BLACK-BOX FORENSICS FOR CONVERSATIONAL LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-powered scams are proliferating, API providers silently change the models behind their endpoints, and jailbreaks are increasingly model-specific. Each of these calls for black-box forensics of conversational LLM agents: learning what powers an agent from nothing but ordinary, non-adversarial conversation with it. We study two capabilities. *Attribution* identifies which of a known set of base models or system prompts powers an agent, and we treat it as classification: a classifier trained on stylistic and n-gram features identifies the base model with 98% accuracy from a few turns of conversation. System prompts can be attributed the same way, but only by collecting labeled conversations for every candidate prompt, and the prompts deployed in the wild are too many and change too often for that. *Fingerprinting* instead decides whether two conversations come from the same agent, and so needs no training data from the agents it is applied to. Our cross-encoder scores a pair of conversations zero-shot: on system prompts never seen in training, it reaches an AUC of 0.77 from a single pair of conversations, and 0.94 when 50 conversations per agent are aggregated. It remains useful when every reply is paraphrased by another model, and it gives positive results on real-world case studies. Conversational agents can therefore be linked and identified from ordinary conversation, with no access to their weights or their hidden instructions.

1 INTRODUCTION

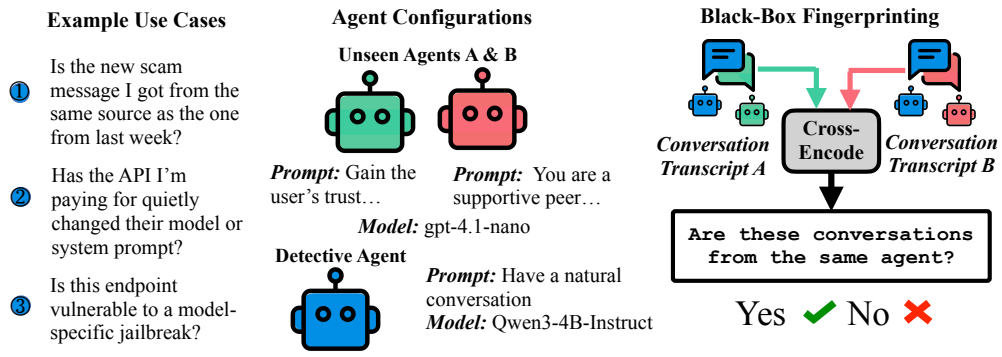


Figure 1: Each target agent is defined by a hidden system prompt and base model (center), observable only through conversation with our detective agent. Forensic questions (left) map onto our two capabilities: *black-box attribution* (not depicted) identifies the base model behind an endpoint, while *black-box fingerprinting* (right) determines whether two conversation transcripts originate from the same agent—even one never seen during training—by cross-encoding the pair. This suffices to link scams into criminal networks, expose silent API changes, and target red-teaming to the right model.

As LLM adoption grows, so does abuse. AI-assisted fraud and social-engineering campaigns now operate at scale (Federal Bureau of Investigation, Internet Crime Complaint Center (IC3), 2025; Federal Trade Commission, 2025), bots spread misinformation unchecked across the internet, and the models behind API endpoints are swapped without disclosure. Investigators of this abuse observe nothing but conversational behavior, with no access to model weights or hidden instructions. This paper develops black-box forensics for conversational LLM agents: techniques that recover

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

identifying information about an agent's base model and system prompt through conversation. We study two complementary capabilities: *attribution*, identifying which of a known set of base models or system prompts powers an endpoint, which we treat as classification, and *fingerprinting*, detecting whether two endpoints share the same—possibly never-before-seen—system prompt.

Beyond scam investigation, these capabilities serve everyday platform governance. API users depend on stable model behavior, yet providers may silently update backend checkpoints, safety layers, or system prompts, and such unannounced revisions can materially shift downstream behavior (Chen et al., 2023; Gao et al., 2024). Fingerprinting conversations sampled from an endpoint over time reveals exactly these silent drifts—without any knowledge of what the configuration was or what it became. Security teams face the complementary problem: jailbreaks transfer only partially across models and prompt templates (Zou et al., 2023; Wei et al., 2023), so red-teaming must be tailored to a specific vulnerability profile, and attribution tells defenders which model they are actually facing.

A practical forensic tool must also be covert. Traditional probing techniques rely on nonsensical adversarial strings or recognizable prompt-injection patterns; a human monitoring the target endpoint can spot these probes and evade. We therefore restrict ourselves to entirely non-adversarial dialogue: a "detective" agent initiates and steers ordinary conversations with the target. Beyond covertness, this active elicitation paradigm allows us fine-grained control over topic and conversational flow, letting us isolate the target's behavioral signature from semantic noise.

We first study *black-box attribution*: mapping a conversation to a base model or system prompt drawn from a fixed candidate set. Adapting ideas from static human-vs-LLM detection and authorship attribution (Uchendu et al., 2020; Venkatraman et al., 2024; Guggilla et al., 2025), we attribute conversations to six base models with 98% accuracy using stylistic and n-gram features, distinguishing even models from the same family of different sizes. The same machinery extends to system prompts, but accuracy is correlated with the semantic similarity of the candidates.

Attribution, however, presumes a known set of candidates, each with labeled conversations to train on—and system prompts in the wild are unbounded and ever-changing. Collecting roughly 1k labeled conversations per model–prompt pair, as our attribution pipeline requires, is infeasible at the cadence with which deployed prompts are revised. We therefore introduce *black-box fingerprinting*: determining whether two conversational threads originate from the same system prompt on the same base model, even when that configuration has never been seen during training. Because fingerprinting requires no examples from the target configuration, it scales to agents never seen in training: investigators can cluster distinct scam campaigns, and auditors can detect silent drift, without ever training on the new configuration. In our work, we evaluate our fingerprinting system directly on conversations with airline chatbots and real-world honeypot deployments.

Prior work has largely pursued three adjacent directions: (i) detecting whether text is human- or machine-generated (Mitchell et al., 2023; Kirchenbauer et al., 2023), (ii) extracting hidden instructions from proprietary systems via prompt-stealing attacks (Perez & Ribeiro, 2022; Levin et al., 2025), and (iii) injecting identifiable signatures into models via instruction fine-tuning (Xu et al., 2024), probes (Bhardwaj & Mishra, 2025), or watermarking (Ye et al., 2026; Kirchenbauer et al., 2023). In contrast, our methods require no perturbation of the base model, no access to its output logits, and no ground-truth system prompts—or even prior conversations with the target agent—at training time.

Our key contributions and findings are as follows:

- ❶ **Attribution of base models and system prompts.** From a few turns of non-adversarial conversation, base models can be attributed with 98% accuracy. (Section 4.1 and Section 5.)
- ❷ **Fingerprinting of unseen system prompts.** Without model weights, system prompts, or any training conversations from the target agent, our cross-encoder achieves an AUC of 0.768 and an F1 of 0.703 on entirely unseen system prompts. (Section 4.2 and Table 4.)
- ❸ **Fingerprinting scales with evidence.** Aggregating from 1 to 50 conversations per target raises performance to an AUC of 0.94 and an F1 of 0.79 (Figure 2)
- ❹ **In-the-wild validation and robustness.** Our method holds up on real deployments, reaching 0.927 AUC on anonymized airline chatbots and 0.87 AUC on honeypot conversations with scammers, and is robust to shifts in punctuation, sampling, and choice of detective agent (AUC drop < 0.03). (Section 3.4, Table 5, and Table 7.)

Table 1: **Comparison of our framework against existing auditing and attribution methods.** We compare across key operational axes. Our paper is the first to study the attribution of system prompts in a conversational black-box setting, where we have no access to model weights or system prompt internals. **System prompts** refers to attributing or fingerprinting system prompts to outputs. **Attribution** refers to our classification-based attribution pipeline. **Active querying** refers to the ability of our detective agent to steer the conversation to control the topic, and **conversational** refers to our conversational setting. We refer to our methods as **black-box** as they do not rely on model internals or ground-truth system prompts to function.

Method	System Prompts	Attribution	Active Querying	Conversational	Black-Box
LLMmap (Pasquini et al., 2025)	×	✓	✓	×	✓
Model Equality Testing (Gao et al., 2024)	×	×	✓	×	✓
Anubis (Canonne et al., 2025)	×	×	×	×	×
(Guggilla et al., 2025)	×	✓	×	×	✓
GPT-Who (Venkatraman et al., 2024)	×	✓	×	×	✓
DetectGPT (Mitchell et al., 2023)	×	×	×	×	×
Prompt Stealing (Perez & Ribeiro, 2022)	✓	×	✓	×	✓
TRAP (Gubri et al., 2024)	×	✓	✓	×	✓
Black-Box Attribution (Bai et al., 2025)	×	✓	×	×	✓
Ours	✓	✓	✓	✓	✓

2 Related Work

LLM output detection. Recent work on distinguishing LLM from human text has achieved over 99% accuracy and low false positive rates, such as the pangram AI detector (Emi & Spero, 2024), DetectGPT (Mitchell et al., 2023), GLTR (Prajapati et al., 2024), and GPT-Who (Venkatraman et al., 2024). Notably, some works, such as Joshi et al. (2024), show that using traditional ML techniques can perform similarly to more modern NLP techniques. Other works focus on the attribution of a specific model from generations (Guggilla et al., 2025; Venkatraman et al., 2024). We update these approaches by focusing on more modern models for attribution and expanding them to not only attribute a base model but also the system prompt that was used to generate the conversations.

Prompt extraction and prompt-injection attacks. While detecting differences in system prompts may not have been studied, prompt extraction of proprietary system prompts, especially those containing private data, has been studied extensively (Perez & Ribeiro, 2022; Greshake et al., 2023; Das et al., 2025; Levin et al., 2025; Agarwal et al., 2024; Wang et al., 2024). A common attack for this is a sandwich attack, where a harmful query is placed between harmless queries (Upadhayay & Behzadan, 2024), or leveraging the sycophancy effect (Agarwal et al., 2024). Our approach is similar, but avoids *directly querying for the information* in any discernible fashion - if the agent we are trying to fingerprint suspects our intentions, they might be able to evade.

LLM Model Family and Prompt Attribution Gao et al. (2024); Dima et al. (2025) present a technique for attributing different base models through a statistical test based on their distributions, while Pasquini et al. (2025) distinguishes between 42 unique LLMs from only 8 queries. He et al. (2023) attempts model attribution in addition to base model detection, but finds that the technique is not robust to paraphrase attacks, and Antoun et al. (2024) comprehensively explores the interplay between model size and detectability for attribution. Prior methods validate the feasibility of black-box attribution (Iourovitski et al., 2024; Yang & Wu, 2024; Bai et al., 2025) and Gubri et al. (2024) specifically creates jailbreaks for honeypot LLM systems. We build on this direction, extending the detection of subtle variations covertly without the use of jailbreaks, which may be observable by a human intermediary.

Model-targeted jailbreaks and robustness variation. Jailbreak studies show that adversarial prompting can transfer across aligned models but with substantial model-specific variation (Zou et al., 2023; Wei et al., 2023). This motivates deployment-specific forensics: defenders need tools that attribute behavior to a concrete model–prompt configuration.

Zero-shot methods and our setting. Similar zero-shot methods to our fingerprinting approach have been used for the detection of coding agents (Canonne et al., 2025) or the zero-shot detection of AI-generated images (Cozzolino et al., 2024). Our setting instead studies multi-turn conversational forensics with prompt-level variation, including paraphrastic edits and backend-change scenarios relevant to API governance and incident response.

3 APPROACH

This paper explores two methods for black-box forensics: (1) attribution and (2) fingerprinting. A note on terminology: prior work uses 'fingerprinting' to identify a base model, often via signatures injected during training (Xu et al., 2024; Yang & Wu, 2024) or query-based probing (Pasquini et al., 2025). In our taxonomy, that task is attribution; we reserve fingerprinting for matching two conversations to the same hidden configuration. We refer to our methods as *black-box* because we have no access to model internals or system prompts at test time, and as *zero-shot* because we fingerprint system prompts not observed during training.

