

# Utterance-level Detection Framework for LLM-Involved Content Detection in Conversational Setting

Muyang Zhou, Huaxia Rui

University of Rochester

{mzhou36, huaxia.rui}@simon.rochester.edu

## Abstract

As Large Language Models (LLMs) increasingly power chatbots, social media, and other interactive platforms, the ability to detect AI in conversational settings is critical for ensuring transparency and preventing potential misuse. However, existing detection methods focus on static, document-level content, overlooking the dynamic nature of dialogues. To address this, we propose an utterance-level detection framework, which integrates features from individual and combined analysis of dialogue participants' responses to detect LLM-generated text under conversational setting. Leveraging a transformer-based recurrent architecture and a curated dataset of human-human, human-LLM, and LLM-LLM dialogues, this framework achieves an accuracy of 98.14% with high inference speed, supported by extensive results of experiments on different models and settings. This work provides an effective solution for detecting LLM-generated text in real-time conversations, promoting transparency, and mitigating risks of misuse.

## 1 Introduction

The research dedicated to detecting LLM generated text in written content is growing rapidly, while much of the focus remains on identifying text specific to LLMs in isolated documents or articles. However, there is a notable gap when it comes to detecting LLMs in real-time conversational contexts. Conversations present a unique challenge compared to document-level detection, where the structure and content are more static, due to their dynamic and interactive nature. The conversation provided for detection is often hybrid text with human-generated and LLM-generated text mixed. As LLMs increasingly power chatbots, customer support systems, and other interactive platforms, the ability to detect LLM in conversational settings is critical to ensuring transparency and preventing potential misuse. The main task is to design an

appropriate dataset and model for LLM-Involved Dialogue Detection. To address the challenge, we constructed a comprehensive dataset and developed a lightweight, efficient model tailored for LLM-Involved Dialogue Detection. The dataset is composed of three parts: human-human dialogues, LLM-LLM dialogues, and human-LLM dialogues. The dataset covers multiple conversational domains, and the composition is to balance and to support robust training and evaluation.

The proposed model is a recurrent structure leveraging an LLM tokenizer and a customized transformer architecture, combining self-attention and cross-attention mechanisms to process conversational context effectively. To enhance detection accuracy, we also introduced a detection focus augmentation strategy, which prioritizes the semantic features of the second speaker's response while retaining context from the first speaker's input. The model was trained on over 80,000 dialogues, with a focus on minimizing bias across different LLMs and ensuring real-time inference capabilities.

Experiments were conducted to evaluate the model's performance against various detection baseline methods. Our model demonstrated superior accuracy in detecting LLM-generated content across all tested datasets, including dialogues generated by different LLMs. Additionally, the model exhibited strong zero-shot generalization capabilities when tested on out-of-domain datasets.

In conclusion, our work addresses the critical need for LLM-Involved Dialogue Detection with the following key contributions:

- **A Comprehensive and Balanced Dataset:** We constructed a novel dataset providing a robust foundation for training and evaluating models in detecting LLM-generated content in dynamic, interactive settings.
- **An Efficient and Lightweight Model Structure:** We proposed a customized recurrent model ar-

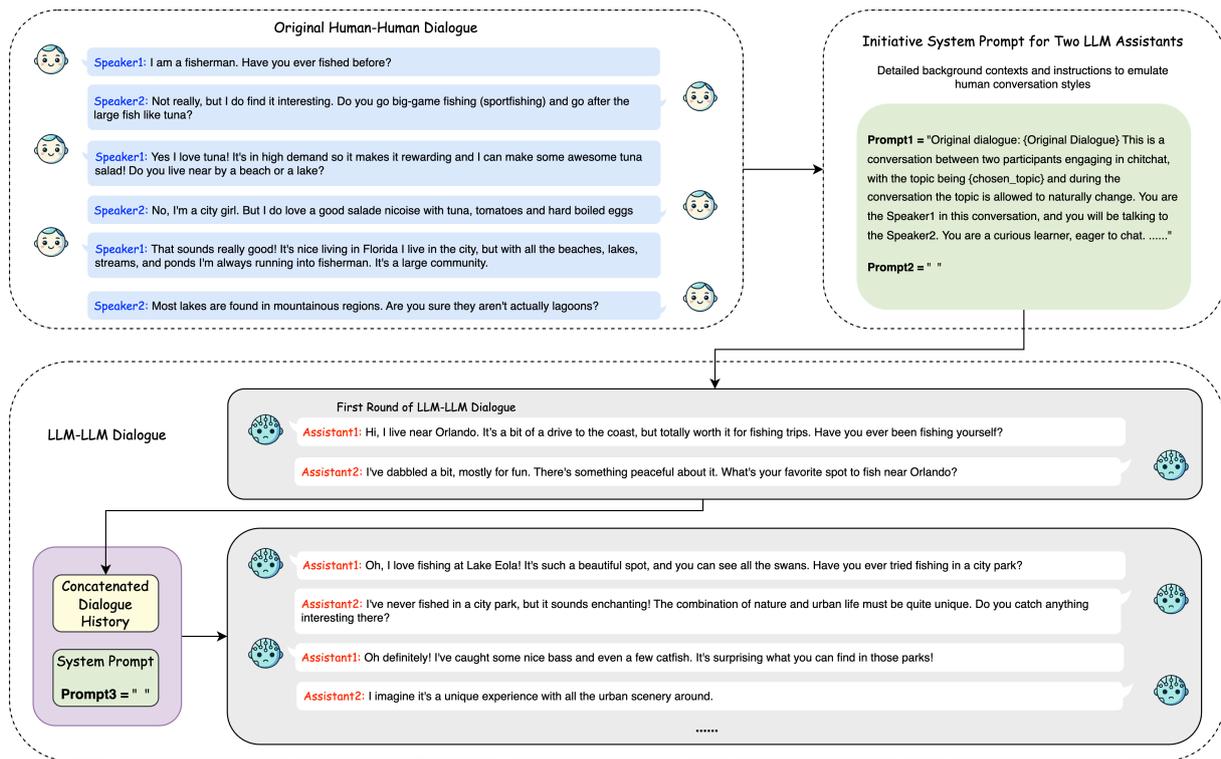


Figure 1: **Data Construction Pipeline for LLM-LLM Dialogues.** Large language models are used to understand and summarize original Human-Human dialogue, the results are concatenated into an initializing system prompt for two LLM assistants. Two LLM assistants begin to generate utterance alternatively.

chitecture which enables effective processing of conversational context while maintaining real-time inference capabilities.

