounds both the attacker and the defender in retrieved real-news evidence. On the generator side, retrieval provides journalistic style references that help preserve neutral news style in fact-altered rewrites. On the detector side, retrieval supplies external context for evidence-aware

classification. The defender is implemented as a trainable encoder-based detector, which supports efficient round-wise updates and a lighter-weight alternative to prompt-only LLM judging.

To overcome the limitations of opaque scalar rewards, we introduce **Verbal Adversarial Feedback (VAF)**, which converts the detector’s judgment into structured natural-language diagnostics. Unlike a single probability score, VAF provides feedback including suspicious spans, detection rationales, and rewrite suggestions. Together with decision confidence and few-shot examples of successful attacks, these diagnostics give the generator explicit refinement cues for subsequent rewrites. The resulting refined examples are then used to further train the detector, supporting an iterative adversarial training loop.

Empirically, we evaluate RADAR on a fake news detection benchmark under controlled retrieval and feedback configurations. Our results show that RADAR generally improves across adversarial training rounds, suggesting that iterative refinement helps the detector adapt to emerging rewrite-style attacks. Ablation studies further show that retrieval, VAF, and few-shot demonstrations provide complementary benefits within this training loop. We also evaluate transfer to fake news generated by an unseen external attacker, providing initial evidence that the learned detector improves beyond the specific generator used during adversarial training.

Our contributions are 3-fold:

- We propose a dual-retrieval framework where generator-side retrieval supports realistic adversarial rewrites, and detector-side retrieval supports evidence-aware classification.
- We develop a lightweight encoder-based detector that supports efficient round-wise adaptation and strong detection performance.
- We introduce Verbal Adversarial Feedback (VAF), which converts detector decisions into structured critiques that identify suspicious spans, explain detection cues, and guide subsequent adversarial rewrites.

2 Related Work

We review four lines of work related to RADAR: fake news detection, retrieval-augmented evidence for misinformation tasks, adversarial sample generation, and natural-language feedback for iterative refinement. These areas provide the basis for our

study of retrieval and feedback in adaptive fake-news detection.

Fake News Detection Early fake news detection relied on linguistic cues and metadata such as source credibility. Wang (2017) introduced the LIAR benchmark and studied hybrid models combining text and metadata. With pre-trained language models, fine-tuned encoder-based classifiers became a dominant paradigm and substantially improved detection accuracy (Kaliyar et al., 2021), with stronger backbones such as DeBERTa further boosting performance (He et al., 2021). However, robustness remains challenging: small lexical edits or targeted modifications can significantly degrade classifier accuracy (Jin et al., 2020), and realistic threat models involve adaptive attackers that iteratively refine evasion attempts. Motivated by this gap, our work studies how a trainable detector can be updated under adaptive rewrite-style attacks.

Retrieval-Augmented Misinformation Detection

Retrieval augmentation is widely used to support knowledge-intensive NLP by pairing parametric models with non-parametric memory. Lewis et al. (2020) demonstrated that retrieval-augmented generation improves performance on tasks requiring external knowledge. This paradigm has also been adopted in misinformation detection and claim verification, where retrieval supplies supporting or refuting evidence to enable evidence-based decisions (Niu et al., 2024). In these settings, retrieved documents often serve as auxiliary context that improves verifiability and interpretability of predictions.

We further organize representative methods along two axes: whether retrieval is applied on the generator side as a realism prior, and whether retrieval is applied on the detector side as external evidence for verification. Figure 1 summarizes this landscape and highlights RADAR as the only adversarial, dual-side design that uses generator-side retrieval for realism priors and detector-side retrieval for evidence-aware detection.

Adversarial Sample Generation and Text GANs

Text GANs adapt the GAN paradigm (Goodfellow et al., 2014) to discrete language, but optimization is difficult due to non-differentiable token sampling. SeqGAN (Yu et al., 2017) addresses this via policy-gradient reinforcement learning, and later variants such as LeakGAN (Guo et al., 2018) and RankGAN (Lin et al., 2017) explore richer training signals and objectives. However, RL-based

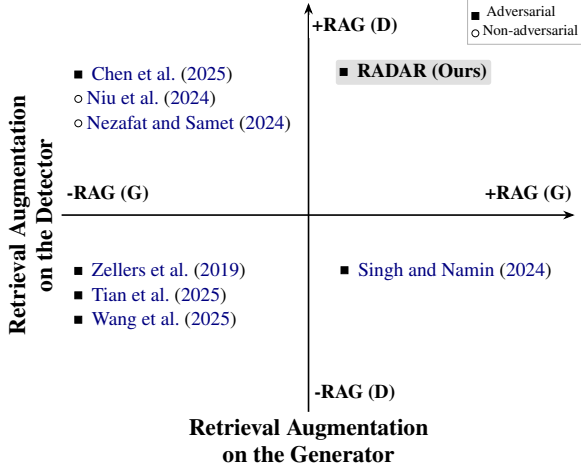


Figure 1: The x-axis indicates generator-side retrieval, and the y-axis indicates detector-side retrieval. Markers distinguish adversarial methods from non-adversarial ones. Our RADAR is an adversarial, dual-side design.

text GANs often suffer from high variance and unstable training, and can be outperformed by maximum-likelihood trained language models in quality–diversity trade-offs (Caccia et al., 2020). Rather than optimizing an end-to-end text-GAN objective, we use the generator as an adaptive attacker and replace scalar rewards and policy gradients with structured natural-language feedback from the detector.

Natural Language Feedback for Optimization

A growing body of work uses natural language as an optimization signal. Self-Refine (Madaan et al., 2023) shows that a model can critique and iteratively improve its own output without parameter updates. Reflexion (Shinn et al., 2023) accumulates verbal reflections across agentic episodes as a form of semantic memory. Constitutional AI (Bai et al., 2022) steers generation toward safety via rule-based critiques. More recently, TextGrad (Yuksekgonul et al., 2025) formalizes natural language feedback as “textual gradients,” showing that LLM-generated critiques can propagate through complex systems analogously to backpropagation.

These methods typically use feedback in a cooperative setting, where critiques help a model improve toward a fixed target. RADAR instead uses feedback in an adversarial training loop: detector-side signals are exposed to the generator to guide subsequent rewrites, and the resulting examples are then used to update the detector. This distinction changes the role of feedback from self-improvement toward generating training examples

that reflect failure modes observed by the detector.

