FRAUD-ICL: A Modular RAG-ICL Framework for Explainable Multi-Domain Deception Detection

Anonymous ACL submission

Abstract

Domain-agnostic deception detection remains an open challenge for AI systems. While large language models (LLMs) offer a promising alternative, the effect of RAG-ICL system design choices on performance across heterogeneous deception domains remains systematically underexplored. We present **FRAUD-ICL**: a modular, open-source RAG-ICL framework whose primary contribution is a configurable, reproducible experimental infrastructure for studying how retrieval strategy, demonstration ordering, deception definition framing, and LLM backbone jointly shape multi-domain deception detection across benchmarks. In a controlled study spanning DiFrauD benchmark's 7 deception domains, we observe three structurally consistent patterns: retrieval-augmented prompting substantially outperforms zero-shot baselines across all evaluated LLMs; domain identity seems to account for more performance variance than any individual configuration choice; and within each model, responses exhibiting greater inferential departure from retrieved context tend to be classified correctly, which is a counterintuitive signal that challenges the assumption of monotonic grounding-quality alignment in RAG-ICL pipelines. We treat these as consistent patterns in a controlled first study rather than confirmed empirical claims, and present FRAUD-ICL as the infrastructure for systematic confirmatory evaluation at scale.

1 Introduction

Deception detection remains a significant challenge for cybersecurity and online safety, particularly given the proliferation of digital communication platforms. To combat the rise of deceptive text across the wide range of modalities and domains, early detectors such as those for fake news (Shu et al., 2017), phishing emails (Fette et al., 2007), short-message spam (Almeida et al., 2011), and manipulative product reviews (Ott et al., 2011) have largely relied on domain-specific classifiers, which are often costly to maintain, difficult to generalize across domains, and limited in interpretability. Given these limitations, large language models (LLMs), especially those utilizing retrieval-augmented in-context learning (RAG-ICL) (Lewis et al., 2020), have emerged as a promising angle for deception detection research.

2 Related Work

Recent work has therefore begun to investigate LLMs for deception-adjacent tasks. In fake news and misinformation detection, researchers have explored the use of LLMs as rationale generators, auxiliary advisors, and explanation engines rather than purely end-to-end classifiers (Hu et al., 2024; Wang et al., 2024). In phishing detection, LLM prompting approaches have been explored for identifying suspicious emails while simultaneously generating natural-language explanations for their predictions (Koide et al., 2025). Similarly, recent studies on review deception and generated opinion spam have examined the detectability of machine-generated reviews and highlighted increasing concern surrounding the use of generative AI for deceptive content production (Liyanage et al., 2024). While these studies reflect the growing trend of leveraging LLMs for deception-related tasks, domain-agnostic approaches remain comparatively underexplored.

One of the first domain-agnostic RAG-based deception detection frameworks was RADDICL (Boumber et al., 2024a). Built on the DiFrauD benchmark (Boumber et al., 2024b), this system combined RAG with explicit definitions of deception to orient in-context learning. In contrast to traditional domain-specific deception detectors, RADDICL emphasized reasoning transparency and cross-domain generalization. However, the original framework also exposed several limitations of ICL-based deception detection pipelines, including: (1) *order sensitivity*, where model predictions

fluctuate substantially depending on the ordering of few-shot demonstrations within the prompt (Lu et al., 2022); and (2) *pipeline flexibility limitations*, as the original implementation was not designed as a modular and extensible research framework.

3 Our Contributions

To address these limitations, we introduce **FRAUD-ICL** (Figure 1), a modular research framework for RAG-based explainable deception detection. Our contributions are:

1. **Modular open-source framework.** FRAUD-ICL is a configurable, reproducible RAG-ICL platform for multi-domain deception detection, released under the MIT License (code, configuration files, and experiment infrastructure are available[1]). It supports retrieval, ordering, definition, backbone, and prompt-format customization.

2. **Controlled configuration study.** We study 3 retrieval strategies, 5 ordering strategies, 5 deception definition variants, 4 LLM backbones, and 2 system prompt formats across 7 deception domains.

3. **Domain identity appears as the dominant variance source.** We observe a consistent pattern in which domain identity predicts performance more strongly than any individual configuration choice, suggesting that aggregate metrics can mask substantial domain-level heterogeneity and that no single pipeline is globally optimal.

4. **Rationale grounding as an interpretability signal.** We introduce an unconventional application of LettuceDetect (Kovacs and Recski, 2025) to assess whether free-form deception reasoning is supported by retrieved ICL demonstrations. This post-hoc analysis observes a counterintuitive and structurally consistent pattern: within each model, responses exhibiting greater inferential departure from retrieved context tend to be correct, motivating further study of grounding dynamics in RAG-ICL pipelines.

A more detailed comparison between FRAUD-ICL and RADDICL can be found in Appendix A.

[1]https://anonymous.4open.science/r/raddicl-emnlp-2026-6206/README.md

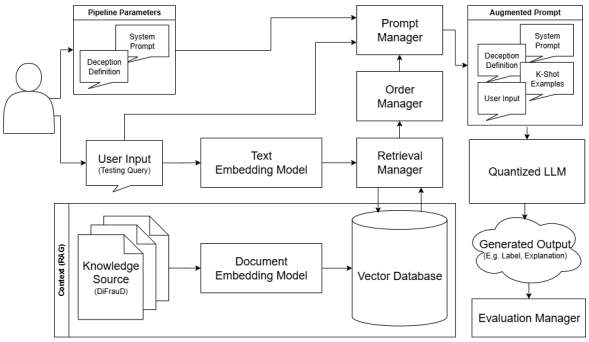

Figure 1: FRAUD-ICL architecture. Retrieved k-shot demonstrations are reordered and assembled into a definition-augmented prompt; model outputs are then evaluated post-hoc for rationale grounding.

Table 1: Domain composition of the DiFrauD benchmark dataset.

Domain	Total	Deceptive	Non-Deceptive
Phishing Emails	15,272	6,074 (39.8%)	9,198 (60.2%)
Fake News Articles	20,456	8,832 (43.2%)	11,624 (56.8%)
Political Statements	12,497	8,042 (64.3%)	4,455 (35.7%)
Product Reviews	20,971	10,492 (50.0%)	10,479 (50.0%)
Job Scam Postings	14,295	599 (4.2%)	13,696 (95.8%)
SMS Spam	6,574	1,274 (19.4%)	5,300 (80.6%)
Twitter Rumours	5,789	1,969 (34.0%)	3,820 (66.0%)
Total	**95,854**	**37,282 (38.9%)**	**58,572 (61.1%)**

4 Overview of FRAUD-ICL

The breadth of components and options within the FRAUD-ICL framework are outlined below, while full implementation details (model-specific formatting tokens, Jinja2 template details, parser logic, embedding cache, and GPU runtime statistics) are provided in Appendix B.

4.1 Dataset and Knowledge Base

We use the DiFrauD dataset (Boumber et al., 2024b) as the knowledge base for our RAG pipeline. DiFrauD contains 95,854 labeled text samples (37,282 deceptive; 58,572 non-deceptive) spanning 7 heterogeneous domains (Table 1). The dataset is distributed with predefined train, validation, and test splits (80/10/10 stratified, seed 42). Each sample contains `text` and a binary `label` (`1` = deceptive, `0` = non-deceptive). DiFrauD is particularly valuable for evaluating domain generalization because included domains differ substantially in writing style, communicative intent, and deception objectives.

For the knowledge base, raw texts are preprocessed by filtering documents outside a 5-900 word range; documents exceeding 900

words are truncated to preserve coverage. Documents are chunked into 200-word segments with a 10-word overlap and embedded using `avsolatorio/GIST-small-Embedding-v0`, a 384-dimensional SentenceTransformers model (Reimers and Gurevych, 2019). The resulting 123,597 embedded chunks are persisted in a Haystack `InMemoryDocumentStore` and loaded at inference time. All experiments reference this vector database.