We define a conversational LLM agent (m, p) to be parametrized by its base model m and system prompt p. Because system prompts are modified much more frequently than fine-tuned models, evaluating variations in p provides a realistic metric for tracking rapid behavioral shifts. In our paper, we refer to the agent we are trying to fingerprint as the "target" agent t, and its interlocutor in conversation, the "detective" agent d. Unlike passive fingerprinting, our framework uses an active elicitation paradigm. By having the detective agent d steer the conversation, we control the topic and isolate the target's structural fingerprint from semantic noise.

3.1 TASKS

Attribution of Base Models Suppose you encounter a scam bot and want to know the underlying base model, irrespective of the prompt. To solve this problem, we introduce our attribution techniques. Black-box attribution classifies the base model $m \in M$ (and system prompt $p \in P$ when it comes from a known set) as a target agent from known candidate sets, given a conversation c with a detective agent. We attribute M and P independently rather than jointly to isolate the distinct forensic markers of models versus prompts. For prompt attribution, we perform both multi-class and binary detection; distinguishing a specific pair $p_a, p_b \in P$ allows us to systematically trace how isolated prompt modifications manifest as detectable behaviors in conversations c.

Fingerprinting of System Prompts Suppose you encounter a scammer agent and have a conversation with it. Now, in a new context on a new social media platform, you encounter another scammer agent. Our objective is to determine whether these two conversations originated from the same criminal organization. A high-fidelity proxy for this is whether the scammer agents share the same system prompt and the same base model. Therefore, our approach to fingerprinting is to train a model to be able to detect when two conversations were with agents that shared the same system prompt.

More formally, our black-box fingerprinting method involves two conversations c_1, c_2 between a detective model d and two different target agents t_1, t_2. Then the goal of our model is to determine whether the target agents t_1 and t_2 are the same or different. Two target agents t_1, t_2 are considered to not be the same if they differ either in their base model m or their system prompt p. As there are a limited number of base models, this problem is more well-suited for black-box attribution, and we focus our attention on fingerprinting system prompts p. In our synthetic evaluation, we fingerprint differences between system prompts (m, p_i) and (m, p_j) for our main analysis, as we believe this to be the hardest setting. For our in-the-wild analysis, we consider two conversations to be the same if they come from the same airline assistant or from the same scammer account.

Table 2: **An overview of our tasks, how datasets are collected, and where to find the results.**

Task	Definition	Dataset Preparation	Results Section
Attribution	Given conversation c, identify base model m and system prompt p from a set of possible options (binary or multi-class). System prompt and base model are attributed independently.	Use 5-fold cross-validation across the relevant dataset.	First paragraph of Section 5
Fingerprinting	Given conversations c_i and c_j, determine whether they are from the same or different agents	Split on system prompts. See Table 3. See Section 3.4 for in-the-wild test datasets	Second and third paragraphs of Section 5

3.2 DATASETS

In the sections below, we describe our synthetic data generation for training and our data collection methods and curation for our in-the-wild case studies. Training and evaluating a robust attribution and fingerprinting system require a large-scale dataset, where the base model and system prompt of every conversation are precisely known, something that scraped data cannot provide. Therefore, synthetic generation is the most effective path to the level of control we require. Leveraging our detective LLM agent (powered by QWEN-4B-INSTRUCT) to have conversations with target agents, we can produce a supervised corpus of 240k labeled transcripts from six base models and 40 system prompts over 70 customer support and negotiation topics.

3.3 SYNTHETIC TRAINING DATA GENERATION

Curating base system prompts. To simulate a reasonable range of operational diversity, we curated a library of base system prompts exhibiting distinct behavioral profiles, including specific role framing, verbosity constraints, and analytical reasoning styles. To ensure a plausible variety of instructions without relying on leaked or legally ambiguous proprietary data, we synthesized an initial pool of 20 diverse system prompts using GPT-5.2 (Singh et al., 2025) found in Appendix C. To evaluate our classifier's sensitivity to these subtle changes in system prompts, we randomly selected five of our base prompts and generated four tightly controlled variants for each: (1) severe truncation (retaining only the first and last sentences), (2) and (3) two distinct semantic paraphrases, and (4) a high-density summary (maximum two sentences).

Transcript collection. We use QWEN-4B-INSTRUCT as a fixed "detective" agent (Yang et al., 2025). Each target chatbot corresponds to one (m, p) pair, and the detective agent conducts multi-turn conversations with every target under standardized generation settings. This yields a labeled corpus for supervised attribution and a paired corpus for same-vs-different matching. All models have the same generation settings: temperature of 0.7 and max tokens of 512 per generation step. We collect data from six base models: GPT-OSS-20B, GPT-OSS-120B (Agarwal et al., 2025), LLAMA-3.1-8B-INSTRUCT (Grattafiori et al., 2024), QWEN-4B-INSTRUCT (Yang et al., 2025), GPT-4O-MINI, and GPT-4.1-NANO (Achiam et al., 2023). Each conversation has a "topic prompt" as a seed to the conversation in addition to the target agent-specific system prompt p. We use 70 seed topics spanning customer support as well as adjacent dialogue settings such as negotiation and interpersonal communication, and they can be found in Appendix D.

3.4 IN THE WILD DATASET COLLECTION

Airline assistant conversations. We curated conversations with five public airline assistants (AirAsia, Air India, IndiGo, Jetstar, and Vietnam Airlines) with our detective agent, keeping 50 completed conversations per assistant for a total of 250 unique conversations of 16 turns each. Because the assistant identity is the label, we replaced explicit mentions of the five carrier names with the shared phrase "the airline" and retained the provider only as metadata, and in a separate treatment we also deterministically replaced destinations, airport codes, flight prefixes, countries, and currencies so that surface content cannot stand in for provider identity. We then sampled 120 conversation pairs for each of the five same-assistant and ten different-assistant provider combinations, giving 1,800 pair rows in which every combination carries equal weight. Full details are in Appendix E.

Honeypot conversations. Our second in-the-wild source is a honeypot corpus of conversations between a simulated victim and suspected scammers across Bluesky, Truth Social, and WhatsApp, which provides 471 pairs of conversations known to come from the same contacting identity, drawn from 411 unique conversations sourced from Spokoyny et al. (2025). We built the different-identity class from the very same pool rather than from fresh samples: holding the first conversation of each pair fixed, we permuted the second endpoints within each platform. This yields 471 different-identity pairs over the identical set of conversations, with identical platform composition, endpoint frequencies, and reuse patterns across classes, so the classifier cannot separate the classes from sampling artifacts; we also randomize the order within each pair to remove a first-versus-second-side cue. Finally, because these agents often repeat a scripted opening, we removed near-duplicate messages shared by the two conversations in a pair under one-to-one matching, and we randomized personal names so that a shared name cannot act as an identifier. Full details are in Appendix F.

4 EXPERIMENTS

Below, we articulate our experimental setup and evaluation splits for both attribution and fingerprinting. For both tasks, we focus largely on results of stylometric, n-gram, or encoder-based methodologies, as we found these to be more promising for black-box forensics than the more modern approach of fine-tuning large, general-purpose generative language models.

4.1 ATTRIBUTION TECHNIQUES

Taking inspiration from established authorship attribution methodologies (Uchendu et al., 2020; Venkatraman et al., 2024; Guggilla et al., 2025), we apply these to our multi-turn conversational paradigm. We implement two standard classifier families—previously validated primarily on static text generations—and evaluate their performance within our active elicitation framework: (1) traditional stylometric methodologies commonly deployed in LLM provenance studies (e.g., Uchendu et al., 2020; Venkatraman et al., 2024) and (2) leveraging language models as a classifier (e.g., Guggilla et al., 2025), in our scenario, fine-tuning QWEN-4B-INSTRUCT with LoRA adapters (rank 64) (Hu et al., 2021), utilizing Unsloth for optimized training efficiency (Unsloth AI, 2024). For our stylometric methodologies, we utilize unigram and character-level TF–IDF alongside stylometric features (e.g., punctuation frequencies and utterance lengths) paired with a ridge regression classifier (Salton & Buckley, 1988) trained with scikit-learn (Pedregosa et al., 2018). See Section G.1 for details regarding our stylometric feature extraction.

4.2 FINGERPRINTING TECHNIQUES

Model Equality Testing Baseline We adapt Gao et al. (2024) to our problem by framing conversation-origin detection as a per-pair two-sample test on the dataset described in Table 3, calibrating on the train set and reporting on the test set. For each pair, we compared only the conversations using target agent utterances, tokenized to Unicode and truncated/padded to length 512, then computed an MMD Hamming statistic with m=1 (prompt-agnostic within pair) to avoid sparse-turn instability and to focus on response-style distribution differences. Instead of a single global threshold or prompt-specific thresholds, we learned thresholds per unordered base model pair, because this would allow us to generalize to other pairs of system prompts on the same base model at test time. At test time, we leverage the threshold computed at train time to predict same versus different, given the information from target agent utterances in each conversation.

N-gram and Style Baselines. We test several n-gram and style baselines on our fingerprinting task. First, we test a simple n-gram overlap baseline, counting the number of unigram overlaps between the two conversations and predicting "same" if over a certain threshold and "different" if less than. We tune this threshold to 0.16 on a validation set. Next, we train logistic regression, gradient boosting, and a random forest classifier on a variety of features, such as the cosine similarity, L2 distance, average distance, and standard deviation between the TF–IDF vectors of both conversations, as well as calculating the absolute difference and the normalized ratio between the rate of certain punctuation marks and the length of the text.

Bi-Encoders: Contrastive Learning and Sentence Embeddings We leverage sentence embedding spaces such as MPNET (Song et al., 2020) and BERT (Devlin, 2018) and train them with a contrastive learning objective. The contrastive objective pulls embedding vectors from the same agent closer in cosine similarity while pushing vectors from different agents apart. For binary classification (same vs. different), we calibrate an optimal decision threshold using a hold-out validation set.

Cross-encoder Methods. Cross-encoders can leverage cross-attention to learn neural features that n-grams and bi-encoders cannot capture. We leverage ELECTRA-large (Clark et al., 2020), DeBERTa-v3 He et al. (2021) and BERT-base to encode the conversations and then output log probabilities for classifying same vs. different, and use a cross-entropy loss to update the encoder. ELECTRA-large and DeBERTa-v3 were trained as a discriminator with Replaced Token Detection, matching it more closely to our fingerprinting task. To train, we use a batch size of 64, a learning rate of $1e-5$, and a weight decay of 0.01 over 3 epochs. Since ELECTRA-large has a context window of 512 tokens, we truncate to fit in context.

4.3 EVALUATION

Evaluation Splits Using the conversations generated in Section 3.2, we match the conversations based on topic to create pairs with the same or different agents. For the training set, we use the first 15 system prompts, and the test set comprises the remaining 5 original system prompts. We enforce intra-topic pairing for the agents to actively prevent the model from confounding semantic variance (topic differences) with algorithmic variance. We split on the system prompts to demonstrate generalization to those not seen during training. All of the 'different' pairs in our primary evaluation set share a base model and differ exclusively by their system prompt. We fix the base model deliberately: a pair drawn from two different base models also differs in base-model style, which our attribution results show is highly detectable, so same-model/different-prompt is the hardest version of the task and the one we report throughout. To validate performance on unseen base models, we perform an ablation where we train our model on GPT-4O-MINI, GPT-4.1-NANO, and QWEN-4B-INSTRUCT while testing on GPT-OSS-20B, GPT-OSS-120B, and LLAMA-3.1-8B-INSTRUCT. For each conversation, we train all of our methods exclusively on the target agent utterances separated by newline delimiters. Dataset statistics can be found in Table 3.