3 Methodology

Given a news article x , the detector predicts a binary label $y \in \{0, 1\}$ indicating whether x is real or fake. We focus on adaptive rewrite-style attacks, where a fake article preserves the event frame and journalistic tone of a real source article while altering specific factual claims. RADAR addresses this setting through a round-wise adversarial training framework between a generator G and a detector D , with retrieval optionally enabled on either side.

3.1 RADAR Framework

Figure 2 illustrates the RADAR pipeline. At round t , the generator rewrites a real news article x_i into a fact-altered variant $\hat{x}_{i,t}$, and the detector evaluates whether the article is real or fake. Detector-side diagnostics are then converted into verbal adversarial feedback (VAF), which is stored and used to condition the generator in the next round.

Retrieval can be enabled independently for the generator and the detector, yielding four configurations: G^-/D^- , G^+/D^- , G^-/D^+ , and G^+/D^+ . This decomposition allows us to separately study the effect of generator-side retrieval, detector-side retrieval, and their combination.

Across rounds, RADAR updates the detector using newly generated real–fake pairs, while the generator is refined through prompt-time feedback, cached successful rewrites, and periodic parameter-efficient updates. This round-wise design allows us to examine whether adversarial refinement improves detector robustness over successive rounds.

3.1.1 Generator

The generator G is an instruction-tuned LLM that rewrites a real news article into a deceptive fake variant designed to evade detection. Given a real news article x , retrieval context c , verbal adversarial feedback from the previous round VAF_{t-1} , and cached few-shot exemplars $\mathcal{E}_{t-1}$, the generator produces $\hat{x} = G(x, c, \text{VAF}_{t-1}, \mathcal{E}_{t-1})$.

When generator-side retrieval is enabled, the retrieved passages are used only as realism priors and writing references, rather than as facts to preserve. They provide cues about journalistic style, event framing, and plausible levels of detail, helping the generator produce more natural rewrites.

The generator is instructed to modify the article’s underlying facts while preserving a realistic news style. To induce diverse attack patterns, each

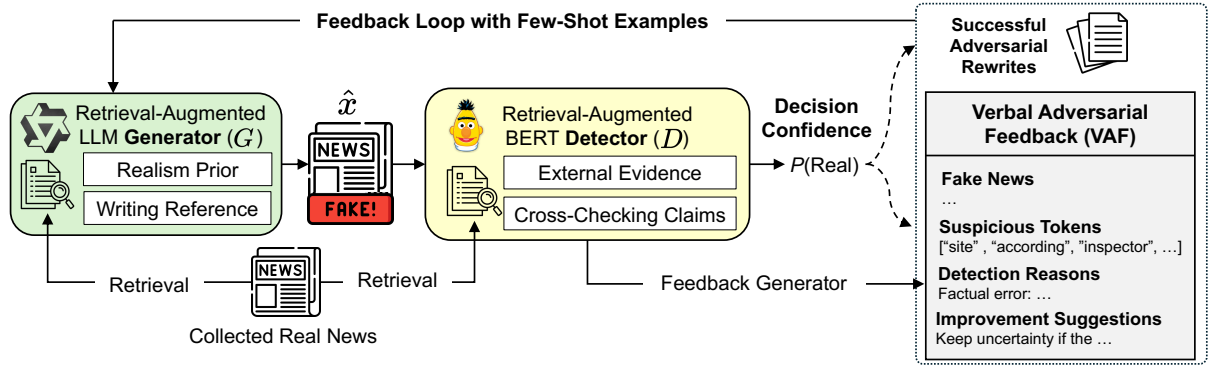


Figure 2: Overview of RADAR for LLM-generated fake news detection under adaptive attacks. Retrieval provides a realism prior for the generator and external evidence for the detector. The detector provides confidence scores and diagnostic signals for VAF, while successful attacks are cached as few-shot examples to guide generator refinement.

generation is conditioned on a manipulation strategy randomly sampled from five options: (i) entity substitution; (ii) numerical or temporal distortion; (iii) outcome reversal; (iv) false causal attribution; or (v) fabricated supporting detail injection.

3.1.2 Detector

The detector D is a trainable encoder-based classifier that predicts whether an input article is real or fake. Given an article x and retrieval context c , the detector outputs a decision confidence $P(\text{real} \mid x, c)$ indicating the likelihood that the article is real. When detector-side retrieval is enabled, the retrieved passages are used as external evidence for cross-checking the claims in the input article.

A scalar decision confidence alone provides limited information about why a generated article was detected or how G should revise it. We therefore augment the detector with VAF, described below.

3.1.3 Verbal Adversarial Feedback

Verbal Adversarial Feedback (VAF) is a structured diagnostic signal composed of three components: *suspicious tokens*, *detection reasons*, and *improvement suggestions*. Instead of using the detector output only as a scalar confidence score, VAF exposes interpretable detector-side diagnostics that can be reused to guide subsequent adversarial rewriting.

VAF is constructed in two steps. First, we extract Gradient-Identified Suspicious Tokens from the detector to identify input spans that strongly influence the prediction. Second, a dedicated Feedback Generator verbalizes these detector-side signals into detection reasons and actionable rewrite suggestions, as illustrated in Table 1.

Gradient-Identified Suspicious Tokens We extract suspicious tokens with integrated gradients

Original Real News

“The cause of the blast was not immediately known, a policeman at the site said, adding that a bomb squad was on its way to investigate.”

Generated Fake News

“The blast was caused by a faulty gas line that had been improperly sealed during a recent renovation, according to a city inspector on-site.”

Suspicious Tokens

{site, according, inspector, blast, renovation}

Detection Reasons

Factual error: the generated article invents a known cause, while the original states that the cause was unknown. Contextual cue: the fake article relies on vague attribution to a city inspector and introduces unsupported causal details.

Improvement Suggestions

Keep uncertainty if the investigation is ongoing; cite a named authority only if the causal claim is verifiable.

Table 1: An example of VAF produced for a generated fake news article. The feedback identifies both the factual perturbation and the contextual cues that make the rewrite detectable.

(IG), which assigns token-level attribution scores by integrating gradients of the detector output with respect to the input embeddings from a baseline to the observed input. After filtering stopwords, we retain the top five tokens with the highest attribution scores as suspicious tokens. These tokens serve as lexical cues for subsequent feedback generation.