- **Superior Performance in Dialogue Settings:** Our model outperforms existing models within conversational contexts in speed and accuracy, demonstrating its effectiveness in handling the unique challenges posed by hybrid human-LLM dialogues.
- **Strong Out-of-Domain Generalization:** The model exhibits remarkable zero-shot capabilities when tested on out-of-domain datasets, highlighting its robustness and adaptability to diverse conversational scenarios beyond its training scope.

## 2 Related Work

### Datasets for LLM-Involved Dialogue Detection.

Numerous datasets support LLM-generated content detection. HC3 (Guo et al., 2023) and HC3 Plus (Su et al., 2024) include human and ChatGPT responses for web text, QA, and semantic tasks, while CHEAT (Yu et al., 2024) targets academic content detection. OpenLLMText (Chen

et al., 2023) spans multiple LLMs in rephrasing tasks, and TweepFake (Fagni et al., 2021) and GROVER (Zellers et al., 2019) provide social media and news data. Other datasets, such as GPT2-Output and ArguGPT, focus on web and academic contexts. However, detection datasets still lack resources for everyday dialogue or short conversational exchanges. Multi-turn conversational datasets, such as Wizard of Wikipedia (Dinan et al., 2018) and Persona-Chat (Zhang et al., 2018), simulate structured dialogues, focusing on human-human interactions without mixed LLM-human exchanges. Developing a dataset with mixed human and LLM conversations across varied contexts is essential for advancing real-time multi-turn detection capabilities.

**Multi-Turn Dialogue Comprehension.** Effective GPT detection in conversations requires understanding evolving context across dialogue turns. Models like DialogBERT (Gu et al., 2021) and Hierarchical Transformers (Wang et al., 2022) capture multi-turn dependencies, enhancing consistency in responses. Memory-based models (Wang et al., 2020; Zhang et al., 2020), retain information from prior turns, aiding in differentiating human responses from LLM-generated content. Addition-

ally, DialogueRNN (Majumder et al., 2019) and DialogueGCN (Ghosal et al., 2019) track sentiment and intent, which often vary naturally in human conversations, making them useful for hybrid detection tasks. Incorporating multi-turn context tracking can help models recognize the subtle shifts that may indicate LLM involvement.

**Methods for LLM Detection.** Detection models for LLM-generated content typically fall into statistical and neural approaches. Statistical methods, such as DetectGPT (Mitchell et al., 2023) and Fast-DetectGPT (Bao et al., 2023), analyze structural and probabilistic differences but are less effective in short or multi-turn dialogues. Neural models, like SeqXGPT (Wang et al., 2023) and TriBERT (Zeng et al., 2023), enable sentence-level detection through token probabilities and segmentation. While promising, these models face challenges in dynamic conversations with brief responses. Integrating memory-augmented approaches in neural based models may improve detection accuracy by maintaining context continuity across dialogue turns.

## 3 Dataset

### 3.1 Overview

The dataset for this study supports conversation-level LLM generated text detection, featuring three main parts: human-human dialogues, LLM-LLM dialogues, and human-LLM dialogues. The conversation data in the first part are collected from existing datasets to capture diverse, realistic human-human exchanges. Human-LLM dialogues will then be generated based on these topics, and fully LLM-generated conversations will be constructed via prompts and programming.

Drawing on datasets for open-domain and goal-oriented dialogues, this collection will include over 80,000 dialogues for training. We resort to Wizard of Wikipedia (Dinan et al., 2018), PersonaChat (Zhang et al., 2018), and MuTual (Cui et al., 2020) for real world human conversations. The human-human dialogues and LLM-LLM dialogues make up most of the dataset and are used primarily for training, offering a large amount of diverse data for detection emphasizing on the second party of the dialogue. The third part, human-LLM dialogues, is used mainly for testing process to validate the model’s detection capability. Part of it also serves as supporting the training. All dialogues will be preprocessed and balanced to provide a ro-

bust resource for detecting LLM-generated content across multiple conversational domains.

### 3.2 Data Construction Pipeline

For human-human dialogues dataset, we leverage the three benchmark datasets mentioned above. For generating human-LLM dialogues, human participants engage with different LLM agents with randomly selected summarized topics from the human-human dialogue dataset, mimicing the conversation style. The human always initiates the dialogue, and LLM responds as the second participant. The number of turns in these dialogues matches the turn count of the first dataset, ensuring consistency across all generated data. The details are covered in Appendix A.2.

For LLM-LLM dialogues dataset, we generate multi-turn dialogues between two large language models by setting detailed background contexts and instructions for them to emulate human conversations. Each LLM assistant is provided with customized system prompts that include specific persona information (both their own and that of their conversational partner), scenario details, and conversation topic. Additionally, each assistant receives reference samples from the real-world human conversations, guiding them to adopt a similar conversational style and topic focus, and enabling generated dialogues to be directly comparable to corresponding human dialogues. The system prompts also include instructions of maintaining a conversational tone and concise response length.

The conversation begins immediately within the system prompts, with one assistant initiating the first utterance. The dialogue generation process in this pipeline operates iteratively on a turn-by-turn basis. At each turn, the active assistant—determined by alternating turns—receives a prompt constructed from both their system prompt and the previous utterances, providing context that guides the assistant’s next response to remain relevant and coherent. The active assistant then generates a new utterance informed by this complete prompt. This generated utterance is then appended to the dialogue history, updating the context for the next turn. This process repeats iteratively, with each assistant building on the previous turns, until the conversation reaches the desired number of turns, which is determined by the statistics gained from real-world human dialogue dataset.

In the presented pseudocode,  $A$  and  $B$  are the two assistants, initialized with system prompts

---

**Algorithm 1** Two-Assistant Dialogue Generation with Persona and Scenario Contexts

---

**Require:** Background Information; Target turns  $N$ ; Prompts  $\text{SYS}_A, \text{SYS}_B$ ; Model  $M$

**Ensure:** Dialogue  $D$  (list of utterances)

```
1: Initialize  $D \leftarrow []$ , assistants  $A$  and  $B$  with  $\text{SYS}_A, \text{SYS}_B, T \leftarrow 0$ 
2: function Select( $T$ )
3:   return  $A$  if  $T \% 2 = 0$  else  $B$ 
4: end function
5: while  $T < N$  do
6:   if  $T = 0$  then
7:      $u \leftarrow M.\text{gen\_init}(\text{SYS}_A)$ 
8:   else
9:      $C \leftarrow \text{Select}(T)$ 
10:     $\text{Hist} \leftarrow \text{concatenate}(D)$ 
11:     $\text{Prompt} \leftarrow \text{SYS}_C + \text{Hist}$ 
12:     $u \leftarrow M.\text{gen\_utt}(C, \text{Prompt})$ 
13:   end if
14:   Append  $u$  to  $D$ 
15:    $T \leftarrow T + 1$ 
16: end while
17: return  $D$ 
```

---

$\text{SYS}_A$  and  $\text{SYS}_B$ , respectively.  $T$  is the turn counter, which iterates up to the target number of turns  $N$ . The function  $\text{Select}(T)$  alternates between A and B, depending on whether  $T$  is even or odd. On each turn,  $u$  represents the current utterance generated by the model.  $\text{Hist}$  is the concatenated dialogue history from  $D$ , and  $\text{Prompt}$  is the combined system prompt and history given to the model.  $M.\text{gen\_init}$  generates the initial utterance, while  $M.\text{gen\_utt}$  generates subsequent utterances based on the current assistant and prompt.