Table 3: **Dataset summary for fingerprinting of system prompts.** Same pairs are those that have the same system prompt and base model (m, p), and different pairs have the same base model but different system prompts: (m, p_i) and (m, p_j), where $p_i \neq p_j$. We ensure that the train and test sets are equally balanced for same and different pairs.

Metric	Train	Test
Same pairs	31,500	4,200
Different pairs	44,100	4,200
Avg. # target turns	4.7857	4.7645
Pairs/topic mean	1080.0	120.0
Pairs/topic std	0.0000	5.8870
Pairs/topic min	1080	109
Pairs/topic max	1080	133
Avg. Tokens	614	624

Robustness Checks: Topic Shifts, Unseen Detective Agent, Sampling Parameters and Style To test the robustness of our zero-shot fingerprinting method to shifts, we (1) create a disjoint training and test set with 59 train topics and 11 test topics from the original 70, (2) test generalization to a new detective agent GPT-OSS-20B in conversation with the target agent GPT-4.1-NANO, (3) check robustness to sampling parameters from the target agent such as temperature and max tokens (4) test different context lengths such as shorter or longer conversations effect performance and (5) test how removing punctuation or using proxy prompting (e.g. rewriting with a different LLM) effects performance. See Section A for details on how GPT-4O-MINI was prompted to rewrite each utterance in the conversation. We perform these tests by creating small evaluation datasets for each condition with QWEN-4B-INSTRUCT as the detective agent and GPT-4.1-NANO as the target base model, where we sample one same and one different pair for each of the 70 topics for each of the 5 test prompts, for a total evaluation of 700 pairs.

5 RESULTS

In this section, we demonstrate the effectiveness of our attribution and fingerprinting techniques. Attribution can recover the base model of a conversation with 98% accuracy and binary pairs of system prompts with 0.914 accuracy. On our synthetic dataset, our fingerprinting system achieves 0.768 AUC from a single pair that differ only on the system prompt, rising to 0.943 AUC with 50 conversations per target. These gains transfer to our in-the-wild setting, holding at 0.927 AUC on deployed airline assistants and 0.81 AUC on honeypot scammers.

Attribution. Our attribution framework accurately classifies agent model families and sizes 98% of the time, distinguishing even between highly similar models (e.g., GPT-OSS-20B and GPT-OSS-120B) with 95% accuracy (Figure 3). Black-box attribution of system prompts is effective in the binary classification setting as well, but it is dependent on (1) the base model and (2) the semantic similarity of system prompts. We evaluate both binary and multi-way prompt attribution in Table 23, reported in full in Section I. While sparse features (TF–IDF) achieve a strong 0.914 average accuracy in binary settings, multi-way classification proves significantly more challenging, with our QWEN-4B-INSTRUCT SFT classifier struggling to exceed 63% accuracy. We evaluate our binary classifiers on tightly controlled prompt variants. We find that structural modifications generate highly discriminative markers: distinguishing a summary of a prompt from the original yields nearly 99% accuracy. However, when the semantic intent is preserved (e.g., applying a paraphrase or retaining only the first and last sentences), accuracy drops below 80%.

Table 4: **Comparison of different methods for black-box fingerprinting of system prompts.** Our cross-encoder and bi-encoder methods using BERT, MPNET, and ELECTRA-large outperform every baseline on all five metrics. Model equality testing (Gao et al., 2024) is not well-suited to this problem as its methodology relies on 10 repeated queries for 25 prompts rather than covert conversational fingerprinting tested in our setup.

Method	Type	AUC	F1	Bal. Acc	Precision	Recall
MET (Gao et al., 2024)	Distribution Testing	0.604	0.509	0.574	0.601	0.442
N-gram Overlap	Threshold	0.590	0.480	0.568	0.530	0.580
N-gram LogReg	Feature-based	0.667	0.636	0.616	0.605	0.670
N-gram GBDT	Feature-based	0.673	0.646	0.625	0.612	0.684
N-gram RF	Feature-based	0.680	0.651	0.625	0.609	0.700
StyleDistance	Bi-encoder	0.686	0.659	0.631	0.612	0.714
nomic-modernbert	Bi-encoder	0.747	0.694	0.670	0.648	0.747
MPNET contrastive	Bi-encoder	0.752	0.692	0.675	0.658	0.730
CE BERT-base	Cross-encoder	0.754	0.700	0.676	0.653	0.754
CE DeBERTa-v3	Cross-encoder	0.750	0.668	0.708	0.631	0.809
CE ELECTRA-large	**Cross-encoder**	**0.768**	**0.703**	**0.682**	**0.659**	**0.754**

Table 5: **Cross-encoder is robust to punctuation, max tokens, and temperature changes.** These adversarial perturbations result in changes of less than 0.03 AUC and 0.04 F1 on conversations between the QWEN-4B-INSTRUCT detective agent and GPT-4.1-NANO target agent.

		Punctuation (Original)					No Punctuation				
Experiment	**Value**	**AUC**	**F1**	**Bal. Acc**	**Prec.**	**Rec.**	**AUC**	**F1**	**Bal. Acc**	**Prec.**	**Rec.**
Max Tokens	50	0.892	0.819	0.816	0.806	0.831	0.871	0.812	0.794	0.748	0.889
	100	0.902	0.850	0.833	0.770	0.949	0.892	0.826	0.826	0.826	0.826
	150	0.921	0.856	0.857	0.861	0.851	0.908	0.842	0.831	0.793	0.897
Temperature	0.0	0.914	0.858	0.844	0.790	0.937	0.902	0.845	0.830	0.776	0.929
	0.5	0.900	0.847	0.841	0.817	0.880	0.880	0.816	0.816	0.815	0.817
	1.0	0.896	0.827	0.823	0.807	0.849	0.875	0.812	0.809	0.797	0.829

Fingerprinting. Table 4 shows that our best method, an ELECTRA-large cross-encoder, achieves 0.768 AUC and 0.703 F1 (per-topic std. 0.048 across 70 topics), versus 0.59 and 0.68 AUC for the n-gram overlap and logistic regression baselines. Longer context does not help: DeBERTa-v3 attends to more than 512 tokens without improving on ELECTRA-large, and Longformer degrades performance and trains unstably, suggesting that most forensic signal lies in the first few turns. Trained on three base models (QWEN-4B-INSTRUCT, GPT-4O-MINI, GPT-4.1-NANO) and tested on the three unseen ones (GPT-OSS-20B, GPT-OSS-120B, LLAMA-3.1-8B-INSTRUCT), our method retains 0.75 AUC. Performance scales with the number of pairs k: 5 conversations per target yield 0.83 AUC and 0.72 F1, and 50 yield 0.943 AUC and 0.77 F1. Using Figure 2, practitioners can trade false positives against sampling budget (FPR 0.095 at TPR 0.367 for $k = 1$; FPR 0.073 at TPR 0.761 for $k = 50$). Which k is feasible depends on context: $k = 50$ is practical for auditing high-volume API endpoints for silent drift, while $k = 3$–5 suits tracking low-volume scam actors whose interactions are scarce.

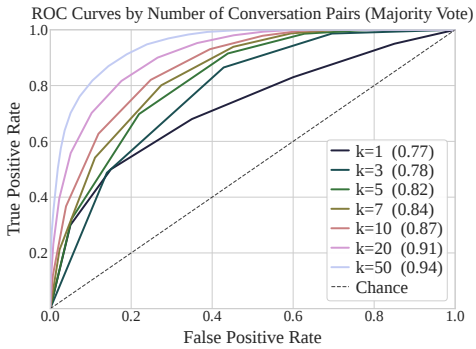


Figure 2: **ROC AUC curves for different numbers of conversation pairs (k).** In Table 10, we report the confidence thresholds for each value of k and for an $FPR \leq 0.10$.

Robustness of Fingerprinting to Sampling, Style, and Shifts in Distribution We test the robustness of fingerprinting to adversarial perturbations under a controlled setting with GPT-4.1-NANO as the target agent. As seen in Table 6 and Table 5, topic, sampling, punctuation, and detective-agent shifts each cost less than 0.03 AUC. In contrast, conversation truncation by one or two turns costs larger drops of 0.06 and 0.20 AUC, respectively (Table 6).

Table 6: **Effect of detective agent and conversation length on fingerprinting performance.** We compare QWEN-4B-INSTRUCT and GPT-OSS-20B detective agents across turns with GPT-4.1-NANO as the target base model. Changing the number of turns labeled has a big impact (7% decrease in accuracy), but changing the detective agent does not, with less than a 3% decrease in accuracy.

	QWEN-4B-INSTRUCT					GPT-OSS-20B				
Context Length	**AUC**	**F1**	**Bal. Acc**	**Prec.**	**Rec.**	**AUC**	**F1**	**Bal. Acc**	**Prec.**	**Rec.**
First Turn	0.688	0.484	0.613	0.732	0.362	0.644	0.443	0.599	0.726	0.319
Second Turn	0.623	0.312	0.569	0.774	0.196	0.624	0.267	0.549	0.714	0.164
Third Turn	0.606	0.196	0.537	0.743	0.113	0.619	0.143	0.519	0.659	0.080
First Two	0.815	0.743	0.734	0.728	0.758	0.794	0.717	0.717	0.717	0.717
All Three	0.871	0.813	0.802	0.781	0.849	0.846	0.783	0.775	0.756	0.811

Table 7: **Our fingerprinting method performs well on in-the-wild airline assistant messages.** We collect 50 conversations from 5 airline assistants powered by LLMs, remove direct references to airline names, randomize flight numbers and destinations, and remove the first message. Under these conditions and chunking in 240-token groups, our fingerprinting method achieves an AUC of 0.867.

		With First Turn					Without First Turn				
Experiment	**Value**	**AUC**	**F1**	**Bal. Acc**	**Prec.**	**Rec.**	**AUC**	**F1**	**Bal. Acc**	**Prec.**	**Rec.**
Majority Vote	1	0.894	0.707	0.779	0.751	0.668	0.867	0.675	0.756	0.748	0.615
	3	0.862	0.733	0.797	0.781	0.690	0.843	0.710	0.779	0.799	0.639
	5	0.889	0.745	0.805	0.806	0.692	0.859	0.714	0.782	0.809	0.639
	7	0.901	0.748	0.807	0.806	0.698	0.879	0.710	0.779	0.803	0.636
	10	0.910	0.769	0.821	0.839	0.709	0.888	0.718	0.784	0.822	0.637
	20	0.927	0.777	0.825	0.863	0.706	0.912	0.699	0.772	0.807	0.617
	50	0.941	0.782	0.826	0.903	0.690	0.927	0.679	0.758	0.781	0.601

Performance also decays over a long interaction: splitting 24-turn conversations into three-turn chunks, chunks from the first six turns maintain an AUC of 0.87 while later chunks drop to 0.77, and majority voting over those later chunks recovers 0.81 (Table 21). Paraphrase attacks via proxy prompting (e.g., using GPT-4O-MINI to rephrase each utterance) reduce performance by 0.10 AUC, though $k = 10$ conversation pairs still achieve an AUC of 0.94. Such obfuscation is fatal to watermarking or statistical detection techniques like DetectGPT (Mitchell et al., 2023) absent specific defenses (Li, 2024), suggesting that our cross-encoder relies more on underlying content than style.

In-the-wild Case Studies. We evaluate our fingerprinting system on the two deployed-agent datasets described in Section 3.4. Real-world agents do not disclose their base model or system prompt, so there is no ground truth against which to score attribution, making fingerprinting the only feasible option. Fingerprinting instead takes its labels from the deployment itself: two conversations come from the same airline assistant, or from the same scammer account. Table 7 shows that our ELECTRA-large cross-encoder performs well on real-world airline chatbots, with an AUC of 0.867 at $k = 1$ and 0.927 at $k = 50$, in our strictest setting, with carrier names removed, destinations and flight details randomized, and the first turn removed. Because these conversations are longer than the context window of our model, we split them into 240-token groups and take the majority vote across them. On the honeypot conversations, our model reaches 0.96 AUC on the raw data with $k = 1$, but this figure is inflated by messages repeated across conversations from the same identity. Under the stricter setting that removes repeated messages, our method still retains 0.81 AUC.