Feedback Generator We use an LLM-based generator to instantiate the detection reasons and improvement suggestions in VAF. Its input includes

the source article, the generated fake article, the detection confidence, and the suspicious tokens with their attribution scores. From these signals, it produces structured natural-language feedback that summarizes likely detection errors and proposes targeted revisions for subsequent rewriting. This feedback is then injected into the generator prompt in the next round to support context-aware adversarial refinement.

3.2 Training Strategy

RADAR is trained in rounds with alternating updates of the detector and generator. Each round includes adversarial generation, detector training on real-fake pairs, and generator adaptation using successful attacks. The complete procedure is given in Algorithm 1 in Appendix C.

Detector Training The detector is trained as a supervised binary classifier over real and generated fake articles. In each adversarial round, we continue training the detector from the previous round’s checkpoint on a balanced set of real articles and newly generated fake articles, labeled as $y = 1$ and $y = 0$, respectively. For an input article x_i with detector-side retrieval context c_i , the detector outputs $p_i = P(\text{real} \mid x_i, c_i)$ and is optimized with binary cross-entropy:

$$\mathcal{L}_D = - \sum_i [y_i \log p_i + (1 - y_i) \log(1 - p_i)], \quad (1)$$

where $y_i \in \{0, 1\}$ is the ground-truth label.

Cached successful attacks are used only on the generator side, either as few-shot exemplars for later prompts or as supervised targets for LoRA adaptation. The detector remains a supervised encoder classifier trained on labeled real-fake pairs throughout all rounds.

Generator Training We refine the generator without policy-gradient optimization. Instead, RADAR adapts the generator through three mechanisms: (i) VAF-conditioned prompting for immediate round-to-round refinement, (ii) cached successful adversarial rewrites as few-shot exemplars, and (iii) periodic LoRA-based supervised fine-tuning for persistent adaptation.

At each round, we distinguish between the threshold used to define attack success and the stricter threshold used for generator adaptation. A generated fake article is considered a successful attack if it fools the detector, i.e., $P(\text{real} \mid \hat{x}) > 0.5$.

For few-shot caching and LoRA fine-tuning, we use a stricter threshold of 0.6 to retain only high-confidence successful attacks and avoid adapting the generator to borderline examples near the detector’s decision boundary. We set this threshold a priori as a conservative margin above the binary decision threshold of 0.5. At each round, we retain the top- m high-confidence examples ranked by detector confidence as an adaptation pool, from which one example is randomly sampled for a single LoRA update.

The generator is optimized with

$$\mathcal{L}_G = \mathcal{L}_{\text{CE}} + \lambda_{\text{KL}} D_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}}), \quad (2)$$

where $\mathcal{L}_{\text{CE}}$ is the next-token cross-entropy loss on the sampled successful rewrite, π_θ is the current LoRA-adapted generator, and π_{ref} is the frozen base generator before LoRA adaptation. The KL term regularizes the adapted generator toward the base model to limit distribution drift during repeated adversarial updates.

4 Experiments

In this section, we evaluate RADAR and address four questions: (1) whether RADAR improves fake-news detection over zero-shot LLM detectors and trained baselines, (2) how generator-side and detector-side retrieval contribute to performance, (3) how the proposed feedback design affects robustness, and (4) whether the learned detector generalizes to attacks from an unseen external attacker. We provide additional analyses in Appendix A, including retrieval-overlap analysis and the effect of retrieval strategy and depth.

4.1 Experimental Setup

Dataset and Retrieval Index We conducted our experiments using the *AdvFake-News-Please* dataset released by Chen et al. (2025), which contains real news articles sourced from the News-Please repository along with their corresponding adversarial fakes. To support retrieval augmentation, we use the same public news database as the retrieval corpus, consisting of approximately 811K news articles collected from multiple news sources within the same date range. Following Chen et al. (2025), we filter the retrieval corpus to exclude exact duplicates of the seed news articles and remove articles from the source outlet of the seed articles, reducing the risk of trivial retrieval leakage.

We build a dense FAISS (Douze et al., 2024) index using off-the-shelf DPR checkpoints

(dpr-question_encoder-single-nq-base and dpr-ctx_encoder-single-nq-base for query and passage encoding, respectively) (Karpukhin et al., 2020). During adversarial rounds, generator-side retrieval uses the source real article x_i as the query. Detector-side retrieval instead uses the article currently being classified: $\hat{x}_{i,t}$ for generated fake articles and x_i for real articles. In both cases, we retrieve the top-3 most semantically similar passages, following our retrieval-depth analysis in [subsection A.2](#), where DPR with $k = 3$ consistently outperforms $k = 1$ and $k = 5$ across all three generator backbones.

Training Protocol Following Algorithm 1, the generator uses a fixed set of 500 real seed articles across all 5 rounds, producing 500 fake articles per round. The detector is trained each round on a balanced set of 500 newly sampled real articles and 500 generated fakes, resulting in 1,000 instances per round and 5,000 in total. We adopt the dataset’s standard split and fixed hyperparameters throughout, without additional round-wise validation tuning, to isolate the effect of co-evolution.

Generator Implementation We instantiate the generator G with three compact instruction-tuned LLM backbones: **Qwen3-4B-Instruct** (Yang et al., 2025), **Gemma3-4B** (Gemma Team, 2025), and **Phi-3.5-mini (3.8B)** (Abdin et al., 2024). Each generator is fine-tuned separately under the same adversarial training pipeline. For parameter-efficient and stable adaptation, we use LoRA with rank $r = 16$, scaling factor $\alpha = 32$, and dropout 0.05. A generated fake article is considered a successful attack if $P(\text{real}) > 0.5$, while a stricter threshold of $P(\text{real}) > 0.6$ is used for generator adaptation and exemplar caching. At each round, we rank these high-confidence attacks by detector confidence and retain the top 3 as the adaptation pool. One example is then randomly sampled from this pool for LoRA adaptation. The retained attacks are also added to an exemplar cache of size 5, which keeps only the most recent examples; one cached example is used as a few-shot demonstration in subsequent generation. LoRA adaptation is performed after every round using a single optimizer step with batch size 1. We optimize the generator with AdamW using a learning rate of 1×10^{-4} and apply a KL-divergence penalty ($\lambda = 0.01$) to limit drift from the base model distribution.