Different large language models have their own priors and distributions. To broaden the dataset, diverse state-of-the-art models, such as GPT-4o (Achiam et al., 2023), LLaMA-3.1-70B-Instruct (Dubey et al., 2024) and Qwen2.5-72B-Instruct (Yang et al., 2024), are employed to generate both LLM-LLM and human-LLM dialogue data.

## 4 Methodology

### 4.1 Model Structure

The structure utilizes LLaMA3 tokenizer (Dubey et al., 2024) to tokenize all sentences as the input of the neural network model. Leveraging large language models (LLMs) to obtain text embeddings

offers several advantages over using relatively outdated encoder-only structure to understand text. Considering the need for a lightweight approach during inference, the LLaMA3 tokenizer and a custom 4-layer transformer is used together as feature extraction, which has the hidden dimension of  $d$ . The embeddings are then processed through two transformer blocks, one is fully self-attention and another contains cross-attention layer. The sentence modules are built into a structure similar to a recurrent structure to construct conversations model, where the input conversation is processed according to their classes (question or answering) and mixed by cross-attention. The recurrent structure will give an output by using a decoder in each stage as well as using the intermediate result as part of the input for the next recurrent. The effectiveness of recurrent structure is supported in A.4.2.

### 4.2 Detection Focus Augmentation

In this experiment, the goal is to train a model to distinguish whether the one responding in a conversation is human or a LLM. The semantic features of the second speaker’s responses are prioritized, as these contain stronger indicators of the speaker’s identity and significantly affect conversational dynamics. Although emphasis is on the second speaker’s responses, the initial utterances from the first speaker are also included since they provide essential conversational context. To ensure a balanced analysis, during training, a portion of the first speaker’s utterances will be randomly dropped, and their encoding weights reduced. A higher dropout rate is applied to dialogues fully generated by LLMs (LLM-LLM) compared to those involving human-generated context (Human-LLM), minimizing the influence of artificially generated context and enhancing model reliability. It helps the model focus more on the linguistic and semantic information present in the second speaker’s responses, which are more indicative of the detection purpose.

The detection focus augmentation (DFA) addresses a common failure case in dialogue settings. When assistant1’s utterance is significantly longer or richer than the assistant2’s, the detection for the brief, low-variance responses from assistant2 will benefit from our detection focus augmentation. As shown in Table 3, document-level models and our ablation variant (without detection focus augmentation) often struggle to detect subtle assistant-

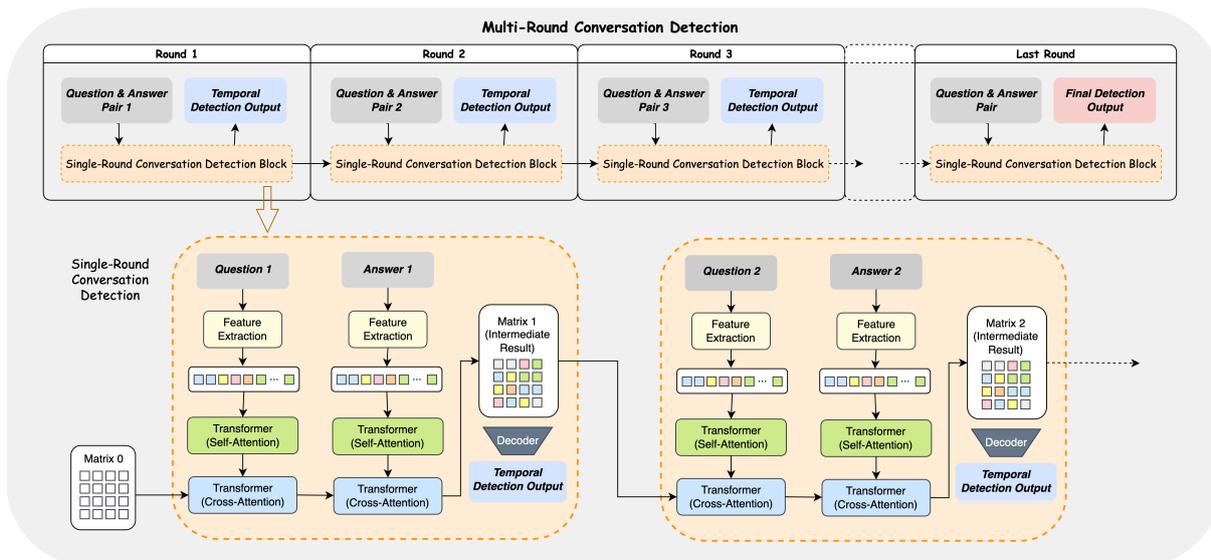


Figure 2: **Overview of the Multi-round Question and Answering Detection Network and the recurrent Block in Multi-round Question and Answering Detection Network.** The structure is a recurrent neural network. The network takes the whole conversation as input, where each round is fed to the single round recurrent block in turn. As the conversation is processed, there will be temporal detection outputs, and the final detection output will be obtained at the end. Each block processing the single round detection takes a matrix and a pair of question and answering as input, and the intermediate matrix for next round as well as a temporal detection result as output. Matrix<sub>0</sub> is an initial matrix. Questions and answers are first encoded by the feature extraction layer and then passed on to transformer blocks.

generated responses in such imbalanced cases.

## 5 Experiments

### 5.1 Experimental Setup

**Implementation details.** We train our model for 50 epochs on one NVIDIA A6000 GPU, employing a learning rate of  $1 \times 10^{-4}$  with the Adam optimizer. Detection focus augmentation method has been applied to all data containing LLM generated contents. Additionally, prompts are applied to GPT-4o (Achiam et al., 2023), LLaMA-3.1-70B-Instruct and Qwen2.5-72B-Instruct (Yang et al., 2024) to generate different LLM generated dialogues. LLM-LLM dialogues generated using LLaMA-3.1-70B-Instruct and Qwen2.5-72B-Instruct are executed on 2 NVIDIA H20 GPUs. All prior models used in the experiment is GPT-neo-2.7B.