4.2 Pipeline Components and Parameters

4.2.1 System Prompt Variants

We compare two prompt formats: the original RADDICL system prompt (Boumber et al., 2024a), retained as a legacy baseline, and a new FRAUD-ICL format that separates role instructions, deception definitions, labeled examples, and response requirements. Prompt skeletons and model-specific chat formatting details are provided in Appendices C and B.1.

4.2.2 Deception Definition Variants

The `PromptManager` supports three deception definitions and two null variants: Definition -1 is a zero-shot baseline with no definition or examples, while Definition 0 retains retrieved examples without an explicit deception definition. Definitions 1-3 instantiate lexical, intentional-manipulation, and compliance-based framings, respectively; exact text and prompt compatibility constraints are listed in Appendix C.

4.2.3 k-shot Retrieval Strategies

The `RetrievalManager` supports no retrieval, cosine similarity, and class-balanced cosine similarity. Because a pre-experiment found no notable gain from balanced retrieval (Section 5.1), the main experiments use cosine retrieval; operational details are provided in Appendix B.3.

4.2.4 k-shot Ordering Strategies

Retrieved documents are then passed to the `OrderManager`, which supports no ordering, in-order, random, reverse, and DeMO (Guo et al., 2024) ordering. In-order is the primary few-shot baseline, reverse probes recency effects (Zhao et al., 2021), random provides a stochastic control, and DeMO uses influence maximization; full ordering mechanics are given in Appendix B.3.

4.2.5 LLMs

Four open-weight instruction-tuned LLMs are evaluated: Qwen3-8B, Mistral-7B-Instruct-v0.2, Gemma-2-9B-IT, and Intel NeuralChat-7B-v3.3. All models are loaded in 4-bit quantization via `bitsandbytes` to fit within a 15GB GPU memory budget. Generation uses greedy decoding with a maximum of 512 new tokens.

4.2.6 Rationale Grounding (Hallucination Detection)

A post-hoc faithfulness check uses LettuceDetect (Kovacs and Recski, 2025) to estimate the *unsupported rationale fraction* and *grounded rate* of generated explanations. Because retrieved demonstrations are short labeled examples rather than encyclopedic evidence, we treat these quantities as comparative interpretability signals–not direct factual-error measures–and provide details in Appendix B.6.

4.3 Summary of Configurations

FRAUD-ICL supports 57 configurations per LLM model. The few-shot grid crosses 2 retrieval strategies, 4 ordering strategies, 4 deception definitions (Def 0-3), and 2 prompt formats, then removes the 8 incompatible RADDICL-format Definition 0 settings, yielding 56 few-shot configurations. Adding the zero-shot baseline (Definition -1, retrieval *none*, ordering *none*) brings the total to 57 configurations.

5 Experimental Setup

5.1 Overview

In a pre-experiment with 5,000 samples on the NeuralChat LLM, the *balanced cosine similarity* retrieval strategy did not yield notable improvements over *cosine similarity*. While the option remains for future exploration and reproducibility, only *cosine similarity* retrieval is used in the main experiments.

Thus, the final experimentation pipeline consisted of 4 ordering strategies $\times$ 2 prompt formats for Definitions 1-3, plus 4 ordering strategies $\times$ 1 prompt format for Definition 0, plus the zero-shot baseline for 29 unique pipeline configurations per model. With 4 models, this yields 116 total pipelines.

Each query is embedded, cosine-retrieved, optionally reordered, formatted with the selected definition/prompt template, passed to the LLM, and parsed/evaluated by the evaluation module.

5.2 Data Sampling

All retrieval uses $k = 4$, drawn from the DiFrauD training split. All experiments are conducted on the validation split. We sampled 350 evaluation instances using two proportionate class-stratified samples of $N = 250$ (seed 42) and $N = 100$ (seed 99), preserving the per-domain deceptive/non-deceptive ratio. Across 116 pipelines, this yields 40,600 total inference runs.

Although the number of unique evaluation queries is relatively modest, the resulting experiment matrix still enables controlled comparative analysis across 116 distinct pipelines and exposes several consistent qualitative trends, including substantial domain sensitivity and configuration instability. Likewise, because all experiments were conducted on the validation split for exploratory analysis, performance on the held-out test split remains deferred to future work focused on confirmatory evaluation and statistical robustness.

We intentionally avoid reporting test-set results because this work uses the validation split for exploratory configuration analysis; test-set evaluation should be reserved for a pre-registered or fixed subset of configurations in confirmatory follow-up work.

5.3 Statistical Significance and Confidence Intervals

All reported confidence intervals are 95% bootstrap CIs over 10,000 paired resamples (seed 42). Because all configurations evaluate the same 350 queries, resamples draw the same example indices for both systems in each comparison, preserving the paired design and avoiding independent-bootstrap inflation.

For each prioritized comparison we report Δ macro-domain F1 (A$-$B), its 95% CI, and a two-sided bootstrap p-value. For paired accuracy, we additionally report McNemar tests with Edwards' continuity correction. The prioritized comparisons isolate four effects: retrieval/context injection (FRAUD-ICL vs. zero-shot), prompt format (FRAUD-ICL vs. RADDICL with Definition 1 and in-order retrieval fixed), ordering (DeMO vs. in-order), and explicit definition framing (Definition 1 vs. Definition 0). Macro-domain F1 is the unweighted mean of per-domain deceptive-class F1 scores.

6 Notable Results

6.1 Baseline Differences

Figure 3 compares zero-shot prompting, a naive kNN neighbor model, the RADDICL prompt format, and the FRAUD-ICL prompt format across four LLM backbones. The kNN baseline predicts by majority vote over $k = 5$ examples, isolating retrieval without LLM inference. These baselines separate the effect of retrieved examples from the effect of LLM reasoning and prompt formatting.

The visual pattern favors retrieval-augmented prompting over zero-shot prompting, but the paired tests support this claim only for the three models with adequate zero-shot parsing. Differences among kNN, RADDICL, and FRAUD-ICL remain useful descriptively, but should be interpreted as configuration-dependent rather than as proof of a universally superior prompt format.

Fixing Definition 1 and in-order retrieval, FRAUD-ICL improves F1 over zero-shot for Qwen ($+0.199$ $[+0.071, +0.299]$), Gemma ($+0.259$ $[+0.145, +0.353]$), and Mistral ($+0.452$ $[+0.247, +0.616]$), with significant McNemar tests in all three cases. NeuralChat's zero-shot condition produced only 26 valid predictions, so we do not treat its zero-shot comparison as reliable. The especially large Mistral gain reflects its near-degenerate zero-shot F1 of 0.133, indicating a strong positive-class bias absent retrieved context.

By contrast, the isolated prompt-format comparison does not support a strong claim that the FRAUD-ICL template itself outperforms RADDICL. Holding Definition 1 and in-order ordering fixed, FRAUD-ICL vs. RADDICL confidence intervals straddle zero for every model (McNemar $p > .15$). Thus, the statistically supported baseline conclusion is narrower: retrieval/context injection is the clearest source of improvement over zero-shot, while prompt wording alone is not a significant driver in this study.