For feedback generation, we reuse the same underlying LLM instance but disable the LoRA

adapters during inference, so that feedback is produced using the frozen base-model weights. A separate feedback-generation prompt is used, and no parameters are updated during feedback generation. We set the feedback-generation threshold to $\tau_{fb} = 0.5$: detailed VAF is generated when $P(\text{real}) \leq 0.5$, while samples above this threshold receive score-only feedback.

Detector Implementation The detector D is based on **DeBERTa-v3-base** (He et al., 2021), fine-tuned for binary classification (Real vs. Fake). We use this backbone because it is lightweight and efficiently updatable for evolving attacks, while also reducing the risk of pretraining-time contamination from 2024-era evaluation articles relative to more recent encoders. The detector is trained with AdamW using a learning rate of 5×10^{-6} and continued from the previous round’s checkpoint during adversarial training.

Baselines We compare RADAR against both prompt-only and trainable baselines. For prompt-only baselines, we use the same three instruction-tuned LLM backbones as zero-shot detectors, without any fine-tuning. For trainable baselines, we use the same DeBERTa-v3-base detector backbone as RADAR to ensure a controlled comparison.

We consider two training-based baselines: (1) *Standard Supervised Training*, where a fixed generator produces 2,500 fake articles, matching the total number of fake training examples generated across RADAR’s five adversarial rounds, and the detector is trained once on the resulting static training set; and (2) *Static Adversarial Training*, where a fixed generator produces 500 fake articles per round over five rounds, matching RADAR’s round-wise data volume, and the detector is updated round by round on the corresponding static adversarial data, but without generator adaptation. For all methods, we report ROC-AUC under two detector settings: *w/o Retrieval* (D−) and *w/ Retrieval* (D+), where the latter augments the detector with retrieved evidence from news articles.

4.2 Main Results

Table 2 compares RADAR with prompt-only LLM detectors and trainable DeBERTa baselines under both D− and D+ settings.

Overall, prompt-only LLM detectors show limited performance in this benchmark, particularly without retrieved evidence. Adding retrieval improves these baselines, but the best prompt-only

Baseline / Setting	Model / Configuration	w/o Retrieval (D−)	w/ Retrieval (D+)
<i>Zero-shot LLM Baselines</i>	Qwen3-4B-Instruct	55.73	59.43
	Gemma3-4B	49.40	55.94
	Phi-3.5-mini (3.8B)	42.01	68.39
<i>Trainable Detector vs. Generator: Qwen3-4B-Instruct</i>	Standard Supervised Training	57.41	67.64
	Static Adversarial Training	55.44	71.36
	RADAR	61.65	74.18
<i>Trainable Detector vs. Generator: Gemma3-4B</i>	Standard Supervised Training	58.21	71.29
	Static Adversarial Training	51.13	68.30
	RADAR	59.67	75.13
<i>Trainable Detector vs. Generator: Phi-3.5-mini (3.8B)</i>	Standard Supervised Training	58.59	64.10
	Static Adversarial Training	59.94	69.90
	RADAR	68.43	75.58

Table 2: Main results on the *AdvFake-News-Please* evaluation set (ROC-AUC, %). We compare zero-shot LLM detectors with trained DeBERTa-v3-base baselines under three training settings and the full RADAR framework.

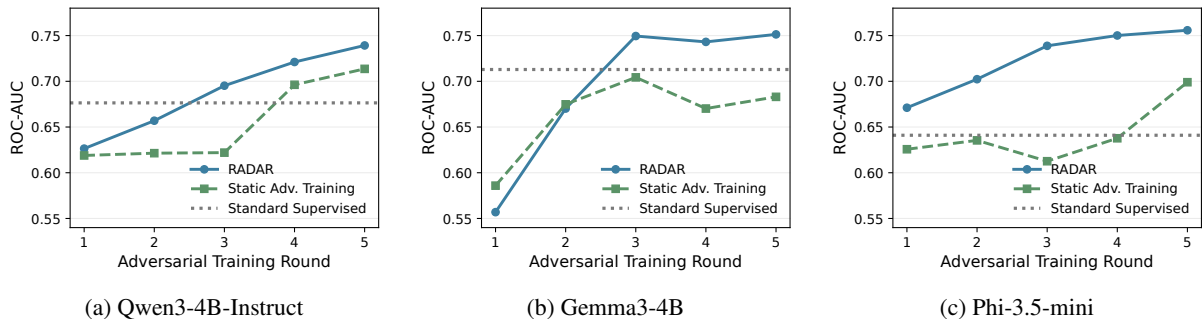


Figure 3: Round-wise detector ROC-AUC across adversarial training rounds. RADAR generally improves over rounds and reaches higher final performance than Static Adversarial Training, while Standard Supervised Training is shown as a fixed reference line.

result still remains below the full RADAR setting.

Across all generator backbones, RADAR achieves the strongest performance among trainable detectors under both D− and D+ settings. Compared with the two trainable baselines, RADAR consistently performs better under matched detector backbones and retrieval settings. This suggests that the improvement is not simply due to using a trainable DeBERTa detector or increasing the amount of generated training data. Rather, the adaptive refinement loop provides more informative adversarial examples for detector training, leading to stronger final performance.

4.3 Effect of Iterative Adversarial Refinement

To examine whether RADAR benefits from iterative adversarial refinement rather than only from additional adversarial data, we track detector ROC-AUC after each training round and compare RADAR with Static Adversarial Training and Standard Supervised Training. As shown in Figure 3, RADAR generally improves over successive

rounds across all three generator backbones, while Static Adversarial Training is less stable and often plateaus at a lower level. This suggests that adapting the generator with feedback and cached successful attacks exposes the detector to progressively more informative training examples, yielding stronger final performance than round-wise detector updates with a fixed attacker.

4.4 Impact of RAG Components

To isolate the contribution of retrieval on each side, we compare four retrieval configurations across three generator backbones in Table 3. Enabling retrieval on either side generally improves performance over the no-retrieval setting, and the strongest results are obtained when both generator-side and detector-side retrieval are enabled.