**Baselines and metrics.** We adopt neural models, such as SeqXGPT (Wang et al., 2023) and TriBERT (Zeng et al., 2023), as our primary comparison. SeqXGPT employs a sequence labeling approach, containing perplexity extraction and alignment, for utterance-level GPT generated text detection, while TriBERT is a fine-tuned BERT-based method. Moreover, we evaluate statistics-based methods like DetectGPT (Mitchell et al., 2023) and Fast-DetectGPT (Bao et al., 2023), which are zero-

shot methods depending on LLM priors, for example, GPT-neo-2.7B and GPT-J-6B. Considering that DetectGPT and Fast-DetectGPT are primarily designed for document-level LLM-generated text detection tasks, which differ slightly from our task setting, we will implement both methods in two modes to ensure a fair comparison under the same benchmark. In the first mode, the entire conversation between the first speaker and the second speaker will be concatenated into a single paragraph as one input. This mode is defined as document level (D.). In the second mode, each response from the second speaker will be treated as a separate input for evaluation within our experimental setting. This mode is defined as utterance level (U.). We will consider classification accuracy and inference speed as evaluation metrics.

### 5.2 Quantitative Results

#### 5.2.1 Overall Performance

We tested the inference performance of our model and other models, and results are shown in Table 1. The inference process of our model is shown in Figure 3. For statistics-based models, we evaluate DetectGPT (Mitchell et al., 2023) and Fast-DetectGPT (Bao et al., 2023) at both document level (D.) and utterance level (U.). For neural-based models, we evaluate SeqXGPT (Wang et al.,

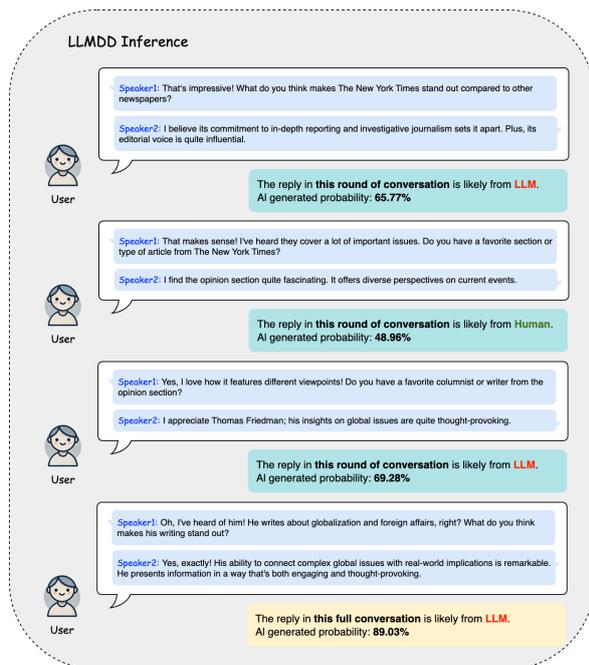


Figure 3: **Model Inference Process Demonstration.** The model gives the detection results and scores when the user input each round of dialogue. The final detection result and score are given after the whole dialogue is processed.

2023) and our model specifically at the utterance level. Considering our application setting, our model is specifically designed for real-time detection of LLM-generated text in dialogues. Therefore, greater emphasis should be placed on utterance-level results rather than document-level results, as the latter are only available after the entire conversation was concluded.

Our model outperforms DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2023), and SeqXGPT (Wang et al., 2023) across all human-LLM dialogues involving dialogues generated by GPT-4o (Achiam et al., 2023), LLaMA3.1-70B-Instruct (Dubey et al., 2024), and Qwen2.5-72B-Instruct (Yang et al., 2024). Since our model does not depend on a prior model, it exhibits minimal bias across dialogues generated by different LLMs. This ensures that our model's performance remains stable and consistent, with minimal fluctuation regardless of the LLM used to generate the text. Notably, even though our model is not trained on dialogue data generated by Gemini-2.0 or DeepSeek-v3, it still demonstrates strong performance on these datasets. This highlights the generalizability of our approach when applied to dialogues generated by previously unknown LLMs. However, all other baseline models require prior models, such

as GPT-neo-2.7B and GPT-J-6B. Based on these priors, the baseline models tend to perform better on GPT-4o-generated dialogues compared to dialogues generated by other LLMs. Compared with all other baseline models, our model only use LLM's tokenizer to tokenize text input, avoiding any bias caused by prior models.

As a real-time LLM-Involved Dialogue Detection algorithm, it is also crucial to evaluate models based on their inference speed and on-device capabilities to assess their practical value in real-world applications. When using GPU inference, our model is 34 times faster than Fast-DetectGPT (Bao et al., 2023) and 58 times faster than SeqXGPT (Wang et al., 2023). Moreover, our model requires minimal GPU memory, while all other baseline models have to load a prior model at first, making them not only incapable of fast inference on CPUs but also requiring significant GPU memory.

## 5.2.2 Out of Domain Performance

The results presented in Table 5 demonstrate that our model effectively performs document-level LLM-involved content detection. To evaluate the model's capabilities in this task, we employed the HC3 dataset (Guo et al., 2023). This dataset consists of a series of questions, each accompanied by long responses from human experts and ChatGPT and is not covered in our training data. These questions span five domains: finance, medicine, open Q&A, Reddit, and Wikipedia. To adapt this dataset for our model that is originally designed for conversation-level LLM-involved content detection, we preprocessed the data. All questions in the dataset were treated as the first-turn output of the conversation initiator (assistant 1). For subsequent turns, assistant 1 was left empty, and the long responses were segmented into multiple turns based on a predefined maximum length for assistant 2 outputs.

Our model exhibited strong zero-shot capabilities across these domains. Despite not being trained on data from these specific domains, it demonstrated the ability to discern whether the content was generated by an LLM, showcasing the model's generalization ability and robustness. These findings further validate the model's potential for cross-domain applications. This feature highlights the model's versatility in handling extended document-level tasks.