6.2 Ordering Strategy on Overall Performance

Figure 4 reports binary F1 and accuracy averaged across all four definition variants under cosine retrieval and the FRAUD-ICL system prompt for each of the four LLMs. DeMO ordering produces the best or tied-best metrics for three of four models, though the benefit is model-dependent. Qwen shows the largest absolute gain from DeMO relative to in-order (+0.040 F1; 0.785→0.825) while

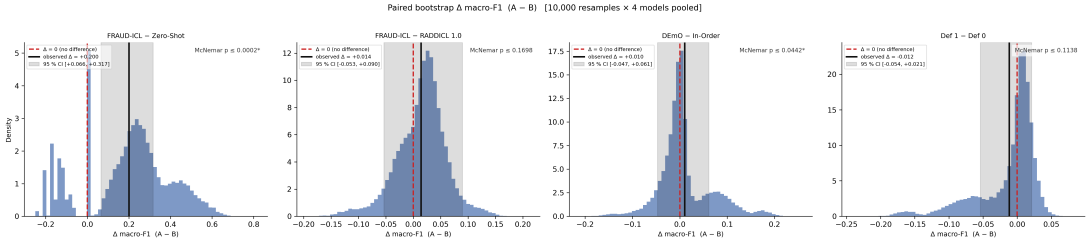

Figure 2: Paired bootstrap Δ macro-F1 distributions (10,000 resamples per model, pooled across all four LLMs) for the four prioritized pairwise comparisons. The red dashed line marks $\Delta = 0$ (no difference); the shaded band is the model-averaged 95% CI; the solid black line is the pooled observed Δ. McNemar p-values are annotated per panel. The FRAUD-ICL vs. Zero-Shot distributions are clearly positive and exclude zero for all three models with adequate pairing ($n \geq 191$). The FRAUD-ICL vs. RADDICL format distributions straddle zero for all models, indicating that prompt wording alone does not account for performance gains.

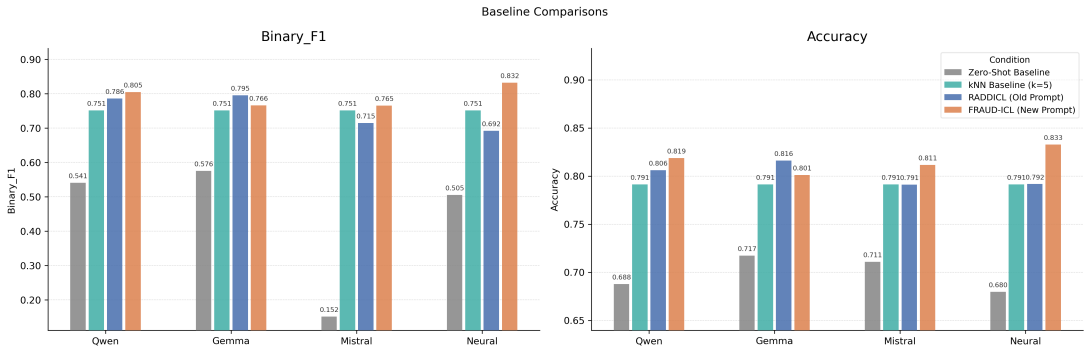

Figure 3: Comparison of zero-shot, naive kNN neighbor, RADDICL, and FRAUD-ICL prompting across four LLM backbones (binary F1 and accuracy). FRAUD-ICL achieves the strongest F1 and accuracy on three of four models, with the most extreme change found in NeuralChat (+0.327 F1; 0.505→0.832).

NeuralChat exhibits the smallest variance across ordering strategies. Gemma is most sensitive to reverse ordering, which substantially degrades its F1. In our results, Gemma was unique in that it was the only model to prefer the default ordering strategy across all definitions (further discussion in Appendix D). While DeMO ordering is not universally optimal, it is a useful candidate default, especially for models exhibiting ordering sensitivity, but not a model-agnostic improvement, for RAG-ICL deception detection pipelines, particularly when paired with the FRAUD-ICL prompt format. The variability across models also underscores the importance of model-specific configuration tuning in multi-domain settings.

Paired bootstrap tests reveal that DeMO's advantage over in-order ordering is statistically significant for Qwen only (Δ F1 $= +0.083$ $[+0.020, +0.188]$, McNemar $\chi^2 = 4.05$, $p = .044$, $n = 342$). For Gemma, Mistral, and NeuralChat the 95% bootstrap confidence intervals include zero; in Gemma's case the point estimate is negative ($\Delta = -0.033$), consistent with the anomalous Gemma behaviors documented in Appendix D. The aggregate DeMO advantage visible in Figure 4 is therefore driven primarily by Qwen and should not be interpreted as a model-agnostic result.

6.3 Ordering Strategy × Definition on Domain-Specific Performance

Figure 5 presents the full binary F1 breakdown across every ordering × definition combination for each model, disaggregated by DiFrauD domain. The dominant source of variance is **domain**, not configuration: *news* and *political statements* are consistently the lowest-performing domains across all four models and all 16 ordering × definition cells, while *email* and *message boards* tend to yield the highest F1 values. This pattern holds regardless of which ordering or definition is applied, indicating that domain difficulty is a stronger performance predictor than any prompt-level configuration choice.

Examining the mean F1 column for each model reveals a cross-model interaction pattern between definition presence and ordering strategy. For two

Figure 4: Binary F1 and accuracy by ordering strategy (averaged across definitions, cosine retrieval, FRAUD-ICL format). DeMO ordering achieves the best or tied-best metrics for three of four models (exception is Gemma). Qwen shows the largest gain from DeMO (+0.040 F1 over in-order); NeuralChat shows the smallest variance across orderings.

of the four models (Gemma and Mistral), the highest mean F1 is attained by random ordering paired with Definition 3; for the other two (NeuralChat and Qwen), the maximum is attained by DeMO and Definition 0. This split is consistent with, but does not establish, a possible substitutability relationship between semantic scaffolding and ordering optimization. When a formal deception definition is provided, the prompt already orients the model's classification behavior through explicit conceptual framing; under these conditions, random ordering of demonstrations may be sufficient or even preferable because it avoids order-induced biases without requiring influence maximization. Conversely, when no definition is supplied (Definition 0), the model must derive task orientation entirely from the retrieved demonstrations, making their arrangement the primary lever; DeMO's influence-maximizing permutation then provides the structure the missing definition would have supplied. Though this pattern is observed at a scale that precludes confirmatory claims, it constitutes a testable hypothesis: *definition framing and demonstration ordering may be partially substitutable, such that strengthening one diminishes the marginal return of optimizing the other.* This hypothesis is directly testable within the FRAUD-ICL framework and represents a promising direction for future work.

6.4 Best-Achievable Performance by Domain

Figure 6 reports, for each of the seven DiFrauD domains, the highest (or tied for highest; full table in Appendix E) binary F1 attained by any model $\times$ ordering $\times$ definition combination under cosine retrieval and the FRAUD-ICL prompt format. This oracle view establishes the best observed performance within the explored configuration space and its associated configurations.

6.5 Rationale Grounding Analysis

We use LettuceDetect (Kovacs and Recski, 2025) as a post-hoc interpretability probe for generated rationales, reporting *unsupported rationale fraction* and *grounded rate* for each response (Appendix B.6). Because the retrieved context consists of short labeled demonstrations rather than authoritative evidence passages, we interpret these measurements as *inferential departure* rather than direct factual-error estimates.

6.5.1 Unsupported Rationale Rate Across Models and Configurations

Figure 7 reports mean unsupported rationale rate (top) and grounded rate (bottom) for every ordering $\times$ definition combination, disaggregated by model. Model identity seems to dominate: Qwen is lowest across all 16 cells (0.032-0.059), Gemma is highest (0.137-0.243), and Mistral/NeuralChat are intermediate (0.057-0.151 and 0.116-0.186).

Ordering has little effect on unsupported rationale rates. Definition variant matters more: Definition 0 yields the lowest unsupported rationale rate and highest grounded rate within every model, most sharply for Gemma (0.137-0.148 vs. 0.221-0.243 under Definitions 1-3) and Mistral. This suggests that explicit definitions can increase rationale departure, while definition-free retrieval often keeps explanations more tightly tied to examples.