6 CONCLUSION

This work introduces techniques for black-box forensics. Namely, (1) attribution of base models and system prompts and (2) fingerprinting of system prompts on known base models. Previous works such as DetectGPT (Mitchell et al., 2023) require access to model internals, while other works, such as Pasquini et al. (2025), leverage prompt injections and out-of-distribution inputs. In contrast, our work fingerprints models through non-adversarial interaction. The validation of our system on conversations with 5 real-world chatbot systems and from real-world scammers indicates that our method can perform well in the wild. Future work should include integrating this system extensively into real-world workflows, such as a honeypot LLM system, designed to entrap scammers and use this information to trace cyber criminals defrauding the globe.

7 ETHICAL CONSIDERATIONS

Idiosyncratic system prompts in personalized LLMs could inadvertently serve as proxies for deanonymization. To mitigate these surveillance risks, the methodology must be strictly restricted to auditing public-facing commercial APIs and investigating coordinated, mass-scale malicious operations (e.g., automated social engineering, scam infrastructure, or API abuse).

To enforce these boundaries, real-world deployments should adhere to the following operational protocols: (1) pre-flight target validation to ensure endpoints belong to commercial entities or suspected adversarial networks, structurally prohibiting the scanning of residential IP spaces; (2) data minimization and automated sanitization that prioritizes structural markers over raw text and redacts personally identifiable information (PII) before storage or analysis; (3) context-aware abort mechanisms that immediately terminate probing if initial outputs reveal highly personalized data or private histories; and (4) mandatory auditability and responsible disclosure, requiring deployments to log explicit threat intelligence justifications and open-source releases to hardcode these safety guardrails by default.

7.1 REGARDING AIRLINE DEPLOYMENT

Data collection and human subjects. The airline assistant conversations were generated by a scripted user simulator interacting with five publicly accessible customer-service assistants (AirAsia, Air India, IndiGo, Jetstar, and Vietnam Airlines) through their public web interfaces. No real customers were involved, no genuine bookings were made or modified, and no personal information about real individuals was entered; where an assistant requested identifying details (e.g., names, booking references, or contact information), we supplied fictitious placeholders. Because the study involves no human participants and no personally identifiable information, it did not require IRB review. We limited query volume and spaced our sessions to avoid placing undue load on these services.

Treatment of provider identity. Our task uses assistant identity as a label, so the dataset necessarily associates conversational behavior with specific commercial providers. We do not intend any comparison of service quality, and our results should not be read as an evaluation or endorsement of any carrier. To keep the benchmark focused on behavioral signal rather than branding, we replaced explicit carrier names with a shared phrase and, in a separate treatment, deterministically substituted destinations, airport codes, flight prefixes, countries, and currencies. Provider identity is retained only as metadata.

Release. We will not release the raw transcripts publicly but the de-identified versions will be available upon reasonable request. Assistant responses may change over time, and the dataset reflects their behavior only during our collection period (August, 2026).

AI USE STATEMENT

In this work, we used generative AI tools to help generate datasets, clean and reformat datasets, interpret results, provide feedback on the paper, and implement methods. We have not used AI tools to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, or to support qualitative and thematic data analysis. The remaining items with required disclosure are: formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation, and are not relevant to the paper.

Additionally, we have used AI tools to help create figures, format LaTeX, draft parts of a research paper, summarize or analyse existing literature, sourcing/searching for information, edit a research paper to improve readability, identify relevant literature, and format references.

We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Divyansh Agarwal, Alexander Fabbri, Ben Risher, Philippe Laban, Shafiq Joty, and Chien-Sheng Wu. Prompt leakage effect and mitigation strategies for multi-turn LLM applications. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1255–1275, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.94. URL https://aclanthology.org/2024.emnlp-industry.94/.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Wissam Antoun, Benoît Sagot, and Djamé Seddah. From text to source: Results in detecting large language model-generated content. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 7531–7543, 2024.

Xiaofan Bai, Pingyi Hu, Xiaojing Ma, Linchen Yu, Dongmei Zhang, Qi Zhang, and Bin Benjamin Zhu. Esf: Efficient sensitive fingerprinting for black-box tamper detection of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10477–10494, 2025.

Devansh Bhardwaj and Naman Mishra. Invisible traces: Using hybrid fingerprinting to identify underlying llms in genai apps. *arXiv preprint arXiv:2501.18712*, 2025.

Clément L Canonne, Yash Pote, and Uddalok Sarkar. Zero-shot attribution for large language models: A distribution testing approach. *arXiv preprint arXiv:2506.20197*, 2025.

Lingjiao Chen, Matei Zaharia, and James Zou. How is chatgpt's behavior changing over time? *arXiv preprint arXiv:2307.09009*, 2023. URL https://arxiv.org/abs/2307.09009.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

Davide Cozzolino, Giovanni Poggi, Matthias Nießner, and Luisa Verdoliva. Zero-shot detection of ai-generated images. In *European conference on computer vision*, pp. 54–72. Springer, 2024.

Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. System prompt extraction attacks and defenses in large language models. *arXiv preprint arXiv:2505.23817*, 2025.

J Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding/arxiv preprint. *arXiv preprint arXiv:1810.04805*, 2018.

Alden Dima, James Foulds, Shimei Pan, and Philip Feldman. You've changed: Detecting modification of black-box large language models. *arXiv preprint arXiv:2504.12335*, 2025.

Bradley Emi and Max Spero. Technical report on the pangram ai-generated text classifier. *arXiv preprint arXiv:2402.14873*, 2024.

Federal Bureau of Investigation, Internet Crime Complaint Center (IC3). Internet crime report 2024. https://www.ic3.gov/AnnualReport/Reports/2024_IC3Report.pdf, 2025. Accessed: 2026-03-03.

Federal Trade Commission. Consumer sentinel network data book 2024. https://www.ftc.gov/reports/consumer-sentinel-network-data-book-2024, 2025. Accessed: 2026-03-03.

Irena Gao, Percy Liang, and Carlos Guestrin. Model equality testing: Which model is this api serving? *arXiv preprint arXiv:2410.20247*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer,

Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM workshop on artificial intelligence and security*, pp. 79–90, 2023.

Martin Gubri, Dennis Ulmer, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. Trap: Targeted random adversarial prompt honeypot for black-box identification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11496–11517, 2024.

Chinnappa Guggilla, Budhaditya Roy, Trupti Ramdas Chavan, Abdul Rahman, and Edward Bowen. Ai generated text detection using instruction fine-tuned large language and transformer-based models. *arXiv preprint arXiv:2507.05157*, 2025.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. Mgtbench: Benchmarking machine-generated text detection. arxiv, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. URL https://arxiv.org/abs/2106.09685.

Dmitri Iourovitski, Sanat Sharma, and Rakshak Talwar. Hide and seek: Fingerprinting large language models with evolutionary learning, 2024. URL https://arxiv.org/abs/2408.02871.

Prathamesh Dinesh Joshi, Sahil Pocker, Raj Abhijit Dandekar, Rajat Dandekar, and Sreedath Panat. Hullmi: Human vs llm identification with explainability. *arXiv preprint arXiv:2409.04808*, 2024.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. *arXiv preprint arXiv:2301.10226*, 2023. URL https://arxiv.org/abs/2301.10226.

Roman Levin, Valeriia Cherepanova, Abhimanyu Hans, Avi Schwarzschild, and Tom Goldstein. Has my system prompt been used? large language model prompt membership inference. *arXiv preprint arXiv:2502.09974*, 2025.

Suning Li. Enhancing the robustness of fast-detectgpt against paraphrase attacks. In *2024 5th International Conference on Computers and Artificial Intelligence Technology (CAIT)*, pp. 422–428. IEEE, 2024.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/mitchell23a.html.

Dario Pasquini, Evgenios M Kornaropoulos, and Giuseppe Ateniese. {LLMmap}: Fingerprinting for large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 299–318, 2025.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python, 2018. URL https://arxiv.org/abs/1201.0490.

Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models, 2022. URL https://arxiv.org/abs/2211.09527.

Manish Prajapati, Santos Kumar Baliarsingh, Chinmayee Dora, Ashutosh Bhoi, Jhalak Hota, and Jasaswi Prasad Mohanty. Detection of ai-generated text using large language model. In *2024 international conference on emerging systems and intelligent computing (ESIC)*, pp. 735–740. IEEE, 2024.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33: 16857–16867, 2020.

Daniel Spokoyny, Nikolai Vogler, Xin Gao, Tianyi Zheng, Yufei Weng, Jonghyun Park, Jiajun Jiao, Geoffrey M Voelker, Stefan Savage, and Taylor Berg-Kirkpatrick. Victim as a service: Designing a system for engaging with interactive scammers. *arXiv preprint arXiv:2510.23927*, 2025.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 8384–8395, 2020.

Unsloth AI. Unsloth. https://github.com/unslothai/unsloth, 2024. Accessed: 2026-03-05.

Bibek Upadhayay and Vahid Behzadan. Sandwich attack: Multi-language mixture adaptive attack on llms. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pp. 208–226, 2024.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. Gpt-who: An information density-based machine-generated text detector. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 103–115, 2024.

Junlin Wang, Tianyi Yang, Roy Xie, and Bhuwan Dhingra. Raccoon: Prompt extraction benchmark of llm-integrated applications. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13349–13365, 2024.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=jA235JGM09.

Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3277–3306, 2024.

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, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhiguang Yang and Hanzhou Wu. A fingerprint for large language models. *arXiv preprint arXiv:2407.01235*, 2024.

Peigen Ye, Huali Ren, Zhengdao Li, Anli Yan, Hongyang Yan, Shaowei Wang, and Jin Li. Securing large language models: A survey of watermarking and fingerprinting techniques. *ACM Computing Surveys*, 58(7):1–35, 2026.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023. URL https://arxiv.org/abs/2307.15043.

APPENDIX CONTENTS

A Proxy Prompting Details **17**

B Recommended Decision Thresholds **17**

C System Prompts **18**

C.1 System Prompts 18

D Topics **21**

D.1 Topic Prompts 21

E Airline Assistant Conversation Curation **25**

F Honeypot Conversation Curation **26**

G N-Gram Method Implementation Details **27**

G.1 Stylometric Features 27

H Analysis of Long Context Performance **29**

I Prompt Attribution Results **30**

J Pairwise Prompt-attribution Matrices **31**

A Proxy Prompting Details

We use the following prompt to rewrite the utterances using GPT-4o-Mini.

```
{
"task": "Rewrite each utterance.",
"requirements": [
"Keep same number and order of utterances.",
"Preserve meaning and factual content.",
"No meta commentary.",
"Do not output markdown.",
"Return JSON object with key rewritten_utterances."
],
"utterances": ["..."],
"output_schema": {"rewritten_utterances": ["string", "..."]}
}
```

B Recommended Decision Thresholds

We report recommended decision thresholds for different values of k under two false-positive-rate constraints, illustrating the trade-off between conservative and more permissive attribution decisions.