Method	Accuracy					Inference Speed	No Prior Model
	GPT-4o	LLaMA3.1	Qwen2.5	Gemini2.0	DeepSeek-v3		
Statistics-based Models							
DetectGPT (U.) (Mitchell et al., 2023)	51.33	59.37	53.73	52.94	57.82	1x	✗
Fast-DetectGPT (U.) (Bao et al., 2023)	52.31	81.01	73.57	78.19	72.46	12x	✗
DetectGPT (D.) (Mitchell et al., 2023)	53.73	65.91	53.52	54.88	60.17	1.14x	✗
Fast-DetectGPT (D.) (Bao et al., 2023)	76.00	97.86	98.27	96.32	94.83	48x	✗
Neural-based Models							
SeqXGPT (U.) (Wang et al., 2023)	98.11	91.73	90.51	92.31	90.88	28x	✗
<b>Our</b>	<b>98.14</b>	<b>98.59</b>	<b>98.73</b>	<b>95.89</b>	<b>96.77</b>	<b>1630x</b>	✓

Table 1: We leverage GPT-4o (Achiam et al., 2023), LLaMA3.1-70B-Instruct (Dubey et al., 2024), Qwen2.5-72B-Instruct (Yang et al., 2024), Gemini-2.0-Flash (Google, 2025), and DeepSeek-v3 (Liu et al., 2024) to generate human-LLM and LLM-LLM dialogues in the same conversational semantics as the test set. We evaluate DetectGPT (Mitchell et al., 2023) and Fast-DetectGPT (Bao et al., 2023) at both document level (D.) and utterance level (U.), while SeqXGPT (Wang et al., 2023) and our model are assessed specifically at the utterance level. Our model is trained on the dataset generated based on GPT-4o, LLaMA3.1-70 Instruct, Qwen2.5-72B-Instruct. Our model shows strong generalizability on unknown LLM generated content, such as Gemini-2.0-Flash and DeepSeek-v3. The test dataset generated by these LLMs follows the same data construction pipeline. Inference Speed of different models under various settings are also presented by setting DetectGPT(U.) as the benchmark for comparison.

Method	Accuracy					Inference Speed	Memory Usage
	GPT-4o	LLaMA3.1	Qwen2.5	Gemini2.0	DeepSeek-v3		
General Tokenizer	89.21	90.33	90.58	89.74	88.37	143x	0.15GB
LLM hidden state	98.52	98.95	99.04	96.32	96.90	1x	15.29GB
LLM Tokenizer	98.14	98.59	98.73	95.89	96.77	124x	0.15GB

Table 2: **Comparison of Three Feature Extraction Methods (general tokenizer, LLM tokenizer, and LLM hidden states).** The comparison is across GPT-4o, LLaMA3.1, Qwen2.5, Gemini2.0, and DeepSeek-v3, and metrics include accuracy, F1 score, inference speed, and memory usage. Results show that general tokenizers are efficient but less accurate, while LLM hidden states achieve the highest accuracy at the cost of significant memory and speed. LLM tokenizers provide a balance, offering near-optimal accuracy with minimal resource requirements.

Average Utterance Length of Assistant2	Our Model w/o DFA	Our Model w/ DFA	Fast-DetectGPT
0.5362 × full utterance length (Entire dataset)	93.27	98.23	76.00
0.60 × full utterance length (Subset of 500 conversations)	91.80	98.40	74.40
0.70 × full utterance length (Subset of 500 conversations)	85.60	98.80	71.40

Table 3: **Performance of Our Model and Fast-DetectGPT on GPT-4o Datasets.** This table compares detection accuracy with and without Detection Focus Augmentation (DFA). The assistant2’s utterance coverage is used to control for the brevity of the responses in detections. Our model consistently outperforms Fast-DetectGPT, especially when DFA is applied.

Method	Number of Conversation Rounds									
	1	2	3	4	5	6	7	8	9	10
Round Specific Training	88.30	94.48	96.26	98.09	98.23	98.30	98.22	98.24	98.13	98.17
5 Rounds Training	85.65	93.28	95.79	98.25	98.23	98.21	98.07	98.14	97.69	97.83

Table 4: **Performance Comparison of 5-Round Training Model and Round-Specific Models.** This table presents the performance of models trained on dialogue datasets with specific numbers of conversation rounds (1 to 10) compared to a model trained on 5-round dialogues, evaluated on GPT-4o generated data. Metrics indicate that round-specific training achieves slightly better performance for the exact number of rounds it is optimized for, especially for fewer rounds (e.g., 1 to 4). However, the 5-round model shows strong adaptability, delivering comparable performance across all scenarios, particularly in handling dialogues longer than 5 rounds.

## 6 Ablation Study

Method	Domains				
	Finance	Medicine	Open Q&A	Reddit	Wiki
Our	78.17	91.67	92.83	82.83	78.06

Table 5: **Performance of the model on document-level LLM-involved content detection across five out-of-domain datasets.** The results highlight the model’s zero-shot generalization capabilities and its ability to detect LLM-generated content effectively in unseen domains.

### 6.1 Feature Extraction with Different Tokenizers

For the feature extraction component of the model, using only a traditional tokenizer to obtain sentence embeddings achieves a limited accuracy of 89%. Significant improvements can be made by integrating parts of a large language model (LLM) into the process. Leveraging LLMs for text embeddings offers several advantages over the relatively outdated encoder-only structures. LLMs, trained on massive datasets with billions of parameters, have a deeper contextual understanding, enabling them to capture complex semantic relationships and long-range dependencies in text. Moreover, they exhibit superior generalization capabilities, understanding recent language trends, diverse oral expressions, and providing more expressive embeddings that capture both subtle semantic differences and intricate linguistic features. To evaluate our model’s performance and efficiency, we compare three encoding methods: using a traditional tokenizer, using the LLaMA3 tokenizer, and using the LLaMA3 encoder.

We implement the feature extraction part of LLaMA3 8B (Dubey et al., 2024) by utilizing its tokenizer and hidden states from 32 transformer layers as new input for the model. The process involves loading the LLaMA3 8B model and applying its tokenizer to convert input text into token IDs. These tokens are then passed through LLaMA3 8B to generate hidden states representing the embeddings for each token in the sequence, with dimensions  $N \times d$ , where  $N$  denotes the number of tokens in the input text and  $d$  is the model’s embedding dimension. This approach significantly enhances the model’s performance, however, it has notable drawbacks in computational requirements. Training with the full 32-layer LLaMA3 model

demands over 32 GB of memory, making it computationally expensive and impractical for certain applications. Such high memory consumption restricts deployment to high-resource environments and limits real-time processing capabilities. Furthermore, the increased computational overhead results in slower processing times, which may be unsuitable for applications requiring rapid responses. While the accuracy improvements are impressive, the high resource demands hinder the practicality of deploying full-scale LLMs in everyday tasks or low-resource settings. This trade-off is evident in Table 2.