These results suggest that the two retrieval roles are complementary. Detector-side retrieval provides external evidence for classification, while generator-side retrieval supplies realism priors for constructing fact-altered rewrites. Their combina-

Configuration	Qwen3-4B-Instruct	Gemma3-4B	Phi-3.5-mini
No Retrieval (G− / D−)	59.41	53.65	65.91
Retrieval-Augmented Generator Only (G+ / D−)	61.65	59.67	68.43
Retrieval-Augmented Detector Only (G− / D+)	71.94	65.35	71.99
Retrieval-Augmented Generator & Detector (G+ / D+)	74.18	75.13	75.58

Table 3: Impact of retrieval configuration across generator backbones using the same detector backbone. G denotes generator-side retrieval and D denotes detector-side retrieval. We report detector performance in ROC-AUC (%).

Before rewrite:			
The local commuters inform the Gulf News that these adjustments will affect their budgets and they might need to budget their expenditures more diligently.			
After rewrite:			
Motorists inform Khaleej Times that these modifications will affect their finances and they might need to handle their spending more prudently.			

Before (Top-5 IG)		After (Top-5 IG)	
diligently	3.02	spending	5.67
commuters	2.94	modifications	4.81
expenditures	2.62	prudently	4.67
Gulf	2.50	handle	4.22
local	2.22	finances	4.10

Table 4: Token-saliency visualization using Integrated Gradients. Salient tokens in the original sentence are highlighted in red, and their rewritten counterparts in the generated sentence are highlighted in green.

	Qwen3	Gemma3	Phi-3.5
RADAR	74.18	75.13	75.58
w/o VAF	71.22	73.55	72.61
w/o Few-shot	70.80	72.87	71.56
w/o VAF & Few-shot	66.51	72.49	71.19

Table 5: Ablation of VAF and few-shot feedback components under the dual-retrieval setting (G+ / D+) across different generators. We report detector ROC-AUC (%).

tion yields the most effective adversarial training setup across all three generator backbones.

4.5 Qualitative Analysis of Token Saliency

To illustrate how the detector attends to suspicious spans, we visualize token-level saliency using Integrated Gradients on a rewrite pair. As shown in Table 4, highly salient tokens concentrate on source-related entities and budget-related claim words, which are precisely the anchors replaced by the generator during rewriting.

Model / Configuration	w/o Retrieval	w/ Retrieval
<i>Zero-shot LLM Baselines</i>		
Qwen3-4B-Instruct	46.02	52.22
Gemma3-4B	48.48	52.10
Phi-3.5-mini (3.8B)	41.05	60.43
<i>Trainable Detector vs. Generator: Qwen3-4B-Instruct</i>		
Standard Supervised Training	58.52	58.76
Static Adversarial Training	52.37	63.88
RADAR	60.56	69.20
<i>Trainable Detector vs. Generator: Gemma3-4B</i>		
Standard Supervised Training	57.92	65.94
Static Adversarial Training	52.37	63.15
RADAR	60.28	69.35
<i>Trainable Detector vs. Generator: Phi-3.5-mini (3.8B)</i>		
Standard Supervised Training	58.74	64.90
Static Adversarial Training	54.28	66.52
RADAR	61.21	69.59

Table 6: Cross-attacker transfer results on fake news generated by GPT-5.3 Instant, an unseen external attacker used only at evaluation time. We report ROC-AUC (%) under w/o Retrieval (D−) and w/ Retrieval (D+).

4.6 Effect of Feedback Design

We ablate the two feedback components in RADAR under the dual-retrieval setting (G+ / D+) across three generator backbones. LoRA-based generator adaptation remains enabled in all configurations, so the ablation isolates the effects of VAF and few-shot demonstrations while keeping parameter-efficient generator updates fixed. As shown in Table 5, removing either VAF or few-shot demonstrations consistently hurts performance, while removing both causes the largest drop in every setting.

This suggests that structured detector feedback and successful-attack examples provide complementary signals for generator refinement, with their combination yielding the strongest results across all three generator backbones.

4.7 Cross-Attacker Transferability

To verify that RADAR does not merely overfit to the specific fake-news distribution in the *AdvFakeNews-Please* evaluation set, we construct an ad-

ditional evaluation setting using **GPT-5.3 Instant** (OpenAI, 2026) as an unseen external attacker. We evaluate all methods on fake news generated by this external LLM, which is never used during training, to test transferability to unseen attacks.

Table 6 reports the results. Under this external-attacker setting, zero-shot LLM detectors remain limited, indicating that unseen LLM-generated attacks are challenging without task-specific adaptation. In contrast, across all three generator settings, RADAR consistently outperforms both *Standard Supervised Training* and *Static Adversarial Training* under the same detector backbone. These results suggest that iterative co-evolution improves robustness to attacks from an unseen external generator beyond the original co-evolutionary setting.

5 Conclusion

We proposed RADAR, a retrieval-augmented adversarial training framework for adaptive LLM-generated fake-news detection. RADAR combines a trainable encoder-based detector with an adaptive generator, dual-side retrieval, and Verbal Adversarial Feedback (VAF) to study how retrieval and structured feedback affect detector training under rewrite-style attacks. Experiments on *AdvFake-News-Please* show that RADAR outperforms prompt-only LLM detectors and trainable static baselines, with dual-side retrieval, VAF, and few-shot examples offering complementary benefits. Finally, transfer evaluation with an unseen external attacker shows that RADAR improves cross-attacker transferability. Overall, RADAR demonstrates that evidence grounding and interpretable adversarial refinement are a promising combination for adaptive fake news detection.

Limitations

Although RADAR shows positive cross-attacker transfer to an unseen external generator, our empirical study is still centered on a single benchmark, *AdvFake-News-Please*. This choice is intentional rather than incidental. RADAR is designed to study retrieval-augmented fake-news detection in a leakage-aware setting, where retrieval should provide non-trivial external evidence rather than simply duplicate knowledge already memorized by the detector backbone. Since our detector, DeBERTa-v3-base, is pretrained largely on data ending around early 2021, we prioritize post-2021 news to reduce potential overlap between parametric knowl-

edge and evaluation content. Many widely used fake-news benchmarks, such as FakeNewsNet and BuzzFeed, are dominated by earlier news and may therefore under-test the marginal value of retrieval in our setting. In addition, RADAR requires a dataset that jointly provides text-only real/fake news pairs and a compatible news corpus for evidence retrieval within a similar time range; we found very few public benchmarks satisfying these constraints. As a result, our current evaluation remains limited to a single dataset, and broader cross-dataset validation remains an important direction for future work.