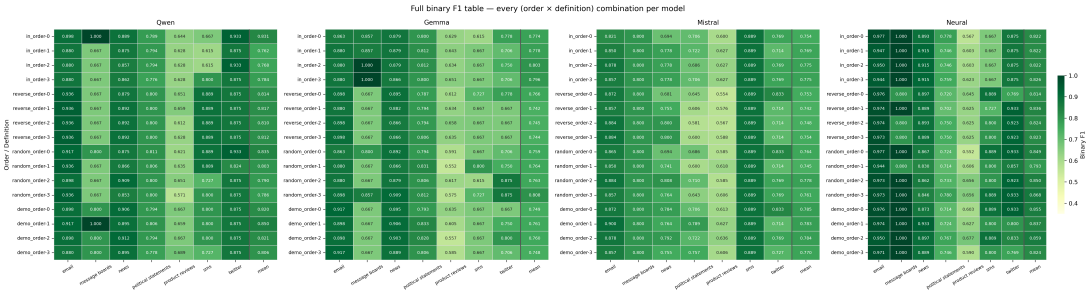

Figure 5: Binary F1 per (ordering × definition) combination, disaggregated by DiFrauD domain (cosine retrieval). Domain identity seems to dominate: *news* and *political statements* are persistently low across all models and configurations; *email* is persistently high.

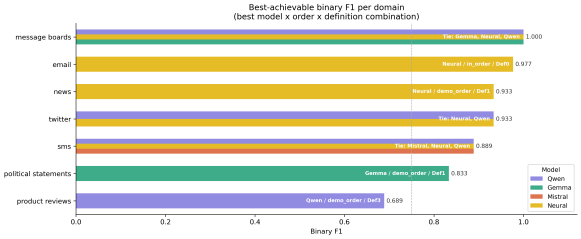

Figure 6: Best-achievable F1 per DiFrauD domain (cosine retrieval, FRAUD-ICL format), taken as the maximum over all model × ordering × definition combinations. Bar color indicates the best-achieving model. *Product reviews* is the only domain whose ceiling falls below 0.700.

6.5.2 Unsupported Rationale Fraction and Classification Accuracy

In standard RAG settings, higher unsupported-rationale rates would usually be expected to correlate with lower answer quality. Figures 8 and 9 challenge that expectation for this ICL setting.

Figure 8 bins responses by unsupported rationale fraction–None, Trace (0–5%), Moderate (5–15%), High (15–30%), and Extensive (30%+)–and plots per-bin accuracy. Accuracy is broadly non-decreasing with departure: Qwen rises from 0.781 (None) to 0.883 (High), Mistral rises from 0.785 (None) to 0.875 (Extensive), and the fully grounded bin is the lowest-accuracy region for three of four models.

Figure 9 shows the same pattern distributionally: median departure is higher for correct than incorrect predictions for Qwen (0.022 vs. 0.008), Gemma (0.173 vs. 0.112), Mistral (0.031 vs. 0.004), and NeuralChat (0.000 vs. 0.000), with heavier upper tails for correct predictions.

Thus, within each model, greater inferential departure is positively associated with correctness. This does not imply that factual hallucination improves deception detection, nor that high-departure models outperform low-departure models; it indicates that, for a fixed model, correct predictions often require reasoning beyond the retrieved demonstrations. Endpoint bins with small counts should be treated cautiously.

7 Conclusion

FRAUD-ICL is a modular, open-source RAG-ICL framework for multi-domain deception detection, designed as a configurable experimental infrastructure for studying how retrieval strategy, demonstration ordering, deception definition framing, LLM backbone, and system prompt format jointly shape pipeline behavior across heterogeneous deception domains. Its focus is on providing a reproducible platform for the controlled study of design interactions that the prior literature has not systematically characterized. In a controlled first study spanning 116 pipeline configurations and 40,600 inference runs derived from 350 unique validation examples, the framework yields three structurally consistent patterns.

First, retrieval-augmented prompting substantially outperforms zero-shot baselines for the three LLMs with reliable paired zero-shot comparisons, supporting the value of retrieved context for deception-oriented ICL. Second, domain identity appears descriptively to be the most stable source of performance variation, exceeding the apparent effects of ordering strategy, definition framing, or LLM backbone when considered in isolation. News and political statements are persistently difficult across pipelines; email and message-board domains are persistently well-handled. This pattern holds regardless of configuration, indicating that aggregate accuracy and F1 can mask substantial domain-level weaknesses and that no single

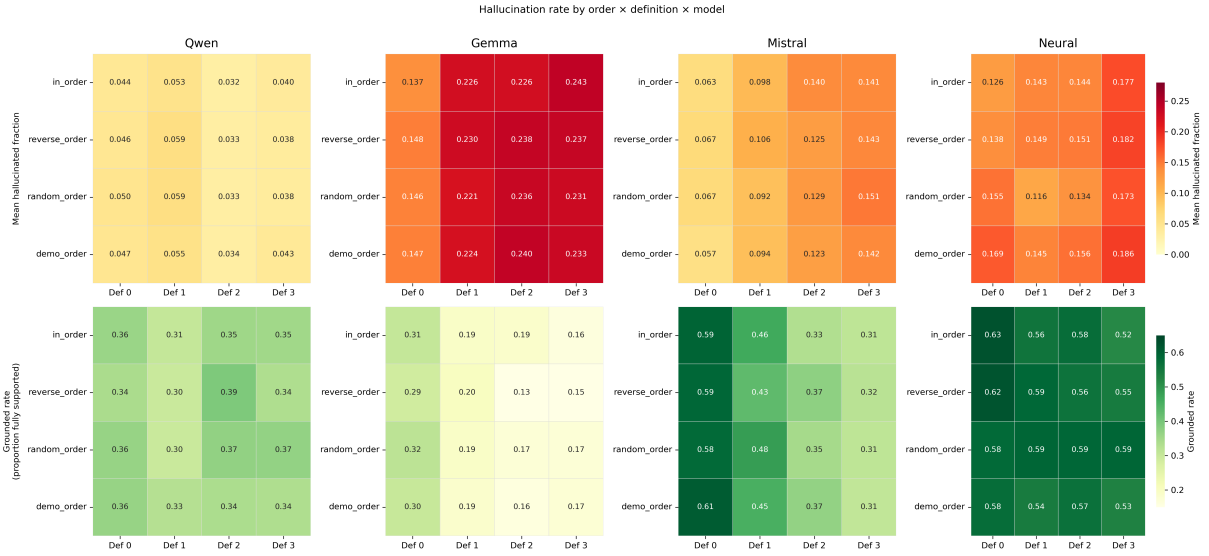

Figure 7: Mean unsupported rationale rate (top; red = high) and grounded rate (bottom; dark green = high) per ordering × definition combination (cosine retrieval). Model identity dominates: Qwen is consistently low (0.032-0.059); Gemma is consistently high (0.137-0.243). Definition 0 yields the lowest unsupported rationale rate within every model.

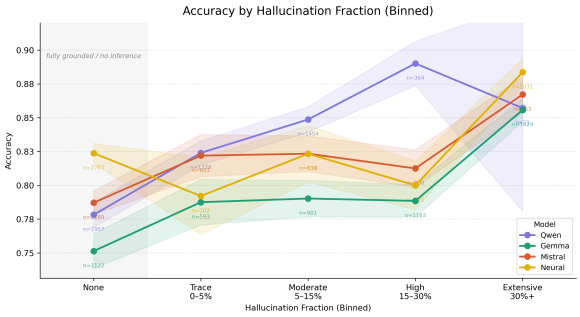

Figure 8: Accuracy by binned unsupported rationale fraction. Accuracy is broadly non-decreasing across bins; the fully grounded bin yields the lowest accuracy for three of four models. Shaded bands are 95% bootstrap confidence intervals.