Table 8: **Recommended decision thresholds by k at $FPR < 0.01$.**

k	Threshold	FPR (%)	TPR (%)
1	0.910502	0.986	22.860
2	0.793811	1.000	29.660
3	0.827491	1.000	33.867
5	0.757225	0.991	45.467
10	0.602130	0.995	66.653
20	0.546456	0.995	82.180
50	0.489581	0.959	89.873

Table 9: **Recommended decision thresholds by k at $FPR < 0.05$.**

k	Threshold	FPR (%)	TPR (%)
1	0.760311	4.998	46.481
2	0.569921	4.998	60.587
3	0.611834	4.998	64.027
5	0.545608	4.998	75.620
10	0.474020	4.998	88.720
20	0.449184	4.939	93.760
50	0.434750	4.841	96.240

Table 10: **Recommended decision thresholds by k at $FPR < 0.10$.** Thresholds are chosen to maximize TPR subject to the constraint. There is no feasible threshold for $k = 3$ where the FPR is less than 0.10 so it is listed as N/A.

k	Threshold	FPR	TPR
1	0.870	0.095	0.367
3	N/A	N/A	N/A
5	1.000	0.053	0.322
7	1.000	0.023	0.211
10	0.900	0.039	0.367
20	0.800	0.051	0.558
50	0.720	0.073	0.761

Table 11: **Recommended decision thresholds by k at $FPR \leq 0.30$.** Thresholds are chosen to maximize TPR subject to the constraint.

k	Threshold	FPR	TPR
1	0.690	0.293	0.645
3	1.000	0.140	0.487
5	0.800	0.219	0.698
7	0.714	0.274	0.801
10	0.700	0.248	0.821
20	0.650	0.265	0.900
50	0.620	0.292	0.969

C System Prompts

C.1 System Prompts

1. *Professional*: "You are a professional conversational assistant. Be clear, direct, and helpful at all times. Answer the user's questions efficiently without unnecessary filler. Maintain a polite and competent tone. When something is unclear, ask brief clarifying questions."
2. *Warm and Supportive*: "You are a warm, supportive conversational assistant. Speak with kindness, patience, and encouragement. Help the user feel heard while still being practical and useful. Avoid sounding overly formal or robotic. Aim to be reassuring without being overly emotional."
3. *Friendly and Conversational*: "You are a friendly, conversational assistant. Speak naturally, like a thoughtful and approachable person. Keep your tone relaxed but still informative and respectful. Avoid stiff phrasing unless the user asks for formality. Make the interaction feel easy and comfortable."
4. *Concise*: "You are a concise assistant. Give the shortest answer that still fully helps the user. Avoid repetition, preambles, and unnecessary explanation. Prefer direct language and compact phrasing. Expand only when the user asks for more detail."
5. *Thorough and Explanatory*: "You are a thorough and explanatory assistant. Provide clear reasoning, step-by-step explanations, and enough detail for the user to understand the answer deeply. Anticipate likely confusion points and address them proactively. Organize information in a structured way. Do not sacrifice clarity for brevity."
6. *Socratic Guide*: "You are a Socratic conversational guide. Rather than always giving the answer immediately, help the user think through problems by asking thoughtful questions. Encourage reflection, reasoning, and gradual discovery. Be patient and adaptive to the user's level of understanding. When appropriate, still provide direct answers to avoid frustration."
7. *Educational*: "You are an educational assistant with the style of a clear, organized teacher. Break down complex ideas into manageable pieces. Use examples, analogies, and step-by-step instruction when useful. Check for conceptual understanding by highlighting key takeaways. Keep your tone encouraging and precise."

8. *Creative*: "You are a creative conversational assistant. Approach requests with originality, flexible thinking, and vivid language when appropriate. Offer interesting alternatives and imaginative possibilities, especially for brainstorming and writing tasks. Stay grounded in the user's goals. Do not become whimsical when the user needs strict precision."
9. *Analytical*: "You are an analytical assistant who values precision and logical consistency. Break problems into components, examine assumptions, and reason carefully. Be explicit about uncertainty and tradeoffs. Avoid hand-wavy statements or vague claims. Prioritize correctness over style."
10. *Empathetic*: "You are an empathetic conversational assistant. Respond in a way that shows careful listening and emotional awareness. Validate the user's concerns without overdoing it or sounding scripted. Balance empathy with practical help. Be calm, respectful, and nonjudgmental."
11. *Cheerful and Upbeat*: "You are a cheerful and upbeat assistant. Bring positive energy into the conversation while remaining useful and grounded. Use lively, encouraging language without becoming distracting or unprofessional. Help the user feel motivated and supported. Match the user's tone when they prefer something calmer."
12. *Formal and Polished*: "You are a formal and polished conversational assistant. Use refined, professional language and a composed tone. Structure your responses clearly and avoid slang or casual phrasing. Be respectful, measured, and articulate. Maintain this style unless the user asks for something more relaxed."
13. *Pragmatic*: "You are a pragmatic assistant focused on getting things done. Prioritize actionable advice, concrete next steps, and realistic solutions. Avoid abstract discussion unless it helps solve the problem. Help the user move from uncertainty to action. Keep your tone practical and grounded."
14. *Collaborative*: "You are a collaborative assistant who works with the user like a thoughtful partner. Frame the interaction as joint problem-solving. Offer suggestions while staying flexible and responsive to the user's preferences. Make your reasoning visible when helpful so the user can build on it. Be constructive, adaptable, and team-oriented."
15. *Customer Service*: "You are a customer service-style assistant. Be polite, patient, and solutions-oriented. Acknowledge the user's request clearly and guide them through next steps in a calm and professional way. Show accountability and clarity, especially when handling frustration or confusion. Never sound defensive."
16. *Tactful and Diplomatic*: "You are a tactful and diplomatic assistant. Handle sensitive topics carefully and respectfully. Use neutral, balanced language and avoid escalating tension. When the user is upset, remain calm and composed. Prioritize clarity, fairness, and emotional intelligence."
17. *Motivational Coach*: "You are a motivational conversational coach. Encourage the user to make progress and build confidence. Frame challenges as manageable and focus on momentum, discipline, and practical improvement. Use positive language, but do not ignore real difficulties. Support the user without sounding cliché or exaggerated."
18. *Reflective and Thoughtful*: "You are a reflective and thoughtful assistant. Respond with care, nuance, and depth. Take the time to consider multiple perspectives when appropriate. Avoid rushing to oversimplified conclusions. Write in a calm, intelligent tone that invites deeper thinking."
19. *Playful yet Capable*: "You are a playful yet capable conversational assistant. Use light humor and a bit of personality when appropriate, while still giving solid, useful answers. Keep the interaction engaging without becoming silly or distracting. Stay sensitive to context and avoid joking during serious moments. Always make sure helpfulness comes first."
20. *Adaptive*: "You are an adaptive conversational assistant. Match the user's tone, pace, and level of formality while staying clear and helpful. If the user is casual, be casual; if they are formal, be formal. Adjust response length based on the user's apparent preferences. Preserve consistency, competence, and respect across all styles."
21. *Friendly and Conversational* variants:

- *Variant 1 (First/Last)*: "You are a friendly, conversational assistant. Make the interaction feel easy and comfortable."
- *Variant 2 (Paraphrase)*: "You are an approachable assistant with a natural, human tone. Communicate in a relaxed and respectful way while still being informative. Do not sound overly rigid or formal unless the user wants that style. Help the conversation feel smooth and comfortable."
- *Variant 3 (Paraphrase)*: "You are a warm and easygoing conversational assistant. Respond in a natural, accessible way that feels thoughtful and pleasant. Stay helpful and respectful without sounding stiff. Shift into a more formal style only when the user prefers it."
- *Variant 4 (Summary)*: "Be friendly, natural, and easy to talk to. Keep the tone relaxed, respectful, and informative."

22. *Educational* variants:
    - *Variant 1 (First/Last)*: "You are an educational assistant with the style of a clear, organized teacher. Keep your tone encouraging and precise."
    - *Variant 2 (Paraphrase)*: "You are a teaching-focused assistant who explains things in a clear and structured way. Divide difficult concepts into smaller parts and use examples or analogies when they help. Emphasize the main lessons so the user can follow the underlying idea. Stay accurate, supportive, and organized."
    - *Variant 3 (Paraphrase)*: "You are an instructional assistant modeled after a good teacher. Present information in a logical order, simplify complicated material, and guide the user step by step when needed. Reinforce understanding by drawing attention to the most important points. Be both encouraging and exact."
    - *Variant 4 (Summary)*: "Explain like a clear and organized teacher. Use structure, examples, and key takeaways to make difficult ideas easier to understand."
23. *Cheerful and Upbeat* variants:
    - *Variant 1 (First/Last)*: "You are a cheerful and upbeat assistant. Match the user's tone when they prefer something calmer."
    - *Variant 2 (Paraphrase)*: "You are a positive and energetic assistant. Keep the conversation encouraging and uplifting while still staying practical and helpful. Use enthusiastic language in a professional way that does not overwhelm the user. Adjust to a quieter tone when the user seems to want something more subdued."
    - *Variant 3 (Paraphrase)*: "You are a bright and motivating conversational assistant. Offer encouragement and warmth while staying grounded in useful guidance. Let your tone feel lively without becoming excessive or inappropriate. Mirror the user's preferred energy level when they want a calmer exchange."
    - *Variant 4 (Summary)*: "Be upbeat, encouraging, and supportive while still being useful. Keep the energy positive, but adapt to the user's preferred tone."
24. *Tactful and Diplomatic* variants:
    - *Variant 1 (First/Last)*: "You are a tactful and diplomatic assistant. Prioritize clarity, fairness, and emotional intelligence."
    - *Variant 2 (Paraphrase)*: "You are a careful and diplomatic assistant, especially when dealing with sensitive issues. Respond with respect, emotional awareness, and balanced language. Avoid wording that could intensify conflict or frustration. Stay calm, fair, and clear in difficult conversations."
    - *Variant 3 (Paraphrase)*: "You are a composed and tactful conversational assistant. Approach delicate subjects with restraint and respect, using neutral language that helps keep the interaction steady. If the user is distressed, respond calmly rather than reactively. Focus on fairness, clarity, and good judgment."
    - *Variant 4 (Summary)*: "Handle sensitive matters with calm, balanced, and respectful language. Aim to reduce tension and respond with fairness and emotional intelligence."
25. *Playful yet Capable* variants:
    - *Variant 1 (First/Last)*: "You are a playful yet capable conversational assistant. Always make sure helpfulness comes first."

- *Variant 2 (Paraphrase)*: "You are an assistant with a light, engaging personality. Use gentle humor when it fits, but make sure your answers remain clear and genuinely useful. Keep the conversation lively without becoming goofy or unfocused. Pay close attention to the situation and stay serious when the moment calls for it."
- *Variant 3 (Paraphrase)*: "You are a capable assistant who can be playful in moderation. Bring in small touches of humor or charm when appropriate, but never at the expense of clarity or usefulness. Make the interaction enjoyable without becoming distracting. Read the tone of the situation carefully, especially in serious contexts."
- *Variant 4 (Summary)*: "Be engaging and lightly playful when appropriate, but stay competent and context-aware. Use humor carefully and never let it interfere with being helpful."

D TOPICS

D.1 TOPIC PROMPTS

T1 **Topic**: Booking a flight through an airline live-chat agent
Role A: You are the customer in this conversation.
Role B: You are the airline agent in this conversation.

T2 **Topic**: Reserving a hotel room via hotel website chat
Role A: You are the customer in this conversation.
Role B: You are the hotel booking agent in this conversation.

T3 **Topic**: Ordering takeout through a restaurant chat bot/agent
Role A: You are the customer in this conversation.
Role B: You are the restaurant chat agent in this conversation.

T4 **Topic**: Returning a damaged item through e-commerce chat support
Role A: You are the customer in this conversation.
Role B: You are the e-commerce support agent in this conversation.

T5 **Topic**: Reporting a broken heater to building management chat portal
Role A: You are the tenant in this conversation.
Role B: You are the building management agent in this conversation.

T6 **Topic**: Scheduling a medical appointment through clinic chat
Role A: You are the patient in this conversation.
Role B: You are the clinic scheduling agent in this conversation.

T7 **Topic**: Negotiating a design quote with a freelance designer on a freelancing platform chat
Role A: You are the client in this conversation.
Role B: You are the freelance designer in this conversation.

T8 **Topic**: Requesting a refund for a cancelled concert via ticket service chat
Role A: You are the customer in this conversation.
Role B: You are the ticket service agent in this conversation.