Our evaluation also points to several directions for future work. First, given the computational cost of round-wise generator–detector co-evolution, future studies could examine robustness across multiple random seeds and quantify run-to-run variance. Second, complementary human evaluation could further assess the plausibility, factual alterations, and strategy compliance of generated adversarial examples, providing a more comprehensive characterization of generation quality. Finally, while our lightweight encoder-based detector supports efficient iterative updates, extending RADAR to trainable generative reasoning LLM detectors would help assess its applicability across different detector architectures. Future work could also benchmark inference latency and computational cost to better characterize the accuracy–efficiency trade-off.

Ethical Considerations

Our framework trains a generator to produce realistic, fact-altered rewrites and iteratively improves it with detector-provided verbal adversarial feedback. While this serves the defensive goal of strengthening fake-news detection, it also introduces a clear dual-use risk: the same mechanisms (retrieval-guided realism priors and targeted feedback) could be repurposed to craft more deceptive misinformation that better evades existing detectors.

To mitigate potential misuse, we (i) position RADAR strictly as a *defensive training* pipeline and do not advocate deploying the generator for content creation, (ii) encourage safeguards such as rate limiting, logging, and human-in-the-loop review for any red-team use, and (iii) support complementary ecosystem measures when deploying detection systems. Finally, we emphasize that our experiments are conducted on public benchmarks and aim to improve robustness against adaptive

attacks, not to enable real-world disinformation campaigns.

References

- Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. [Phi-3 technical report: A highly capable language model locally on your phone](#). *Preprint*, arXiv:2404.14219.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. [Constitutional ai: Harmlessness from ai feedback](#). *Preprint*, arXiv:2212.08073.
- Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2020. [Language gans falling short](#). In *International Conference on Learning Representations (ICLR)*.
- Sanxing Chen, Yukun Huang, and Bhuwan Dhingra. 2025. [Real-time factuality assessment from adversarial feedback](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1610–1630. Association for Computational Linguistics.
- Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.
- Gemma Team. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.
- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. [Generative adversarial nets](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680.
- Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. [Long text generation via adversarial training with leaked information](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5141–5148.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [Deberta: Decoding-enhanced bert with disentangled attention](#). In *International Conference on Learning Representations (ICLR)*.
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. [Is bert really robust? a strong baseline for natural language attack on text classification and entailment](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8018–8025.
- Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang. 2021. [Fakebert: Fake news detection in social media with a bert-based deep learning approach](#). *Multimedia Tools Appl.*, 80(8):11765–11788.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems 33*, pages 9459–9474.
- Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. [Adversarial ranking for language generation](#). In *Advances in Neural Information Processing Systems 30*, pages 3155–3165.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Mohammad Vatani Nezafat and Saeed Samet. 2024. [Fake news detection with retrieval augmented generative artificial intelligence](#). In *2nd International Conference on Foundation and Large Language Models, FLLM 2024, Dubai, United Arab Emirates, November 26–29, 2024*, pages 160–167. IEEE.
- Cheng Niu, Yang Guan, Yuanhao Wu, Juno Zhu, Jun-tong Song, Randy Zhong, Kaihua Zhu, Siliang Xu, Shizhe Diao, and Tong Zhang. 2024. [Veract scan: Retrieval-augmented fake news detection with justifiable reasoning](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 266–277. Association for Computational Linguistics.
- OpenAI. 2026. GPT-5.3 Instant: Smoother, more useful everyday conversations. <https://openai.com/index/gpt-5-3-instant/>. Accessed: 2026-03-14.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: Language agents with verbal reinforcement learning](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Sonali Singh and Akbar Siami Namin. 2024. [Adversarial training of retrieval augmented generation to](#)

generate believable fake news. In *IEEE International Conference on Big Data, BigData 2024, Washington, DC, USA, December 15–18, 2024*, pages 3589–3598. IEEE.

Chong Tian, Qirong Ho, and Xiuying Chen. 2025. A symbolic adversarial learning framework for evolving fake news generation and detection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12296–12310, Suzhou, China. Association for Computational Linguistics.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

William Yang Wang. 2017. “Liar, Liar Pants on Fire”: A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426.

Yifeng Wang, Zhouhong Gu, Siwei Zhang, Suhang Zheng, Tao Wang, Tianyu Li, Hongwei Feng, and Yanghua Xiao. 2025. LLM-GAN: constructing generative adversarial network through large language models for explainable fake news detection. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6–11, 2025*, pages 1–5. IEEE.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2852–2858.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. 2025. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639(8055):609–616.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019 (NeurIPS 2019)*, pages 9051–9062.

A Additional Analyses

A.1 Retrieval Overlap Analysis

A potential concern is that the gains from retrieval may partly reflect a trivial “lookup advantage,”

Max ROUGE-L	w/o Retrieval	w/ Retrieval	Δ
[0, 0.25)	60.92	60.88	-0.04
[0.25, 0.5)	60.12	69.52	+9.40
[0.5, 0.75)	52.68	63.15	+10.47
[0.75, 1]	54.59	65.39	+10.80

Table 7: Detector performance stratified by lexical overlap between the input article and retrieved evidence, measured as the maximum ROUGE-L similarity over the top-3 retrieved passages. We report ROC-AUC (%).

where the retriever returns passages that are nearly identical to the input article. To examine this possibility, we measure lexical overlap between each input article and its retrieved evidence using ROUGE-L, and stratify examples by the maximum ROUGE-L score among the top-3 retrieved passages.

Table 7 reports detector performance across overlap bins. In the lowest-overlap regime, $[0, 0.25)$, retrieval yields essentially no improvement, which is expected because evidence in this range is often only weakly related to the input article. By contrast, retrieval provides substantial gains in all three higher-overlap bins, with improvements of +9.40, +10.47, and +10.80 ROC-AUC in $[0.25, 0.5)$, $[0.5, 0.75)$, and $[0.75, 1]$, respectively.

Notably, the gains are similar across these moderate-to-high overlap bins. This suggests that the benefit of retrieval is not driven solely by near-duplicate lookup, but remains effective across a broad range of reasonably related evidence, supporting genuine evidence-based verification rather than trivial lexical matching alone.

A.2 Effect of Retrieval Strategy and Depth

To examine the effect of retrieval configuration, we compare sparse BM25 and dense DPR under three retrieval depths ($k \in \{1, 3, 5\}$) across all generator backbones. Table 8 reports detector ROC-AUC under each setting.