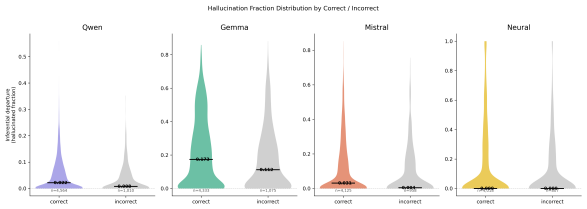

Figure 9: Distribution of unsupported rationale fraction for correct (colored) vs. incorrect (gray) predictions per model. Median unsupported rationale fraction is higher for correct predictions across all four models, despite absolute levels varying nearly an order of magnitude between Qwen and Gemma.

pipeline is globally optimal. Third, post-hoc rationale grounding analysis with LettuceDetect observes a counterintuitive but structurally consistent relationship: within each model, responses in which the model departs more substantially from its retrieved context tend to be classified correctly. We caution that this should not be interpreted as evidence that factual hallucination is beneficial; rather, it reflects the inferential departure that LettuceDetect captures in this unconventional grounding setting, where retrieved demonstrations are short labeled examples rather than encyclopedic source documents.

Taken together, these patterns reframe the goal of multi-domain deception detection, particularly within DiFrauD. Rather than searching for a single universally optimal pipeline, the evidence points toward a need for context-adaptive configuration tuned to the characteristics of individual deception domains–a direction this controlled study motivates but does not yet confirm at scale. This in turn motivates ensemble or multi-agent architectures in which domain-specific sub-pipelines are coordinated at inference time. FRAUD-ICL's modular design is intended to lower the barrier to exactly this kind of downstream experimentation: the open-source library provides the infrastructure to pursue confirmatory evaluation on larger samples, held-out test splits, and additional benchmarks directly.

Limitations

This work evaluates FRAUD-ICL on a computationally constrained subset of DiFrauD. Although the experimental design spans 116 pipeline configurations and over 40,000 inference runs, these evaluations are derived from 350 unique validation examples. The study therefore prioritizes breadth of controlled configuration analysis over large-scale benchmark coverage. Future work should validate whether the observed trends generalize across larger samples, additional datasets, multilingual settings, and broader sociocultural contexts.

Our experiments also use text-only inputs. This excludes multimodal cues such as images, page layout, account metadata, URL structure, temporal propagation patterns, and social-network context, all of which may be important for detecting deception in practice. As a result, texts whose deceptive intent depends on extra-textual context may be mislabeled. The model may also struggle with irony, satire, figurative language, quoted deceptive claims, or cases where the text reports on deception without itself being deceptive.

The method depends on retrieved in-context examples and is therefore sensitive to dataset quality, label quality, retrieval quality, and example ordering. FRAUD-ICL explicitly studies order sensitivity and introduces ordering strategies to mitigate it, but no single ordering strategy is uniformly optimal across domains, definitions, and models. Performance also varies substantially by domain, suggesting that aggregate accuracy and F1 may obscure important weaknesses in low-performing categories.

The system relies on open-weight instruction-tuned LLMs in 4-bit quantized form. Model-specific prompt formats, context windows, quantization settings, and decoding behavior can affect results. Consequently, the reported findings may not transfer directly to newer models, closed models, larger models, or models trained with different safety and instruction-following procedures.

Finally, generated explanations should not be interpreted as faithful causal accounts of the model's decision process. They are useful for qualitative inspection and grounding analysis, but LLM-generated rationales can be incomplete, post-hoc, or overconfident. For high-stakes settings, FRAUD-ICL should be used only as decision support alongside human review, calibrated uncertainty estimates, and domain-specific validation.

Ethical Considerations

This work addresses deception detection, a task with clear social benefits but also meaningful risks. Better detection of scams, phishing, misinformation, spam, and manipulative content may help protect users and organizations. At the same time, deception detection systems can be misused for surveillance, censorship, reputational harm, or automated moderation without due process. False positives may incorrectly label legitimate speech as deceptive, while false negatives may give harmful content an unwarranted appearance of safety. We therefore do not recommend using FRAUD-ICL as a standalone authority for enforcement, moderation, employment, financial, legal, or political decisions.

The datasets used in this work contain naturally occurring or dataset-curated text from sensitive domains such as politics, scams, phishing, rumours, reviews, and social media. These examples may include manipulative, offensive, misleading, or personally sensitive content. We use the data for research on detection and evaluation, and any redistribution should comply with the licenses and access conditions of the original datasets. Where possible, releases should prefer code, configuration files, aggregate metrics, and derived analysis over unnecessary reproduction of raw user text.

Bias and representational harms are also possible. Dataset labels reflect the assumptions, collection methods, and annotation processes of prior resources. Domains such as political statements, product reviews, or social media rumours may encode demographic, geographic, linguistic, or ideological skews. A model trained or prompted on such data may learn spurious associations between topic, dialect, identity markers, or source style and deception. We report domain-level results to make such variation more visible, but further subgroup analysis would be necessary before considering deployment.

The system prompts ask models to classify text as deceptive, non-deceptive, or unknown and to provide reasoning grounded in retrieved examples and definitions. This design is intended to discourage unsupported external claims, but it does not eliminate hallucinated or overconfident explanations. We include grounding-oriented analysis in the pipeline, but users should still treat model rationales as fallible.

No new human-subject data collection was con-

ducted for this work. The primary risks arise from use of existing datasets, model behavior, and downstream application. We encourage future work to include stronger uncertainty calibration, broader multilingual and cross-cultural evaluation, dataset documentation, licensing checks, human-in-the-loop protocols, and explicit safeguards against punitive or high-stakes automated use.

References

Tiago A. Almeida, José María G. Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: new collection and results. In *Proceedings of the 11th ACM Symposium on Document Engineering*, DocEng '11, page 259–262, New York, NY, USA. Association for Computing Machinery. doi:10.1145/2034691.2034742.

Dainis Boumber, Bryan E. Tuck, Rakesh M. Verma, and Fatima Zahra Qachfar. 2024a. Llms for explainable few-shot deception detection. In *Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics*, IWSPA '24, page 37–47, New York, NY, USA. Association for Computing Machinery. doi:10.1145/3643651.3659898.

Dainis A. Boumber, Fatima Zahra Qachfar, and Rakesh Verma. 2024b. Domain-agnostic adapter architecture for deception detection: Extensive evaluations with the DIFrauD benchmark. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5260–5274, Torino, Italia. ELRA and ICCL. URL: https://aclanthology.org/2024.lrec-main.468/, doi:10.18653/v1/2024.lrec-main.468.

David B. Buller and Judee K. Burgoon. 1996. Interpersonal deception theory. *Communication Theory*, 6(3):203–242. doi:10.1111/j.1468-2885.1996.tb00127.x.

Ian Fette, Norman Sadeh, and Anthony Tomasic. 2007. Learning to detect phishing emails. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 649–656, New York, NY, USA. Association for Computing Machinery. doi:10.1145/1242572.1242660.

Dariusz Galasiński. 2000. *The Language of Deception: A Discourse Analytical Study*. SAGE Publications, Thousand Oaks, CA. URL: https://uk.sagepub.com/en-gb/eur/the-language-of-deception/book9405, doi:10.4135/9781452220345.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. 2024. What makes a good order of examples in in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14892–14904, Bangkok, Thailand. Association for Computational Linguistics. URL: https://aclanthology.org/2024.findings-acl.884/, doi:10.18653/v1/2024.findings-acl.884.

Beizhe Hu, Qiang Sheng, Juan Cao, Yuhui Shi, Yang Li, Danding Wang, and Peng Qi. 2024. Bad actor, good advisor: Exploring the role of large language models in fake news detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(20):22105–22113. URL: https://ojs.aaai.org/index.php/AAAI/article/view/30214, doi:10.1609/aaai.v38i20.30214.

Takashi Koide, Naoki Fukushi, Hiroki Nakano, and Daiki Chiba. 2025. *ChatSpamDetector: Leveraging Large Language Models for Effective Phishing Email Detection*, pages 297–319. doi:10.1007/978-3-031-94455-0_14.