### 6.2 Conversations with Different Number of Rounds

Our model is trained on dialogue data consisting of 5 rounds of interactions. During the inference phase, different strategies are employed to handle variations in the number of dialogue rounds. For inputs with fewer than 5 rounds, we pad the dialogue with blank content to reach 5 rounds. For inputs exceeding 5 rounds, a sliding window approach is used: dialogues are processed in 5-round segments, and the results are aggregated using a voting mechanism to produce the final output. It is important to note that the choice of 5 rounds was not arbitrary but based on a comprehensive evaluation of experimental results.

To validate the performance of the 5-round model, we conducted comparative experiments against models trained on datasets ranging from 1 to 10 rounds of dialogue and the results are shown in Table 4. In scenarios with fewer than 5 rounds, the 5-round model performed slightly worse than models specifically optimized for shorter dialogue lengths. However, the difference was not significant, as shown in Table 10. For dialogues exceeding 5 rounds, the 5-round model’s performance was comparable to that of models optimized for longer dialogues. These results demonstrate that the padding strategy effectively compensates for insufficient dialogue rounds when processing shorter conversations. Similarly, the sliding window method successfully captures contextual information in longer dialogues, enabling the 5-round model to maintain robust performance. The model trained on 5-round dialogue data exhibits strong adaptability. When combined with the padding and sliding window strategies, it achieves satisfactory performance across varying dialogue lengths. This indicates that our approach has high generalization

capabilities and practical value for multi-turn dialogue tasks. Even if the model is trained for a specific number of rounds, the performance will not be significantly affected.

## 7 Conclusion

In this paper, we propose a novel framework for LLM-Involved Dialogue Detection that addresses the challenges of detecting LLM-generated text in real-time conversations. Our approach leverages a robust dataset of multi-turn hybrid dialogues to capture the nuanced dynamics of conversational settings. The lightweight, customized transformer-based architecture, enhanced by the LLaMA3 tokenizer and detection focus augmentation, achieves state-of-the-art performance with 98.14% accuracy on human-LLM dialogues and faster inference. Ablation studies highlight the importance of using LLM tokenizer, which improves accuracy by 9% over traditional tokenizers, and the effectiveness of adaptive strategies like padding and sliding windows for handling dialogues of varying lengths. The model also demonstrates strong generalization, achieving over 90% accuracy in some of the out-of-domain tasks. These results underscore the framework's practical value for real-time deployment in conversational AI systems, ensuring transparency and accountability in LLM-powered platforms.

## Limitation

Our framework has some limitations. First, the reliance on the LLaMA3 tokenizer, while improving accuracy, introduces higher computational demands, which limits the model's deployment in low-resource environments or on devices with constrained computational capabilities. Second, the current framework may struggle with highly specialized contexts, such as medical or legal discussions, where domain-specific terminology and conversational patterns differ significantly from general dialogue. Third, the dataset, though comprehensive, is limited to single-turn and multi-turn dialogues between two participants, leaving multi-party or group interactions unexplored. Finally, the model's performance, while robust, could benefit from further optimization to reduce inference latency and memory footprint for real-time applications.

Future research directions can address these limitations and expand the framework's applicability. One promising avenue is cross-lingual detection,

enabling the model to handle multilingual conversations, which are increasingly common in global applications. Another direction is multi-party dialogue detection, extending the framework to support group interactions, where multiple speakers and overlapping conversations introduce additional complexity. Incorporating multimodal capabilities to process textual, visual, and auditory data could further enhance the model's ability to detect LLM-generated content in richer, more diverse conversational settings. Additionally, domain-specific customization could improve performance in specialized fields like healthcare, finance, or legal systems, where accuracy and reliability are critical. Finally, optimizing the framework for real-time on-device detection would broaden its practical utility, enabling deployment in resource-constrained environments such as mobile devices or edge computing systems. Addressing these challenges will further advance the transparency, trust, and accountability of LLM-powered interactive platforms.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.
- Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. [Token prediction as implicit classification to identify llm-generated text](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 13112–13120. Association for Computational Linguistics.
- Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. [MuTual: A dataset for multi-turn dialogue reasoning](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1406–1416, Online. Association for Computational Linguistics.
- Emily Dinan et al. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *Proceedings of ICLR*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

- Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.
- Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. [Dialoguegen: A graph convolutional neural network for emotion recognition in conversation](#). *Preprint*, arXiv:1908.11540.
- Google. 2025. Gemini 2.0 flash. <https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/#gemini-2-0-flash>.
- Xiaodong Gu, Kang Min Yoo, and Jung-Woo Ha. 2021. Dialogbert: Discourse-aware response generation via learning to recover and rank utterances. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12911–12919.
- Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. [How close is chatgpt to human experts? comparison corpus, evaluation, and detection](#). *Preprint*, arXiv:2301.07597.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. 2019. Dialoguernn: An attentive rnn for emotion detection in conversations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6818–6825.
- Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. [Detectgpt: Zero-shot machine-generated text detection using probability curvature](#). *Preprint*, arXiv:2301.11305.
- Zhenpeng Su, Xing Wu, Wei Zhou, Guangyuan Ma, and Songlin Hu. 2024. [Hc3 plus: A semantic-invariant human chatgpt comparison corpus](#). *Preprint*, arXiv:2309.02731.
- Jiamin Wang, Xiao Sun, Qian Chen, and Meng Wang. 2022. Information-enhanced hierarchical self-attention network for multiturn dialog generation. *IEEE Transactions on Computational Social Systems*, 10(5):2686–2697.
- Jian Wang, Junhao Liu, Wei Bi, Xiaojiang Liu, Kejing He, Ruifeng Xu, and Min Yang. 2020. Dual dynamic memory network for end-to-end multi-turn task-oriented dialog systems. In *Proceedings of the 28th international conference on computational linguistics*, pages 4100–4110.
- Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. Seqxgpt: Sentence-level ai-generated text detection. *arXiv preprint arXiv:2310.08903*.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.
- Peipeng Yu, Jiahao Chen, Xuan Feng, and Zhihua Xia. 2024. [Cheat: A large-scale dataset for detecting chatgpt-written abstracts](#). *Preprint*, arXiv:2304.12008.
- Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.
- Zijie Zeng, Lele Sha, Yuheng Li, Kaixun Yang, Dragan Gašević, and Guanliang Chen. 2023. [Towards automatic boundary detection for human-ai collaborative hybrid essay in education](#). *Preprint*, arXiv:2307.12267.
- Bowen Zhang, Xiaofei Xu, Xutao Li, Yunming Ye, Xiaojun Chen, and Zhongjie Wang. 2020. A memory network based end-to-end personalized task-oriented dialogue generation. *Knowledge-Based Systems*, 207:106398.
- Saizheng Zhang et al. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of ACL*.



contexts. Wizard of Wikipedia dataset contains knowledge-grounded dialogues covering a wide range of factual topics. Persona-Chat dataset contains persona-based conversations where each speaker maintains predefined personal characteristics. And the third one MuTual contains multi-turn dialogues derived from English learning scenarios, designed for dialogue understanding.