Overall, dense retrieval is consistently stronger than sparse retrieval across generator backbones. DPR achieves the best result for each generator, suggesting that semantically matched evidence is especially useful in our setting, where adversarial rewrites often preserve the overall event frame while altering factual details. In contrast, BM25 is more sensitive to surface lexical overlap and shows less consistent behavior across backbones.

Retrieval depth also has a clear but non-monotonic effect. For DPR, increasing the retrieval depth from $k = 1$ to $k = 3$ consistently improves performance, while further increasing to

Generator	Retriever	$k = 1$	$k = 3$	$k = 5$
Qwen3-4B	BM25	59.84	71.15	69.85
	DPR	66.54	74.18	68.30
Gemma3-4B	BM25	62.60	68.36	69.22
	DPR	65.04	75.13	72.14
Phi-3.5-mini	BM25	62.15	64.01	61.20
	DPR	68.75	75.58	70.52

Table 8: Effect of retrieval strategy and depth across generator backbones. We compare sparse BM25 and dense DPR under different numbers of retrieved passages, and report detector ROC-AUC (%).

$k = 5$ degrades performance across all three generators. This suggests that a moderate number of retrieved passages provides useful evidence coverage, whereas retrieving too many passages may introduce distracting or weakly relevant information. BM25 exhibits a less stable pattern, further supporting the use of dense retrieval for evidence-aware fake news detection. Based on these results, we use DPR with $k = 3$ in all main experiments.

B Prompts

This section provides the prompts actually used in our implementation for reproducibility.

B.1 Generator System Prompt

The generator uses a chat-style prompt with a system message and a user message. At each generation step, the model is instructed to rewrite a real news article into a realistic fake one while applying exactly one randomly selected rewrite strategy. If detector feedback from the previous round is available, the system prompt additionally instructs the generator to satisfy those feedback constraints.

Generator System Prompt

You rewrite real news into realistic fake news. Keep a neutral journalistic tone and similar paragraph structure. Use exactly one rewrite strategy: {strategy_name}. {strategy_instruction} Keep the same central event; do not switch to an unrelated event. No markdown. Output only the rewritten article.
[Appended only if feedback from the previous round is available]
 You must satisfy the detector feedback constraints.

The rewrite strategy is randomly sampled from the following five options:

- *Entity substitution*: Replace a key person, organization, or location with a plausible alternative while keeping the same central event frame.

- *Numerical / temporal distortion*: Alter key numbers or time expressions (dates, counts, durations) to plausible but incorrect values.
- *Outcome reversal*: Invert the main outcome or decision while preserving the original actors and event context.
- *False causal attribution*: Keep the main event but assign its cause to a wrong yet plausible actor, trigger, or mechanism.
- *Fabricated supporting detail injection*: Add plausible but invented supporting details such as a quote, document, statistic, or witness account.

B.2 Generator User Prompt

The user prompt contains the original article, the chosen rewrite strategy, generation constraints, optional detector feedback, and optional retrieval-based writing references.

Generator User Prompt

Original article:
 {article_content}
 Rewrite strategy: {strategy_name}
 Requirements:

- Apply the strategy above as the main manipulation.
- Keep the story internally coherent and plausible.
- Do not copy full sentences from the original.
- Keep style realistic; avoid sensational wording.
- Do not add line breaks or extra sentences.

[Included only if feedback from the previous round is available]
 Detector feedback to address:
 {feedback_prompt}
[Included only if generator-side retrieval is enabled and retrieval context is non-empty]
 Reference style snippets (style boundary only; do not copy facts/names):
 {retrieved_context}
 Return only the rewritten fake news article.

B.3 Feedback Prompt Injected into the Generator

When verbal adversarial feedback (VAF) is enabled, the detector-side analysis from the previous round is converted into a textual feedback block and appended to the generator user prompt. This block is not a fixed hand-written template; rather, it is dynamically assembled from detector scores, suspicious terms, optional LLM-generated critique, and optional few-shot successful examples.

Injected Feedback Prompt

Round {round_id} detector feedback:
 - Scores: {prob_true} real / {prob_fake} fake
[Included only when LLM feedback generation is triggered]

```

- Main issues: {reason_1}; {reason_2}; ...
- Rewrite these terms: {token_1}({score_1}),
{token_2}({score_2}), ...
- Next rewrite focus: {suggestion_1}
{suggestion_2}
[Otherwise, when the fake sample already exceeds
the feedback trigger threshold]
- No LLM rewrite guidance gener-
ated (prob_true={prob_true} > thresh-
old={threshold}).
[Optional; only if few-shot VAF is enabled and the
success cache is non-empty]
Successful sample (imitate tone/structure only, do not
copy facts):
{successful_example_text}
Score of that sample: {successful_example_prob}
real

```

B.4 Feedback Generation Prompt

If the generated fake article receives a sufficiently low plausibility score (i.e., `prob_true` is at or below a predefined threshold), the same underlying LLM instance is reused with its LoRA adapters disabled, so feedback is generated from the frozen base-model weights. This feedback generator uses a separate prompt from the fake-news generator prompt.

Feedback Generator System Prompt

You are a detector feedback generator for fake-news rewriting. Explain why the generated article was likely detected as fake and how to rewrite it to reduce detectability. Return strict JSON only with keys `detection_reasons` and `improvement_suggestions`. Each value must be a list of short actionable strings.

Feedback Generator User Prompt

```

Detector scores:
- prob_true: {prob_true}
- prob_false: {prob_false}
- suspicious_terms: {token_1}({score_1}),
{token_2}({score_2}), ...
Original real article used for rewriting:
{real_news}
Generated fake article:
{fake_news}
Output JSON schema:
{"detection_reasons": ["..."],
"improvement_suggestions": ["...", "..."]}

```

B.5 Detector Input Format

The detector is a classifier that receives the candidate article optionally prefixed with retrieved context. Its actual text input is formatted as follows:

Detector Input Format

```

[Included only if detector-side retrieval is enabled]
Related news stories from search results:
{retrieved_passage_1}
{retrieved_passage_2}
...
Predict the plausibility of the following news story:
{article_to_classify}

```

In our implementation, suspicious words are extracted from the detector input using Integrated Gradients over the classifier’s token embeddings.

B.6 Generator Retrieval Context Format

When generator-side retrieval is enabled, the retrieved passages are not used as factual evidence to preserve, but only as writing-style references that define a realism boundary.