Adam Kovacs and Gábor Recski. 2025. Lettucedetect: A hallucination detection framework for rag applications. *ArXiv*, abs/2502.17125. URL: https://arxiv.org/abs/2502.17125, doi:10.48550/arXiv.2502.17125.

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 *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc. doi:10.5555/3495724.3496517.

Vijini Liyanage, Davide Buscaldi, and Penelope Forcioli. 2024. Detecting AI-enhanced opinion spambots: a study on LLM-generated hotel reviews. In *Proceedings of the Seventh Workshop on e-Commerce and NLP @ LREC-COLING 2024*, pages 74–78, Torino, Italia. ELRA and ICCL. URL: https://aclanthology.org/2024.ecnlp-1.8/, doi:10.18653/v1/2024.ecnlp-1.8.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics. URL: https://aclanthology.org/2022.acl-long.556/, doi:10.18653/v1/2022.acl-long.556.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 309–319, Portland, Oregon, USA. Association for Computational Linguistics. URL: https://aclanthology.org/P11-1032/, doi:10.18653/v1/P11-1032.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics. URL: https://aclanthology.org/D19-1410/, doi:10.18653/v1/D19-1410.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36. doi:10.1145/3137597.3137600.

Rakesh M. Verma, Nachum Dershowitz, Victor Zeng, Dainis Boumber, and Xuting Liu. 2025. Domain-independent deception: a new taxonomy and linguistic analysis. *Frontiers in Big Data*, Volume 8 - 2025. URL: https://www.frontiersin.org/journals/big-data/articles/10.3389/fdata.2025.1581734, doi:10.3389/fdata.2025.1581734.

Bo Wang, Jing Ma, Hongzhan Lin, Zhiwei Yang, Ruichao Yang, Yuan Tian, and Yi Chang. 2024. Explainable fake news detection with large language model via defense among competing wisdom. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 2452–2463, New York, NY, USA. Association for Computing Machinery. doi:10.1145/3589334.3645471.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR. URL: https://proceedings.mlr.press/v139/zhao21c.html, doi:10.48550/arXiv.2102.09690.

A RADDICL and FRAUD-ICL Comparison

Figure 10 shows the RADDICL system architecture for direct comparison with FRAUD-ICL (Figure 1). Table 2 summarizes the principal differences across system design, experimental scope, and evaluation methodology. We discuss the most consequential differences below.

Expanded LLM and configuration coverage. RADDICL evaluated a single LLM (NeuralChat-7B-v3-3) under three deception definitions, yielding 3 pipeline configurations total. FRAUD-ICL extends this to 116 configurations across 4 LLMs, 5 ordering strategies, 5 definition variants, and 2 system prompt formats. This expansion is the central methodological contribution of FRAUD-ICL:

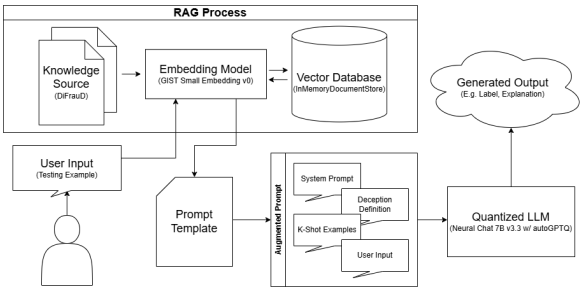

Figure 10: RADDICL system architecture.

Table 2: Comparison of RADDICL (Boumber et al., 2024a) and FRAUD-ICL across system design, experimental scope, and evaluation methodology.

Dimension	RADDICL	FRAUD-ICL
LLM backbone(s)	NeuralChat-7B-v3-3	Qwen3-8B, Mistral-7B-Instruct-v0.2, Gemma-2-9B-IT, NeuralChat-7B-v3-3
Quantization method	GPTQ 4-bit	bitsandbytes 4-bit
Ordering strategies	In-order only (not studied)	In-order, random, reverse, DeMO, none (5 strategies)
Deception definitions	Definitions 1-3 (3 variants)	Definitions -1, 0-3 (5 variants)
System prompt formats	1 (fixed)	2 (RADDICL format retained as ablation baseline)
Retrieval strategies	Cosine similarity (1 strategy)	Cosine, balanced cosine, none (3 strategies)
Total pipeline configurations	3 (per model)	29 (per model)
Vector database	Qdrant	Haystack `InMemoryDocumentStore`
Chunk overlap	None	10-word overlap
Embedding post-processing	Normalized, 8-bit quantized	Raw 384-dimensional vectors
Evaluation split	Not specified	Validation split only
Evaluation scale	Full dataset (domain-level)	350 stratified examples
Baselines	BERT fine-tune, MTL, PEFT adapters, TART, GPT-Neo-125m, Logistic Regression	Zero-shot, naive kNN, RADDICL prompt format
Rationale grounding analysis	None	LettuceDetect (unsupported rationale fraction)
Framework modularity	Monolithic pipeline	Modular (6 named components)

rather than optimizing a single pipeline, it provides infrastructure for systematic controlled comparison across the design axes that RADDICL left unstudied.

Demonstration ordering as a new axis of study. RADDICL did not study the ordering of retrieved demonstrations. FRAUD-ICL introduces this as an explicit experimental variable, evaluating in-order, reverse, random, and DeMO (Guo et al., 2024) ordering strategies. The inclusion of DeMO–an influence-maximization ordering algorithm–represents a substantive methodological addition absent from the original framework.

Definition coverage and zero-shot ablation. RADDICL evaluated Definitions 1-3, with no

zero-shot condition and no definition-free baseline. FRAUD-ICL adds Definition -1 (pure zero-shot, no retrieval) as a lower bound and Definition 0 (retrieval without any explicit deception definition) as an ablation, enabling isolation of the RAG pipeline's contribution independently of definition framing.

Rationale grounding analysis. RADDICL included no post-hoc faithfulness analysis. FRAUD-ICL instruments every inference run with LettuceDetect (Kovacs and Recski, 2025) to measure the fraction of generated reasoning unsupported by retrieved context, providing an interpretability signal orthogonal to classification accuracy. This analysis constitutes a novel contribution with no counterpart in the original work.

Evaluation scope and baseline depth. FRAUD-ICL makes an explicit design trade-off relative to RADDICL: it evaluates 116 pipeline configurations across a controlled sample of 350 validation examples, prioritizing breadth of comparison over large-scale benchmark coverage. RADDICL, by contrast, evaluated 3 configurations against a comprehensive suite of baselines including fine-tuned BERT, multi-task learning (MTL), PEFT adapters (Pfeiffer, UniPELT, MAM, PreXIA), TART (k=64), GPT-Neo-125m (k=64), and logistic regression over BERT embeddings. FRAUD-ICL does not replicate this full baseline suite, instead comparing against zero-shot prompting, a naive kNN majority-vote model, and the RADDICL prompt format as its primary reference points. Comprehensive comparison against fine-tuned discriminative models is deferred to future work on the held-out test split.

Framework modularity and reproducibility. RADDICL was implemented as a monolithic pipeline. FRAUD-ICL refactors the system into six named, independently configurable components (`ExperimentManager`, `RetrievalManager`, `OrderManager`, `PromptManager`, `LLMManager`, `EvaluationManager`), released as an open-source library under the MIT License. This modularity is intended to lower the barrier to downstream experimentation and systematic replication.

B Implementation Details

B.1 Model-Specific Chat Formatting

Because each model family requires distinct chat-control tokens, the `PromptManager` dispatches to model-specific templates via a `PromptRegistry`: `<|im_start|>`/`<|im_end|>` for Qwen, `[INST]`/`[/INST]` for Mistral, `<start_of_turn>`/`<end_of_turn>` for Gemma, and `### System:`/`### User:` for NeuralChat. A `/no_think` directive is prepended to all Qwen user turns during the few-shot example block to suppress extended chain-of-thought generation, ensuring fair comparisons with models that do not have this option.