#### **A.2.1.1 Topics and Persona Information**

##### **Wizard of Wikipedia**

The dataset is designed for knowledge-grounded, open-domain dialogue. It simulates conversations between a curious apprentice and a knowledgeable wizard, aiming to model in-depth topic exploration within natural multi-turn interactions. In each conversation, one participant plays the wizard, who has access to retrieved Wikipedia passages, and the other plays the apprentice, engaging in open-ended dialogue to explore a chosen topic. The conversation begins with a selected topic (e.g., “Gouda cheese”, “commuting”, “podcasts”) from the apprentice, and evolves naturally with the wizard incorporating retrieved knowledge.

This structure emphasizes both engagement and depth of knowledge, distinguishing it from shallow chit-chat datasets. A total of 1,365 open-domain topics were crowd-sourced and linked to Wikipedia articles. The topics span domains such as daily life, food, entertainment and celebrities. These structured topics and passages form the topic source used later in constructing LLM prompts.

##### **Persona-Chat**

This dataset was created to support more engaging, consistent, and personal chit-chat dialogue by conditioning conversations on personas. The dataset was collected via Amazon Mechanical Turk (MTurk) in a three-stage process: persona creation, persona revision, and persona-grounded dialogue collection. The original personas in the dataset contains 1,155 crowd-sourced and fictional personas each consisting of at least 5 profile sentences. (e.g., "I like swimming. I'm a vegetarian. I got a new job in advertising.") And the personas are revised to reduce lexical overlap bias, each original persona was rephrased or semantically generalized by a different set of workers. Then two MTurk workers were paired and each assigned a unique persona, and neither saw the other's persona; they had to learn about it through dialogue. Workers were incentivized to create natural, engaging, and reciprocal conversations. The chat was constrained

to a maximum of 15 words per utterance, with each dialogue containing 6–8 turns per participant on average.

Conversations in Persona-Chat reflect bidirectional persona grounding where participants discuss their assigned profile naturally. The common themes are family, jobs, hobbies, preferences, and personality traits. There may also involve active persona discovery where participants infer each other's persona via questioning and answering. We incorporate Persona-Chat data as a foundation for modeling persona-consistent response generation. And this inspired our prompting framework, where language models are assigned roles and persona traits. This also enables fine-tuning or evaluation for tasks involving speaker grounding, identity, and personal style.

##### **MuTual**

The dataset was originally constructed to evaluate machine comprehension and reasoning in multi-turn dialogue. It was adapted from English listening comprehension exams for non-native speakers, where students were required to understand a short dialogue and choose the most appropriate response from four options. While designed for response selection, the dialogues themselves reflect natural human-human interactions in daily life. Original materials were sourced from publicly available English listening comprehension exams, and dialogues are based on realistic daily scenarios such as buying tickets, attending concerts, or chatting in restaurants.

There are 6,371 original dialogues, and the reasoning categories contains attitude (13%), intention prediction (31%), multi-fact (24%), situation (16%), algebraic (7%), and others (9%). The dialogues in MuTual have multi-turn structure that mimics real-world conversations. There also exists frequent use of implicit knowledge and pragmatic cues such as speaker intentions, relationships, and attitudes.

We repurpose MuTual solely for its dialogue content, omitting the multiple-choice question-answer structure. We extract only the raw dialogues to serve as representative examples of human-human chit-chat in realistic settings. These dialogues are treated as individual conversational segments, showcasing spontaneous and contextually grounded exchanges. Since the original dialogues were designed for comprehension and reasoning, they inherently include diverse linguistic features, speaker intentions, and pragmatic infer-

ence that are all desirable for dialogue modeling.

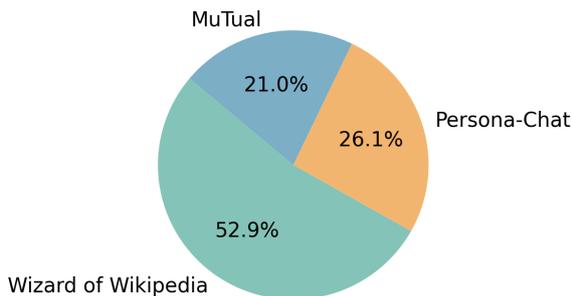


Figure 8: Conversation Distribution Among Different Source Datasets

Method	GPT-4o	LLaMA3.1	Qwen2.5	Gemini2.0	DeepSeek-v3
LLM	5,500	5,500	5,500	5,000	5,000
Human	5,000	5,000	5,000	5,000	5,000
Total	10,500	10,500	10,500	10,000	10,000

Table 6: Distribution of data samples by class (LLM vs. Human) across different models in the testing set.

### A.2.1.2 Statistics

The total conversations of Wizard of Wikipedia is 22,311, the total conversations of Persona-Chat is 10,981, and the total conversation of MuTual is 8,860. The conversation distribution among different dataset is shown in figure 8. The average turns of Wizard of Wikipedia is around 9, the average turns of Persona-Chat is around 8, and the average turns of MuTual is around 5. The average length of all utterances is 19.57. The overall topics across datasets with categories are entertainment (22%), science and technology (18%), personal life (25%), education (15%), and miscellaneous (20%).

### A.2.2 Human-LLM Dataset

We recruited 20 fluent English speakers (age 18–27), including undergraduate and graduate students and crowd-sourced researchers with prior exposure to dialogue systems. Participants were briefed on the purpose but blinded to which model they interacted with. The participants first read 3–5 human-human dialogues within one topic randomly selected from the five categories of the Human-Human dataset and wrote concise 1–2 sentence summaries about the dialogue context. They were then given the instructions of: “Start a conversation with an AI agent. Use the topic summary as back-

ground, adopt a natural and engaging style, and vary your tone across conversations.”

The human always initiates the conversation. LLM responds, turn by turn, for a predefined number of turns. In the end, the dialogues were manually reviewed by two annotators given the criteria of naturalness, relevance, and coherence. Low-quality dialogues (<5%) were discarded.

For model assignment, each topic was used across all models (GPT-4o, LLaMA-3.1-70B-Instruct, Qwen2.5-72B-Instruct) to ensure uniformity and ensure topic-level performance comparison between models.