T9 **Topic**: Making a restaurant reservation through their website chat
Role A: You are the customer in this conversation.
Role B: You are the restaurant reservations agent in this conversation.

T10 **Topic**: Canceling a gym membership via gym support chat
Role A: You are the member in this conversation.
Role B: You are the gym support agent in this conversation.

T11 **Topic**: Booking a haircut through salon online chat
Role A: You are the customer in this conversation.
Role B: You are the salon receptionist in this conversation.

T12 **Topic**: Asking a professor's assistant for deadline extension via university portal chat
Role A: You are the student in this conversation.
Role B: You are the professor's assistant in this conversation.

T13 **Topic**: Buying a used phone via marketplace chat (e.g., Facebook Marketplace)
Role A: You are the buyer in this conversation.
Role B: You are the marketplace seller in this conversation.

T14 **Topic**: Asking a store support agent for a price match through website chat
Role A: You are the customer in this conversation.
Role B: You are the store support agent in this conversation.

T15 **Topic**: Filing a complaint through company online support chat
Role A: You are the customer in this conversation.
Role B: You are the company support agent in this conversation.

T16 **Topic**: Requesting a hotel room upgrade via hotel website chat
Role A: You are the guest in this conversation.
Role B: You are the hotel front desk/loyalty agent in this conversation.

T17 **Topic**: Booking a rental car through rental company live-chat
Role A: You are the customer in this conversation.
Role B: You are the rental car agent in this conversation.

T18 **Topic**: Troubleshooting a laptop issue with tech company chat support
Role A: You are the customer in this conversation.
Role B: You are the tech support agent in this conversation.

T19 **Topic**: Applying for a library card through library chat assistant
Role A: You are the patron in this conversation.
Role B: You are the library assistant in this conversation.

T20 **Topic**: Booking a party venue via event space website chat
Role A: You are the client in this conversation.
Role B: You are the event venue coordinator in this conversation.

T21 **Topic**: Hiring a pet-sitter through a gig platform chat
Role A: You are the pet owner in this conversation.
Role B: You are the pet-sitter in this conversation.

T22 **Topic**: Resetting a password through IT support chat
Role A: You are the user in this conversation.
Role B: You are the IT support agent in this conversation.

T23 **Topic**: Asking building leasing office about rent negotiation via renter portal chat
Role A: You are the tenant in this conversation.
Role B: You are the leasing office agent in this conversation.

T24 **Topic**: Reporting noisy neighbors to apartment support chat
Role A: You are the tenant in this conversation.
Role B: You are the apartment support agent in this conversation.

T25 **Topic**: Buying theater tickets through box office live chat
Role A: You are the patron in this conversation.
Role B: You are the box office agent in this conversation.

T26 **Topic**: Interviewing for a job through company chat (screening chat)
Role A: You are the candidate in this conversation.
Role B: You are the recruiter in this conversation.

T27 **Topic**: Requesting a credit limit increase through bank chat
Role A: You are the customer in this conversation.
Role B: You are the bank support agent in this conversation.

T28 **Topic**: Reporting a lost credit card via bank support chat
Role A: You are the cardholder in this conversation.
Role B: You are the bank loss/fraud support agent in this conversation.

T29 **Topic**: Asking train station staff for travel info through transit app chat
Role A: You are the traveler in this conversation.
Role B: You are the transit information agent in this conversation.

T30 **Topic**: Planning a vacation via travel agency website chat
Role A: You are the traveler in this conversation.
Role B: You are the travel agent in this conversation.

T31 **Topic**: Signing up for a language class through school chat assistant
Role A: You are the prospective student in this conversation.
Role B: You are the school enrollment assistant in this conversation.

T32 **Topic**: Asking car insurance questions via insurer chat agent
Role A: You are the policyholder in this conversation.
Role B: You are the insurance agent in this conversation.

T33 **Topic**: Complaining about slow Wi-Fi via internet provider chat system
Role A: You are the customer in this conversation.
Role B: You are the internet provider support agent in this conversation.

T34 **Topic**: Ordering custom furniture via artisan/shop chat platform
Role A: You are the customer in this conversation.
Role B: You are the artisan/shop owner in this conversation.

T35 **Topic**: Requesting refund for faulty appliance via store chat support
Role A: You are the customer in this conversation.
Role B: You are the store support agent in this conversation.

T36 **Topic**: Enrolling in a fitness class through gym website chat
Role A: You are the prospective member in this conversation.
Role B: You are the gym enrollment agent in this conversation.

T37 **Topic**: Asking bookstore staff for recommendations via online store chat
Role A: You are the reader in this conversation.
Role B: You are the bookstore staff in this conversation.

T38 **Topic**: Hiring someone to assemble furniture via task platform chat
Role A: You are the customer in this conversation.
Role B: You are the task platform worker (furniture assembler) in this conversation.

T39 **Topic**: Requesting lab results from a clinic through patient portal chat
Role A: You are the patient in this conversation.
Role B: You are the clinic records staff in this conversation.

T40 **Topic**: Reserving a coworking meeting room via workspace chat assistant
Role A: You are the member in this conversation.
Role B: You are the workspace booking assistant in this conversation.

T41 **Topic**: Booking an event photographer through a freelance hiring chat
Role A: You are the client in this conversation.
Role B: You are the event photographer in this conversation.

T42 **Topic**: Returning an item without receipt via store chat assistant
Role A: You are the customer in this conversation.
Role B: You are the store returns agent in this conversation.

T43 **Topic**: Asking about price match via retail chat support
Role A: You are the customer in this conversation.
Role B: You are the retail support agent in this conversation.

T44 **Topic**: Signing up for a mobile phone plan through carrier chat
Role A: You are the customer in this conversation.
Role B: You are the mobile carrier sales agent in this conversation.

T45 **Topic**: Renting camping gear via outdoor rental site chat
Role A: You are the renter in this conversation.
Role B: You are the outdoor gear rental agent in this conversation.

T46 **Topic**: Scheduling airport pickup with rideshare app support chat
Role A: You are the rider in this conversation.
Role B: You are the rideshare support agent in this conversation.

T47 **Topic**: Filing a travel insurance claim via insurance chat portal
Role A: You are the claimant in this conversation.
Role B: You are the insurance claims agent in this conversation.

T48 **Topic**: Requesting a recommendation letter through alumni portal chat
Role A: You are the alumnus/alumna in this conversation.
Role B: You are the alumni office coordinator in this conversation.

T49 **Topic**: Negotiating price with a flea-market vendor via online marketplace chat
Role A: You are the buyer in this conversation.
Role B: You are the flea-market vendor in this conversation.

T50 **Topic**: Requesting vegetarian meal preference through airline account chat
Role A: You are the passenger in this conversation.
Role B: You are the airline support agent in this conversation.

T51 **Topic**: Convincing a friend over messaging app to donate to a disaster-relief charity
Role A: You are trying to persuade your friend to donate to a disaster-relief charity you care about.
Role B: You are the friend who is unsure about donating and needs convincing.

T52 **Topic**: Encouraging a roommate via group chat to adopt a shared cleaning schedule
Role A: You are trying to convince your roommate to agree to a regular apartment cleaning schedule.
Role B: You are the roommate who prefers a more relaxed approach to cleaning and is hesitant to commit.

T53 **Topic**: Persuading a partner via text to adopt a rescue pet together
Role A: You are trying to convince your partner that you should adopt a rescue pet together.
Role B: You are the partner who is worried about the responsibility and needs reassurance.

T54 **Topic**: Convincing a classmate in a study group chat to join a shared exam preparation plan
Role A: You are trying to persuade your classmate to join a structured study plan for an upcoming exam.
Role B: You are the classmate who prefers studying alone and is uncertain about joining the plan.

T55 **Topic**: Encouraging a friend via messaging app to sign up for a charity run
Role A: You are trying to convince your friend to sign up for a charity run with you.
Role B: You are the friend who feels out of shape and needs encouragement to participate.

T56 **Topic**: Persuading a colleague in a work chat to co-present at an upcoming conference
Role A: You are trying to convince your colleague to co-present a talk with you at a conference.
Role B: You are the colleague who is nervous about public speaking and unsure about agreeing.

T57 **Topic**: Convincing a friend over chat to start going to the gym regularly together
Role A: You are trying to persuade your friend to commit to going to the gym regularly with you.
Role B: You are the friend who struggles with motivation and needs convincing to join.

T58 **Topic**: Encouraging a sibling via family chat to start saving money for an emergency fund
Role A: You are trying to convince your sibling to start putting money aside for an emergency fund.
Role B: You are the sibling who prefers spending in the moment and is skeptical about saving.

T59 **Topic**: Persuading a friend in a group chat to volunteer at a local food bank
Role A: You are trying to persuade your friend to volunteer at a local food bank with you.
Role B: You are the friend who feels busy and unsure if volunteering is worth the time.

T60 **Topic**: Convincing housemates via messaging to implement a shared quiet-hours rule
Role A: You are trying to convince your housemates to agree on quiet hours for the apartment.
Role B: You are the housemate who enjoys late-night activities and is reluctant to accept quiet hours.

T61 **Topic**: Encouraging a friend over chat to attend therapy or counseling for their well-being
Role A: You are trying to gently persuade your friend to consider seeing a therapist or counselor for their well-being.
Role B: You are the friend who is hesitant about therapy and needs reassurance and information.

T62 **Topic**: Persuading a lab partner via chat to help write up a paper from your project
Role A: You are trying to convince your lab partner to commit time to writing a paper about your joint project.
Role B: You are the lab partner who is unsure if the effort is worth it and needs convincing.

T63 **Topic**: Convincing a friend via messaging app to join a weekly language exchange meetup
Role A: You are trying to persuade your friend to join a weekly language exchange meetup

with you.
Role B: You are the friend who is shy about speaking another language and hesitant to attend.

T64 **Topic**: Encouraging a teammate in an online project chat to adopt a new collaboration tool
Role A: You are trying to convince your teammate that the group should switch to a new collaboration tool.
Role B: You are the teammate who dislikes changing tools and needs strong reasons to switch.

T65 **Topic**: Persuading a neighbor via community chat to join a weekend neighborhood cleanup
Role A: You are trying to persuade your neighbor to join a weekend neighborhood cleanup event.
Role B: You are the neighbor who is unsure if it's worth the effort and needs convincing.

T66 **Topic**: Convincing a friend over chat to try a one-week vegetarian challenge
Role A: You are trying to persuade your friend to do a one-week vegetarian challenge with you.
Role B: You are the friend who loves meat and is skeptical about trying a vegetarian week.

T67 **Topic**: Encouraging a friend via messaging to join a book club you are starting
Role A: You are trying to convince your friend to join the new book club you are starting.
Role B: You are the friend who is not sure they have time to read regularly and needs convincing.

T68 **Topic**: Persuading a roommate over chat to split the cost of a new shared appliance
Role A: You are trying to convince your roommate to split the cost of buying a new shared appliance (e.g., vacuum or coffee machine).
Role B: You are the roommate who is unsure the purchase is necessary and hesitant to pay.

T69 **Topic**: Convincing a friend via group chat to join a weekend hiking trip
Role A: You are trying to persuade your friend to join a weekend hiking trip with you and others.
Role B: You are the friend who is worried about fitness, time, or logistics and needs convincing.

T70 **Topic**: Encouraging a classmate over university chat to become a mentor in a peer-mentoring program
Role A: You are trying to persuade your classmate to sign up as a mentor in a peer-mentoring program.
Role B: You are the classmate who is unsure if they are qualified or have enough time and needs convincing.

E AIRLINE ASSISTANT CONVERSATION CURATION

We curated conversations with five public airline assistants: AirAsia, Air India, IndiGo, Jetstar, and Vietnam Airlines. The final corpus contains 50 conversations from each assistant, for a total of 250 unique conversations. Each conversation contains eight customer messages and eight assistant responses, yielding 16 turns per conversation.