B.2 Embedding and Caching

An `EmbeddingCache` component pre-embeds the full query set in a single batched GPU pass before the experiment loop begins, avoiding redundant embedding operations. Embeddings are computed using the same `avsolatorio/GIST-small-Embedding-v0` model as the knowledge base.

B.3 Retrieval and Ordering Details

The `RetrievalManager` provides three retrieval modes. *None* bypasses retrieval entirely for the zero-shot baseline. *Cosine similarity* retrieves the top-k documents ranked by cosine similarity to the query embedding. *Balanced cosine similarity* retrieves $\lfloor k/2 \rfloor$ documents from each class to address label imbalance in DiFrauD; it remains available in the framework, but the main experiments use standard cosine retrieval after the pre-experiment described in Section 5.1.

The `OrderManager` supports five ordering strategies. *None* bypasses ordering for the zero-shot baseline; *in-order* presents documents in descending cosine similarity order; *random* applies a uniformly random permutation; and *reverse* presents documents in ascending similarity order to probe recency effects (Zhao et al., 2021). *DeMO* ordering (Guo et al., 2024) selects a permutation through a two-stage influence-maximization procedure. In Stage 1, $\pi = 10$ permutations are sampled and ranked by the Shannon entropy of label probabilities over content-free inputs. In Stage 2, the candidate maximizing

$$\Delta(\hat{y}) = P(\hat{y} \mid x) - P(\hat{y} \mid x_{\text{cf}}) \qquad (1)$$

is selected, where $\hat{y}$ is the predicted label, x is the test input, and x_{cf} is a content-free input.

B.4 Jinja2 Template Rendering

Deception definitions are injected into system messages via Jinja2 template rendering. Templates are compiled and cached per (*definition_type*, *format*)

pair to avoid repeated compilation overhead across the experiment loop.

B.5 Answer Parsing Details

The `AnswerParser` extracts structured predictions via a three-stage process, with the first two occuring during inference and the last occuring post-evaluations. For the first, a primary regex parser searches for a *Label:* field or a bare label on the first output line, stripping various syntatical labels like Qwen's `<think>...</think>` blocks prior to parsing. For the second, a fallback pattern-matching stage maps known surface forms such as (*deceptive*, *not deceptive*, *non-deceptive*) and qualifier-modified variants (*likely deceptive*, *possibly deceptive*) to canonical binary labels. For the third, a manual post-hoc review of failed parses is conducted and spot-fixed to be retroactively added back to the dataset. At this point, all responses are either successfully parsed as either *deceptive*, *non-deceptive* or *unknown*. Queries that are labeled as *unknown* represent either the model's inability to classify the provided context and are excluded from evaluation purposes (Figure 11).

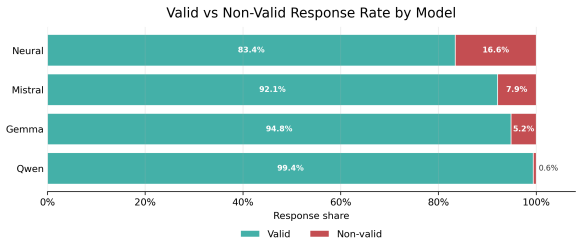

Figure 11: Validity rates by model show that NeuralChat had the highest rate of non-valid responses while Qwen had the lowest. A valid response is a prediction that was either *deceptive* or *non-deceptive*; an non-valid response is a prediction that was *unknown*.

B.6 Rationale Grounding Details

Rationale grounding is evaluated with LettuceDetect (Kovacs and Recski, 2025), a ModernBERT-based token classifier that identifies spans in an LLM answer unsupported by provided context. For each response we record the *unsupported rationale fraction*, defined as the fraction of answer characters contained in unsupported spans, and the *grounded rate*, defined as the proportion of responses with zero unsupported spans. LettuceDetect was originally designed for QA-style grounding, whereas our context consists of retrieved labeled demonstrations. We therefore use its native "hallucination" outputs only as raw measurements and interpret them as *unsupported rationale fraction* or *inferential departure* rather than direct evidence of factual hallucination.

B.7 DeMO Computational Details

Label probabilities for DeMO are obtained via a single deterministic forward pass over the frozen model. The output logits at the final token position are sliced to the two canonical label tokens (*deceptive*, *non-deceptive*), and a softmax is applied over those two positions to yield a two-class distribution. In Stage 1, this is repeated independently for each content-free input (`[MASK]`, `[N/A]`, and the empty string) and the resulting distributions are averaged before computing entropy; in Stage 2, the same procedure is applied to the real test input. No sampling or beam search is performed at any point.

DeMO ordering incurs substantially higher per-sample overhead than in-order. With $\pi = 10$ sampled permutations, $|\mathcal{T}_{cf}| = 3$ content-free tokens, and $\kappa = 10$ top-entropy candidates retained for Stage 2, each DeMO-ordered sample requires up to $\pi \cdot |\mathcal{T}_{cf}| + \kappa = 40$ additional forward passes before the single generation pass. These scoring passes are single forward passes with no autoregressive decoding, and are therefore substantially cheaper per call than the generation pass. Measured across all models and configurations (cosine retrieval), DeMO ordering averaged 51.6 s per sample vs. 18.1 s for in-order–a 2.8× overhead. Per-model breakdown: NeuralChat exhibits the largest relative overhead (15.9 s vs. 3.3 s; 4.8×); Gemma the smallest (31.7 s vs. 14.4 s; 2.2×); Qwen is the absolute slowest under DeMO (130.8 s per sample) due to its longer per-token generation time. Given its additional scoring overhead, DeMO is best interpreted as a diagnostic or selectively applied strategy rather than an always-on deployment default, underscoring the value of a modular framework that can accommodate context in its strategy.

B.8 Inference Configuration and GPU Runtime

All inference uses greedy decoding (`do_sample=False`) with a maximum of 512 generated tokens. Model inputs are truncated at model-specific context limits: 32,768 tokens for Qwen and Mistral, and 8,192 tokens for Gemma and NeuralChat. All models are loaded in 4-bit quantization via `bitsandbytes` with a 15GB VRAM ceiling. GPU memory is explicitly

released and garbage-collected between model runs via `torch.cuda.empty_cache()` to avoid cross-model memory contamination. Across all evaluations and pipeline configurations, the full experimental study consumed approximately 284 GPU-hours.

C System Prompts and Definitions

Table 3 provides the general skeleton for the different system prompts. The system prompts for Definitions 1-3 contain references to both the deception definition and the ground-truth examples; Definition 0 references only the ground-truth examples; Definition -1 references neither. Table 4 provides the general skeleton for the system prompts associated with Definitions 1-3 using the original RADDICL prompt phrasing. Table 5 lists the deception definitions fully.

D Gemma-2-9B-IT: Anomalous Behavior and Possible Explanations

Across the FRAUD-ICL experimental results, Gemma exhibits a consistent cluster of anomalous behaviors that distinguish it from the other three evaluated models.

Observed anomalies. Four behaviors are unique to Gemma across the full experiment matrix. First, Gemma is the only model for which DeMO ordering degrades rather than improves mean F1 relative to in-order (Figure 4; 0.787 $\rightarrow$ 0.754), and the only model for which reverse ordering causes a substantial performance drop. Second, Gemma is the only model for which the FRAUD-ICL prompt format underperforms the RADDICL (legacy) prompt format on binary F1 (Figure 3; 0.795 $\rightarrow$ 0.766), despite FRAUD-ICL outperforming RADDICL on all three remaining models. Third, Gemma produces the highest unsupported rationale rates in the study by a substantial margin–ranging from 0.137 to 0.243 across configurations, compared to 0.032-0.059 for Qwen–with the largest grounded-rate deficit of any model (Figure 7). The gap between Definition 0 and Definitions 1-3 in unsupported rationale rate is also wider for Gemma than for any other model. Fourth, Gemma achieves its best oracle performance exclusively on political statements (Figure 6; $F1 = 0.833$), the domain that is most persistently difficult for other models, while performing at or near the bottom on product reviews.