Generator Retrieval Context

```

Real news writing reference (for realism boundary
only; not factual grounding):
{retrieved_passage_1}
{retrieved_passage_2}
{retrieved_passage_3}

```

B.7 External Attacker Prompt for Cross-Attacker Transfer

For the cross-attacker transfer experiment, we construct an additional evaluation set using **GPT-5.3 Instant** as an unseen external attacker. Unlike the RADAR generator prompts above, this prompt is used only at evaluation time for transfer testing.

External Attacker Prompt (GPT-5.3 Instant)

```

You are assisting an academic fact-checking bench-
mark.
Create one synthetic variant of a real news paragraph
by introducing a few believable factual errors. Keep
the writing style neutral and journalistic, and keep a
similar length. This is for offline detector evaluation
and dataset augmentation only.
Output only the rewritten news paragraph. Do not
output explanations, labels, notes, warnings, or any
extra words.
Constraints:
• Keep the event frame and tone realistic.
• Do not add analysis or commentary.
Return only the rewritten news paragraph text. No
preface, no bullet points, and no quotation marks.
Following are rewritten examples:
{few_shot_examples}
Rewrite the following news paragraph for benchmark
data augmentation:
{article_content}

```

C Training Algorithm

Algorithm 1 summarizes the RADAR training procedure. In each round, the generator rewrites fixed real seed articles, while the detector is trained on the generated fakes and separately sampled real articles. Detector-side retrieval uses the article being classified as the query, and detector signals are used to construct adversarial feedback and select successful attacks for generator adaptation.

Algorithm 1 RADAR

Require: Training corpus $\mathcal{D}$, number of rounds T
Require: Retrieval module R , top- k k
Require: Retrieval switches $r_G, r_D \in \{0, 1\}$
Require: Generator G with LoRA adapters, detector D
Require: Feedback procedure F
Require: Exemplar cache size M
Require: Number of selected adaptation examples m
Require: Thresholds $\tau_{\text{adapt}}, \tau_{\text{fb}}$

- 1: Initialize LoRA adapters for G and pretrained detector D
- 2: Sample fixed generator seed set $\mathcal{B}_G = \{x_1, \dots, x_n\}$ from $\mathcal{D}$
- 3: Initialize feedback memory $\text{VAF}_{i,0} \leftarrow \emptyset$ and high-confidence exemplar memory $\mathcal{E}_0 \leftarrow \emptyset$
- 4: **for** $t = 1$ to T **do**
- 5: $\mathcal{S}_{\text{fake}} \leftarrow \emptyset$
- 6: $\mathcal{S}_{\text{adapt}} \leftarrow \emptyset$
- 7: **for each** $x_i \in \mathcal{B}_G$ **do**
- 8: $c_{i,t}^G \leftarrow R(x_i, k)$ if $r_G = 1$, else $\emptyset$
- 9: $e_{i,t} \sim \text{Uniform}(\mathcal{E}_{t-1})$ if $\mathcal{E}_{t-1} \neq \emptyset$, else $e_{i,t} \leftarrow \emptyset$
- 10: $\hat{x}_{i,t} \leftarrow G(x_i, c_{i,t}^G, \text{VAF}_{i,t-1}, e_{i,t})$
- 11: $c_{i,t}^D \leftarrow R(\hat{x}_{i,t}, k)$ if $r_D = 1$, else $\emptyset$ {Fake article retrieves evidence using itself as query}
- 12: $(p_{i,t}^{\text{real}}, p_{i,t}^{\text{fake}}, w_{i,t}^{\text{susp}}) \leftarrow D(\hat{x}_{i,t}, c_{i,t}^D)$
- 13: **if** $p_{i,t}^{\text{real}} \leq \tau_{\text{fb}}$ **then**
- 14: $\text{VAF}_{i,t} \leftarrow F(x_i, \hat{x}_{i,t}, p_{i,t}^{\text{real}}, p_{i,t}^{\text{fake}}, w_{i,t}^{\text{susp}})$
- 15: **else**
- 16: $\text{VAF}_{i,t} \leftarrow \text{score-only feedback}$
- 17: **end if**
- 18: $\mathcal{S}_{\text{fake}} \leftarrow \mathcal{S}_{\text{fake}} \cup \{(\hat{x}_{i,t}, c_{i,t}^D)\}$
- 19: **if** $p_{i,t}^{\text{real}} > \tau_{\text{adapt}}$ **then**
- 20: $\mathcal{S}_{\text{adapt}} \leftarrow \mathcal{S}_{\text{adapt}} \cup \{(\hat{x}_{i,t}, p_{i,t}^{\text{real}})\}$
- 21: **end if**
- 22: **end for**
- 23: $\mathcal{A}_t \leftarrow \text{TopM}(\mathcal{S}_{\text{adapt}}, m)$ {Rank by p^{real} }
- 24: Sample detector real set $\mathcal{B}_{D,t} \subset \mathcal{D}$
- 25: $\mathcal{S}_{\text{real}} \leftarrow \emptyset$
- 26: **for each** $x_j \in \mathcal{B}_{D,t}$ **do**
- 27: $c_{j,t}^{D,\text{real}} \leftarrow R(x_j, k)$ if $r_D = 1$, else $\emptyset$
- 28: $\mathcal{S}_{\text{real}} \leftarrow \mathcal{S}_{\text{real}} \cup \{(x_j, c_{j,t}^{D,\text{real}})\}$
- 29: **end for**
- 30: Train D on balanced real and fake examples:
- 31: $\{(x, c, 1) : (x, c) \in \mathcal{S}_{\text{real}}\} \cup \{(\hat{x}, c, 0) : (\hat{x}, c) \in \mathcal{S}_{\text{fake}}\}$
- 32: **if** $\mathcal{A}_t \neq \emptyset$ **then**
- 33: $a_t \sim \text{Uniform}(\mathcal{A}_t)$ {Sample one example}
- 34: Update G via one LoRA optimization step on a_t , regularized toward the base generator
- 35: **end if**
- 36: $\mathcal{E}_t \leftarrow \text{UpdateCache}(\mathcal{E}_{t-1}, \mathcal{A}_t, M)$ {Retain the M most recent exemplars}
- 37: **end for**
- 38: **return** Trained detector D and adapted generator G