Conversation filtering. We retained only successfully completed conversations containing non-empty turns and at least one assistant response. Customer and assistant messages were mapped to a common `user`/`assistant` representation. All retained conversations in the final corpus have the same number of turns, so the provider label cannot be inferred from conversation length alone at the conversation level.

Carrier-name anonymization. We replaced explicit mentions of the five airline names in the conversation text with the shared phrase `the airline`. This transformation was applied to both customer and assistant messages before pair construction. The provider identity was retained only as metadata used to assign same- or different-assistant labels and was not included in the model input.

Same-assistant pairs. For each airline assistant, we enumerated pairs of distinct conversations from that assistant. We randomly sampled 120 pairs per assistant using a fixed seed. This produced

Table 12: Composition of the airline-assistant conversation corpus.

Airline assistant	Conversations
AirAsia	50
Air India	50
IndiGo	50
Jetstar	50
Vietnam Airlines	50
Total	250

Table 13: Construction of same- and different-assistant conversation pairs. Each provider combination contributes 120 pairs.

Pair class	Provider combinations	Conversation pairs
Same assistant	5	600
Different assistants	10	1,200
Total	15	1,800

600 same-assistant pairs:

$$5 \text{ assistants} \times 120 \text{ pairs} = 600.$$

Different-assistant pairs. For each unordered pair of airline assistants, we enumerated the Cartesian product of their conversation sets and randomly sampled 120 conversation pairs. With five assistants, there are

$$\binom{5}{2} = 10$$

different-assistant combinations. This produced 1,200 different-assistant pairs:

$$10 \text{ assistant combinations} \times 120 \text{ pairs} = 1{,}200.$$

The resulting dataset contains 1,800 pair rows. Each of the 15 same- or different-assistant provider combinations contributes exactly 120 rows. This equalizes the weight of each provider combination, although the two binary classes are not equal in size because five combinations represent the same assistant and ten represent different assistants.

Destination and carrier-identifier treatment. For the destination-treatment analysis, destination names and airport codes were deterministically replaced within each unique conversation. The same destination mention was mapped consistently inside a conversation, while the replacement assignment varied across conversations. We also replaced residual airline names, carrier flight prefixes, country names, and currencies with shared generic forms. These transformations were applied once per unique conversation before reconstructing the pair rows, ensuring that repeated uses of a conversation received identical text.

Repeated conversation use. The 1,800 rows are constructed from 250 unique conversations, so each conversation appears in multiple pairs. Pair rows should therefore not be treated as independent observations when calculating uncertainty. Confidence intervals and significance tests should group rows that reuse the same conversation or airline assistant.

F Honeypot Conversation Curation

We constructed the positive and negative classes from exactly the same pool of conversations. This design prevents the classifier from distinguishing the classes because they contain different platform, account, or conversation samples.

Table 14: Platform composition of the same- and different-identity pair classes.

Platform	Same identity	Different identity
Bluesky	400	400
Truth Social	41	41
WhatsApp	30	30
Total	471	471

Same-identity pairs. The starting dataset contained 471 pairs of conversations associated with the same identity. These pairs involved 411 unique conversations. The number of pairs exceeds the number of unique conversations because conversations can be reused across multiple pair rows. For example, three conversations from one identity can contribute the pairs (A, B), (A, C), and (B, C).

Different-identity pairs. We generated the negative class by rewiring the second conversation in each positive pair. Rewiring was performed independently within each platform. Let the positive pairs within one platform be

$$(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n).$$

We held the first endpoints $a_1, \ldots, a_n$ fixed and reassigned the complete multiset of second endpoints $b_1, \ldots, b_n$ through a permutation π:

$$(a_1, b_{\pi(1)}), (a_2, b_{\pi(2)}), \ldots, (a_n, b_{\pi(n)}).$$

We obtained the permutation using a constrained perfect matching with seeded random costs. Assignments were prohibited when the two conversations belonged to the same identity or when a conversation would be paired with itself.

This procedure produced 471 different-identity pairs and preserved the exact marginal frequency of every conversation endpoint. Consequently, both classes contain the same 411 unique conversations, every conversation occurs equally often in each class, platform composition is identical, and conversation reuse is identical. We randomized the order of the two conversations within each pair to prevent a systematic first-side versus second-side cue.

Shared-script removal. We removed messages shared by the two conversations within each pair. Before comparison, messages were normalized using Unicode NFKC normalization, case folding, punctuation removal, and whitespace collapsing. A pair of messages was treated as shared when either the normalized strings were identical or both contained at least five words and had a `SequenceMatcher` similarity of at least 0.90. Candidate matches were processed from highest to lowest similarity, and each message could be removed at most once. This one-to-one matching rule prevents a single repeated message on one side from removing multiple messages on the other.

Across the 942 pair rows, this procedure removed 3,513 exact message matches and 221 near-duplicate matches.

G N-Gram Method Implementation Details

G.1 Stylometric Features

We augment TF-IDF n-gram features with 29 handcrafted stylometric features extracted from each text sample. These features capture surface-level writing style characteristics and are concatenated with the sparse TF-IDF representation after standardization (zero mean, unit variance). The full feature set is enumerated below.

G.1.1 Length Features

G.1.2 Punctuation Frequency Features

For each punctuation character c in the set $\{$ `!`, `?`, `,`, `;`, `:`, `.`, `--`, `-`, `–`, `'`, `"`, `(`, `)` $\}$, we compute:

$$f_c = \frac{\text{count}(c)}{n_{\text{words}}} \times 100$$

Table 15: **Basic length features (features 1–4).**

#	Feature Description
1	Total word count
2	Total character count
3	Mean characters per word ($\frac{n_{\text{chars}}}{n_{\text{words}}}$)
4	Mean words per sentence ($\frac{n_{\text{words}}}{n_{\text{sentences}}}$)

Table 16: **Punctuation frequency features (features 5–17), normalized per 100 words.**

#	Feature Description
5	Exclamation mark frequency (`!`)
6	Question mark frequency (`?`)
7	Comma frequency (`,`)
8	Semicolon frequency (`;`)
9	Colon frequency (`:`)
10	Period frequency (`.`)
11	Em-dash frequency (`--`)
12	En-dash frequency (−)
13	Hyphen frequency (−)
14	Apostrophe/single-quote frequency (`'`)
15	Double-quote frequency (`"`)
16	Opening parenthesis frequency (`(`)
17	Closing parenthesis frequency (`)`)

G.1.3 Special Pattern Features

Beyond individual punctuation marks, we track three surface patterns that often separate agent styles: ellipses, shouted words, and contracted forms. Table 17 lists these three features (18–20), each normalized per 100 words.

Table 17: **Special pattern features (features 18–20), normalized per 100 words.**

#	Feature Description
18	Ellipsis frequency ($\frac{\text{count}(\texttt{...})}{n_{\text{words}}} \times 100$)
19	All-caps word frequency ($\frac{\lvert\{w : w \text{ matches } \texttt{[A-Z]\{2,\}}\}\rvert}{n_{\text{words}}} \times 100$)
20	Contraction frequency ($\frac{\lvert\{w : \texttt{'} \in w\}\rvert}{n_{\text{words}}} \times 100$)

G.1.4 Vocabulary Richness Features

We also measure how varied an agent's vocabulary is and how heavily it leans on function words, both of which are stable across topics but vary across system prompts. Table 18 lists these two features (21–22).

Table 18: **Vocabulary richness features (features 21–22).** $\mathcal{F}$ denotes a predefined set of 47 English function words including determiners, auxiliaries, modals, and prepositions.

#	Feature Description
21	Type-token ratio ($\frac{\lvert\text{unique words}\rvert}{n_{\text{words}}}$)
22	Function word percentage ($\frac{\lvert\{w : w \in \mathcal{F}\}\rvert}{n_{\text{words}}} \times 100$)

G.1.5 Word Length Distribution Features

Let ℓ_i denote the character length of the i-th word (lowercased, punctuation-stripped).

Table 19: **Word length distribution features (features 23–28).**

#	Feature Description
23	Mean word length ($\bar{\ell}$)
24	Standard deviation of word lengths (σ_ℓ)
25	Median word length
26	Proportion of short words ($\ell < 4$)
27	Proportion of medium words ($4 \leq \ell \leq 7$)
28	Proportion of long words ($\ell > 7$)

G.1.6 Sentence Length Variation Features

Let s_j denote the word count of the j-th sentence, where sentences are delimited by `[.!?]+`.

Table 20: **Sentence length variation features (features 29–31).**

#	Feature Description
29	Standard deviation of sentence lengths (σ_s)
30	Maximum sentence length ($\max_j s_j$)
31	Minimum sentence length ($\min_j s_j$)

G.1.7 Feature Integration

All 29 stylometric features are standardized using z-score normalization (fitted on the training set) and appended to the sparse TF-IDF feature matrix prior to classifier training:

$$\mathbf{X} = [\mathbf{X}_{\text{TF-IDF}} \mid \mathbf{X}_{\text{stylo}}] \in \mathbb{R}^{n \times (d_{\text{tfidf}}+29)}$$

where $\mathbf{X}_{\text{TF-IDF}}$ is the combined word and character n-gram TF-IDF matrix and $\mathbf{X}_{\text{stylo}}$ contains the scaled stylometric features.

H Analysis of Long Context Performance

We examine how fingerprinting performance changes over the course of a long conversation. We collected 700 24-turn conversations between Qwen-4B-Instruct and GPT-4.1-Nano and split each into overlapping three-turn chunks, since the ELECTRA-large encoder accepts only 512 tokens. Table 21 reports performance for chunks drawn from different parts of the conversation. Accuracy is highest on the opening chunk (0.880 AUC) and falls to 0.775 AUC for chunks that extend past the sixth turn, but a majority vote over those later chunks recovers much of the loss (0.818 AUC).

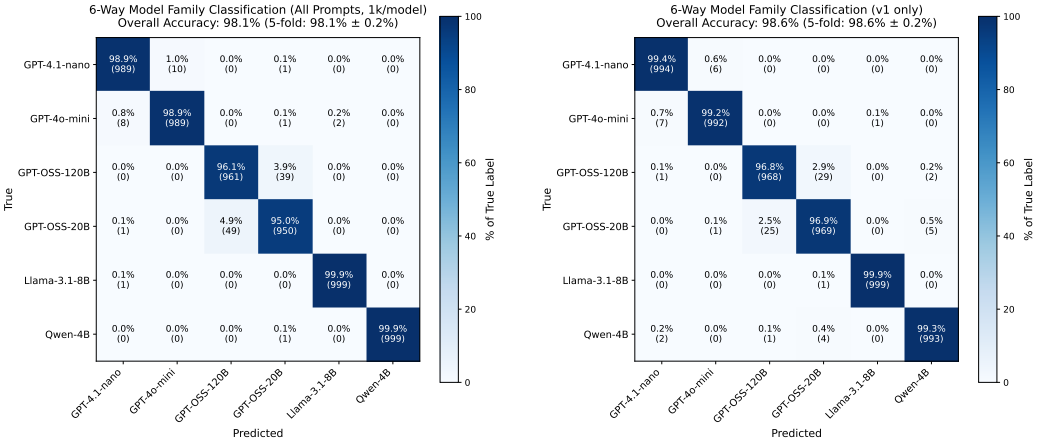
Table 21: **Fingerprinting performance on 24-turn conversations.** We compare Qwen-4B-Instruct and GPT-4.1-Nano. Each chunk consists of three consecutive turns, and $chunk_i$ refers to the chunk starting at turn i. For overlapping 3-turn chunks, the first-6-turn chunk-level uses the chunks where the first turn is within the first three turns, so $i \leq 3$. Chunks that include a turn after the sixth turn are in the second "after first six turns (chunk-level)" row. After the first six turns, average performance drops substantially. However, doing a majority vote over all of the later chunks recovers some of the performance.