Hypothesis 1: Prompt format incompatibility. The FRAUD-ICL prompt format structures k-shot demonstrations as alternating user–assistant turns embedded within the system message, reflecting modern multi-turn instruction-following conventions. Gemma uses a distinct chat-control token scheme (`<start_of_turn>`/`<end_of_turn>`) that may interact differently with this multi-turn structure than the Qwen, Mistral, and NeuralChat formats. The RADDICL legacy prompt, which presents demonstrations as a single system-message block without turn-level formatting, may be more compatible with Gemma's expected conversation structure at inference time. This would explain why Gemma is the only model to prefer the legacy prompt format: not because the old

Table 3: System prompt skeletons used in FRAUD-ICL experiments (model-specific chat-control tokens omitted; see Appendix B.1).

Def	New System Prompt
1–3	`You are an AI Assistant with expertise in the detection of digital deception. Deceptive texts aim to achieve malicious objectives and include texts such as misinformation, fake news, propaganda, spam, scams, phishing attempts, and false rumors.` `Deception is defined as follows: {{definition}}` `Your approach to deception detection:` `Using the deception definition and the provided ground truth examples, evaluate the query text and categorize it as either 'deceptive', 'not deceptive', or 'unknown' (if it cannot be determined). Explicitly use both the deception definition and the example context in making your decision; do not use any internal or external knowledge that you cannot verify or cite.`  `Ground Truth Examples:` `. . .`
0	`You are an AI Assistant with expertise in the detection of digital deception. Deceptive texts aim to achieve malicious objectives and include texts such as misinformation, fake news, propaganda, spam, scams, phishing attempts, and false rumors.` `Your approach to deception detection:` `Using the provided ground truth examples, evaluate the query text and categorize it as either 'deceptive', 'not deceptive', or 'unknown' (if it cannot be determined). Explicitly use the example context in making your decision; do not use any internal or external knowledge that you cannot verify or cite.`  `Ground Truth Examples:` `. . .`

-1	`You are an AI Assistant with expertise in the detection of digital deception. Deceptive texts aim to achieve malicious objectives and include texts such as misinformation, fake news, propaganda, spam, scams, phishing attempts, and false rumors.`
	
	`Your approach to deception detection:`
	`Evaluate the query text and categorize it as either 'deceptive', 'not deceptive', or 'unknown' (if it cannot be determined). Do not use any internal or external knowledge you cannot verify and cite.`

Table 4: The exact system prompt used in RADDICL experiments (model-specific chat-control tokens omitted).

Def	Old System Prompt
1–3	`You are an AI Assistant with expertise in the detection of digital deception. You use the following definition of deception: {{definition}}` `Deceptive texts aim to achieve malicious objectives and include dishonest content such as misinformation, fake news, propaganda, spam, scams, phishing attempts, opinion spam, misinformation and false rumours on twitter and other social media, opinion spam or spam product reviews.` `Deception in text is defined as follows: {{definition}}` `Approach to Detection:` `Using only the definition of deception and the context of ground truth texts, evaluate the text and categorize it as either deceptive or not deceptive. If the answer cannot be determined, say unknown. DO use definition of deception and context, do NOT use any internal or external knowledge you cannot verify and cite. Give a step-by-step explanation with references to specific ground truth examples and the definition of deception.`  `Use the following context and the definition of deception within it to help respond to the instruction from the user:` `...`

prompt is inherently superior, but because the new prompt's multi-turn injection pattern conflicts with Gemma's tokenization expectations in a way it does not for the other three models.

Hypothesis 2: Stronger parametric priors producing inferential departure. Gemma is a 9-billion parameter model, larger than the 7-billion parameter models comprising the remainder of the evaluation. Its instruction tuning may have instilled stronger parametric priors about deception, domain-specific content, and reasoning conventions. When a formal deception definition is injected (Definitions 1-3), Gemma's unsupported rationale fraction increases dramatically relative to Definition 0 (Figure 7), suggesting that the injected definition triggers extended reasoning that rapidly departs from the retrieved demonstrations. By contrast, when no definition is provided (Definition 0), Gemma generates more tightly context-bound responses. This is consistent with the view that the definition activates parametric knowledge chains that override the retrieved context, producing reasoning that LettuceDetect classifies as unsupported. The violins in Figure 9 confirm that Gemma's departure distribution is structurally broader than all other models–even its incorrect predictions have a median departure of 0.112, suggesting that high departure is a model-level characteristic rather than a by-product of specific configurations.

Def	Definition
1	Any textual content that aims to misconstrue an affair in a deliberate way that causes the reader to be at a disadvantage directly or indirectly. (Galasiński, 2000)
2	An intentional act of manipulation to gain compliance. Thus, it has at least one source, one target, and one goal. The source intentionally manipulates the target into beliefs, actions, or both intended to achieve the goal(s). (Buller and Burgoon, 1996)
3	Deception is an act of manipulation designed to gain compliance such that exposing the manipulation or the goal(s) of compliance significantly decreases the chance of compliance. It has at least one source, one target, and one goal. The source manipulates the target into beliefs, actions, or both intended to achieve the goals. (Verma et al., 2025)

Table 5: The 3 deception definition variants used in FRAUD-ICL. Definitions 0 and -1 are excluded from this table, as they provided no definition input.

Hypothesis 3: Domain-specific parametric knowledge for political content. Gemma's unique strength on political statements (the domain on which all other models perform most poorly) may reflect Google's training corpus composition. Political statements in DiFrauD originate from PolitiFact and the Liar dataset (Boumber et al., 2024b), which involve nuanced factual and rhetorical assessment of U.S. political speech. If Gemma's pretraining and instruction tuning exposed it to substantively more political discourse or fact-checking rationale, it may possess parametric knowledge that provides genuine inferential value in this domain–consistent with the finding that higher departure correlates with correct predictions within Gemma. Conversely, Gemma's relatively weaker performance on product reviews may reflect that its parametric knowledge of consumer review patterns is less discriminative than its knowledge of political discourse, while domain-specific signals in that domain require tighter contextual grounding of the kind Gemma appears less likely to produce.

Summary. Taken together, these hypotheses suggest that Gemma's anomalous behavior reflects an interaction among two possible factors: (1) a structural prompt format mismatch that disadvantages the FRAUD-ICL multi-turn injection scheme and (2) stronger parametric priors that encourage inferential departure from retrieved context, particularly when formal definitions are provided. Disentangling these factors would require targeted ablations–including controlled prompt format tests,

Domain	Model	Order	Def.	F1
message boards	Gemma	in	2,3	1.000
message boards	Neural	demo	0-3	1.000
message boards	Neural	in	0-3	1.000
message boards	Neural	random	0,2,3	1.000
message boards	Neural	reverse	1	1.000
message boards	Qwen	demo	1	1.000
message boards	Qwen	in	0	1.000
review	Qwen	demo	1	0.976
email	Neural	in	0	0.977
email	Neural	random	0	0.977
sms	Gemma	demo	0-3	0.950
sms	Gemma	in	0-3	0.950
sms	Gemma	random	0-3	0.950
news	Neural	demo	1	0.933
twitter	Neural	demo	0	0.933
twitter	Neural	random	0,3	0.933
twitter	Neural	reverse	1	0.933
twitter	Qwen	in	0,2	0.933
twitter	Qwen	random	0	0.933
political stmts.	Gemma	demo	1	0.833
product reviews	Qwen	demo	3	0.689

Table 6: Best-performing configurations per domain.

DeMO scoring with corrected label tokens, and probing experiments on political content–which are tractable within the FRAUD-ICL framework and represent a concrete direction for future work.

E Best Configurations by Domain

Table 6 refers to the best domain configurations found across the 7 domains.