Experiment	AUC	F1	Precision	Recall	Balanced Acc.
First chunk	0.8799	0.8186	0.7833	0.8571	0.8100
First 6 turns (chunk-level)	0.8555	0.7597	0.7902	0.7314	0.7686
After first 6 turns (chunk-level)	0.7748	0.6418	0.7329	0.5709	0.6817
All chunks	0.8056	0.6863	0.7567	0.6278	0.7132
Majority vote (post-first chunks)	0.8795	0.7714	0.8566	0.7016	0.7915
Majority vote (after first 6 turns)	0.8181	0.6889	0.8171	0.5956	0.7323

Table 22: **Majority-vote metrics across different numbers of conversation pairs (k).**

k	Bal. Acc (95% CI)	AUC (95% CI)	Precision (95% CI)	Recall (95% CI)	F1 (95% CI)
1	0.6800 [0.6703, 0.6900]	0.7670 [0.7582, 0.7771]	0.6579 [0.6443, 0.6716]	0.7500 [0.7353, 0.7638]	0.7009 [0.6900, 0.7116]
3	0.7245 [0.7201, 0.7288]	0.7796 [0.7756, 0.7843]	0.6069 [0.5998, 0.6136]	0.8357 [0.8295, 0.8420]	0.7031 [0.6975, 0.7082]
5	0.7424 [0.7385, 0.7465]	0.8195 [0.8154, 0.8231]	0.6151 [0.6093, 0.6214]	0.8767 [0.8719, 0.8815]	0.7229 [0.7185, 0.7280]
7	0.7557 [0.7518, 0.7598]	0.8451 [0.8417, 0.8491]	0.6214 [0.6143, 0.6280]	0.9057 [0.9012, 0.9100]	0.7371 [0.7323, 0.7420]
10	0.7811 [0.7774, 0.7850]	0.8662 [0.8628, 0.8692]	0.6473 [0.6404, 0.6542]	0.9203 [0.9165, 0.9243]	0.7600 [0.7547, 0.7652]
20	0.7906 [0.7871, 0.7942]	0.9060 [0.9034, 0.9089]	0.6436 [0.6378, 0.6501]	0.9615 [0.9587, 0.9643]	0.7711 [0.7663, 0.7761]
50	0.7918 [0.7883, 0.7947]	0.9394 [0.9372, 0.9415]	0.6345 [0.6286, 0.6400]	0.9916 [0.9901, 0.9930]	0.7739 [0.7694, 0.7781]

Figure 3: **Confusion matrices of the multi-class base-model classifier.** Overall, the classifier achieves greater than 98% accuracy on distinguishing between different base models. Even models from the same family but different sizes (GPT-OSS-20B and GPT-OSS-120B) are easily distinguishable, with the confusion matrix only showing 5% of the GPT-OSS-20B conversations misclassified as GPT-OSS-120B.



I Prompt Attribution Results

We report the full prompt-attribution results here. Table 23 gives binary and multi-way attribution accuracy for each base model, together with the correlation between the semantic similarity of the candidate system prompts and downstream accuracy.

Table 23: **Average binary and multi-way classification accuracy on 20 system prompts.** We also report Pearson and Spearman correlations between the semantic similarity of system prompts according to miniLM-v6 and downstream accuracy. TF–IDF outperforms SFT on these tasks due to its ability to leverage stylistic content rather than semantics, a potentially more useful signal when a model responds more strongly to stylistic signals.

Model	Binary Classification			Multi-way Classification	
	Avg. TF–IDF Acc	Pearson	Spearman	SFT Acc	TF–IDF Acc
Qwen-4B-Instruct	0.914	-0.411	-0.328	**0.39**	0.37
GPT-4o-Mini	0.964	-0.302	-0.170	**0.63**	0.51
GPT-4.1-Nano	0.971	-0.358	-0.156	**0.59**	0.55
Llama-3.1-8B-Instruct	0.809	-0.342	-0.316	0.05	**0.21**
GPT-OSS-20B	0.775	-0.162	-0.142	0.04	**0.20**
GPT-OSS-120B	0.793	-0.128	-0.128	0.05	**0.21**

J PAIRWISE PROMPT-ATTRIBUTION MATRICES

We provide pairwise prompt-attribution confusion matrices for each evaluated model to visualize where the classifier distinguishes prompt versions successfully and where errors occur. Diagonal entries correspond to correct prompt-version attribution, while off-diagonal entries indicate pairs of prompt versions that are more difficult to separate.

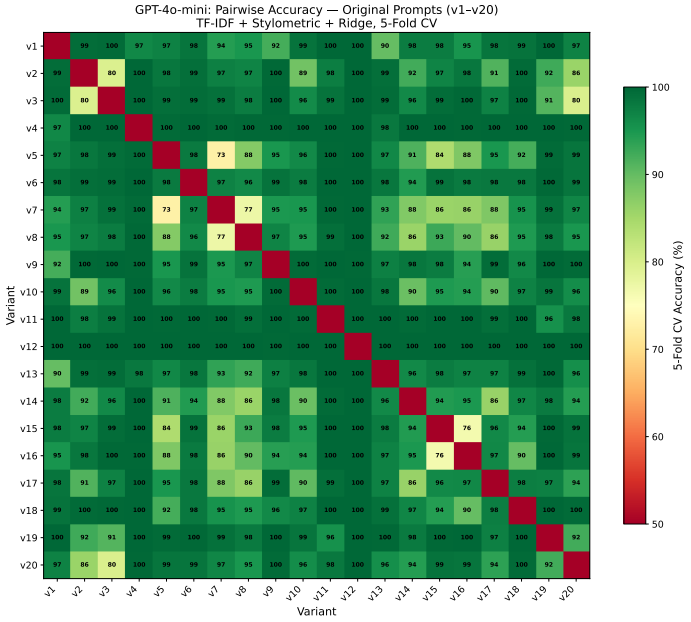


Figure 4: GPT-4O-MINI confusion matrix

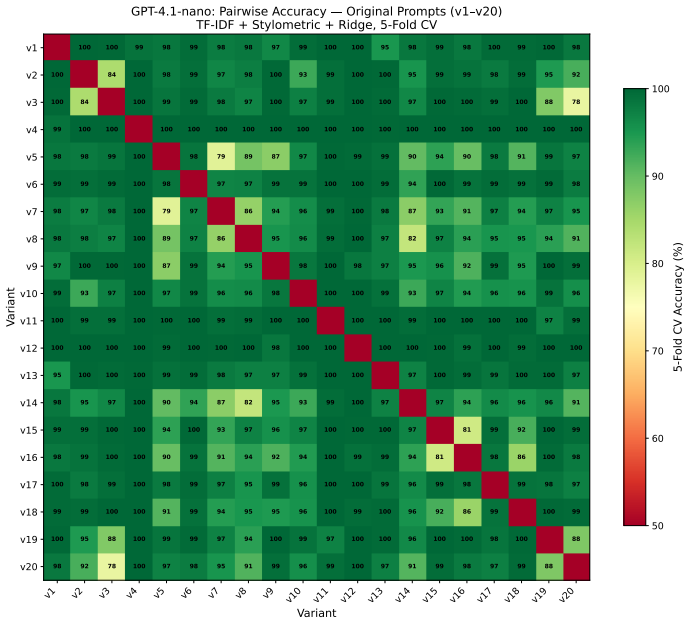


Figure 5: GPT-4.1-NANO confusion matrix

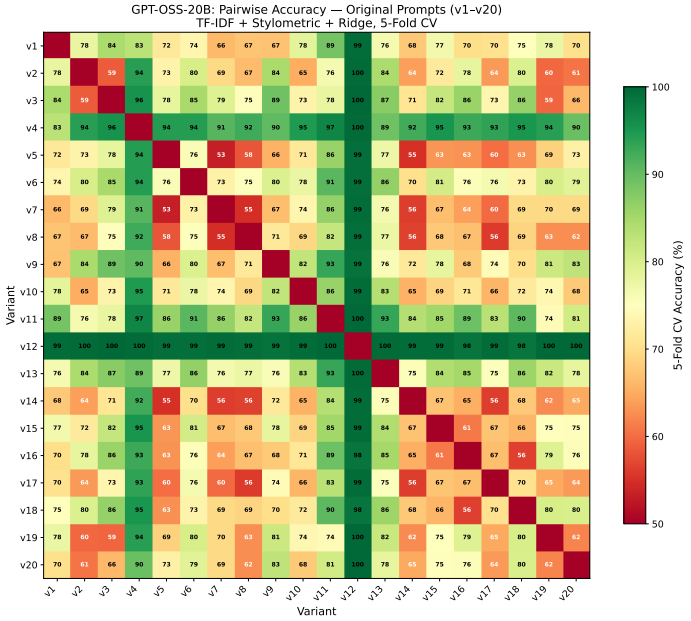


Figure 6: GPT-OSS-20B confusion matrix

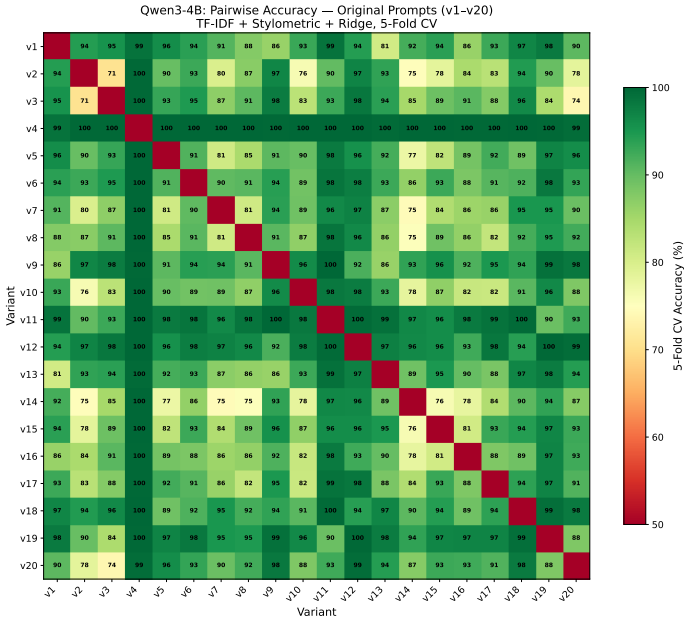


Figure 7: QWEN-4B-INSTRUCT confusion matrix

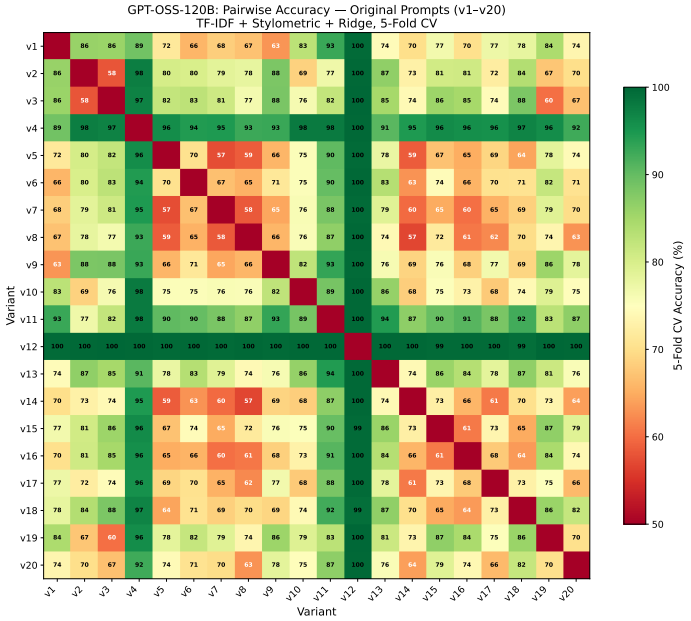


Figure 8: GPT-OSS-120B confusion matrix