### A.2.3 LLM-LLM Dataset

For each topic, system prompts include assistant persona that might contain background and speaking style; scenarios such as social situation, role, or knowledge domain; and topic focus. Each model receives these prompts with adjusted format to match their tokenization and instruction-following behavior.

We fine-tune prompt length, temperature, and sampling parameters for each model to avoid mode collapse, preserve natural variability and stay consistent with human-human dialogue metrics.

#### Example prompt for WoW dataset:

**prompt1** = {human-human dialogues.} This is a conversation between two participants engaging in chitchat, with the topic being {chosen topic} and during the conversation the topic is allowed to naturally change. You are the initiator in this conversation, and you will be talking to another person. You are a curious learner, eager to chat, and the one you will be talking to is a person who knows plenty of knowledge. Your goal is to go into depth about the topic mentioned above while keeping the conversation engaging and fun. Do not generate anything other than your responses. Please start the conversation. Talk in a similar style as the conversation provided.

**prompt2** = {human-human dialogues.} This is a conversation between two participants engaging in chitchat, with the topic being {chosen topic} and during the conversation the topic is allowed to naturally change. You are the second party in this conversation, and you will be talking to another person. You know plenty of knowledge. You have just met the other person, who seems quite curious, and you are eager to discuss a topic with them. Do not generate anything other than your responses. Talk in a similar style as the conversation provided.

### A.3 Examples

Here are some examples from our constructed dataset, and detection is performed by Fast-DetectGPT. These examples of success and failure cases help understand why detecting LLM-generated text for dialogue is particularly challenging for statistical-based models.

#### Example One

"conversation":

"assistant1": "So, if Vitamin C is so important, how much do we actually need each day?"

"assistant2": "For most adults, about 65 to 90 milligrams a day is recommended. But it can vary based on age and lifestyle."

"assistant1": "Oh, interesting! So if I'm more active, do I need more vitamin C?"

"assistant2": "Yes, indeed! Active individuals may need a bit more to support recovery and immune function."

"assistant1": "That makes sense! So, if someone is really active, should they focus more on certain foods to boost their intake?"

"assistant2": "Absolutely! Foods like kiwi, guava, and bell peppers can be great for active individuals. They pack a punch of vitamin C and other nutrients too."

"assistant1": "I've heard kiwi is super nutritious! Do you think it tastes better than oranges?"

"assistant2": "Kiwi is indeed very nutritious! It has a unique, sweet-tart flavor that many find delightful. It's a nice change from oranges, but it all comes down to personal preference!"

"label": "gpt4o"

**Fast-DetectGPT: 36% AI Generated**

#### Example Two

"conversation":

"assistant1": "I wonder, why do you think people are drawn to certain colors like pink or blue?"

"assistant2": "Colors can evoke emotions and memories! Pink often brings feelings of warmth, while blue can inspire calmness."

"assistant1": "That's so true! I feel like pink makes me think of happy moments, like cotton candy at fairs!"

"assistant2": "Absolutely! Colors can evoke memories and feelings. Pink often brings warmth and nostalgia!"

"assistant1": "That's so true! I feel like pink reminds me of cotton candy and childhood fairs! What memories does blue bring to you?"

"assistant2": "Blue brings me memories of calm lakes and clear skies. It feels like serenity and freedom!"

"label": "gpt4o"

**Fast-DetectGPT: 14% AI Generated**

#### Example Three

"conversation":

"assistant1": "Have you ever been to iceland? I have never been, but I would like to go sometime!"

"assistant2": "No but I would love to go! I want to see the sand and lava fields as well as the glaciers."

"assistant1": "That is such a varied place, how big is the country? it must have everything there, so many different weather phenomenons"

"assistant2": "Only the capital, Reykjavik and surrounding areas are really populated. It seems like you'd have to visit there for all your traveling needs."

"assistant1": "Do you know what their total population is? I may have to !"

"assistant2": "No, I am not sure what the population is but I know it's the most sparsely populated country in Europe."

"assistant1": "Oh wow that is interesting, maybe it has something to do with all that crazy weather then!"

"assistant2": "I'm sure that plays a part in it as well as the fact that Iceland is volcanically and geologically active still."

"assistant1": "Yeah that is enough to deter me from living there lol, I would not want to get fried!"

"assistant2": "I agree, also the volcanoes make up the landscape so trekking through the interior of the country would be difficult."

"label": "human"

**Fast-DetectGPT: 76% AI Generated**

#### Example Four

"conversation":

"assistant1": "I love the book so much that even as a non christian it makes alot of sense"

"assistant2": "I agree, just all the sacred texts and scriptures, even if it is old, I still see it as a product of God or a divine being"

"assistant1": "Is there a record as to where the original scripts that was used came from?"

"assistant2": "There is not really a concrete date just because many different authors contributed to it, they are not sure"

"assistant1": "Ok, is it there that there is a relationship between the first few books of the bible and the quaran?"

" assistant2": "Yes, they are practically the same, which has caused some weird attitudes from the christian groups, as that is a tough pill to swallow in a way"

" assistant1": "What is the best version of the bible?"

" assistant2": "A lot of people prefer the king james version of the bible, which has 39 books of the old testament and 27 of the new testament"

"label": "human"

**Fast-DetectGPT: 81% AI Generated**

## A.4 More Quantitative Results

### A.4.1 Overall Performance

Figure 9 highlights a clear trend that newer neural-based methods dramatically improve both accuracy and inference speed, surpassing traditional perturbation-based methods in conversational level LLM-generated content detection.

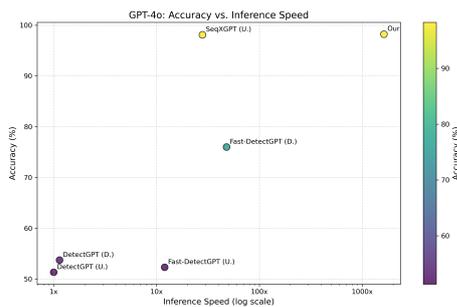


Figure 9: **Comparison of different models on GPT-4o generated content Accuracy vs. Inference Speed.** This figure illustrates the relationship between accuracy (%) and inference speed (log scale) for various AI-generated text detection methods applied to GPT-4o generated content.

Performance comparison between the baseline models and our proposed method in terms of Recall and Precision across five representative LLMs is presented in Table 8.

Configuration	Accuracy
without recurrence	86.79
with recurrence	98.23

Table 7: Accuracy of the Model With or Without Recurrence on GPT-4o Dataset.

### A.4.2 Recurrence Structure

Recurrence provides critical benefits for generalization and efficiency across varied-length dialogues, as shown in table 7. However, the simpler structure without recurrence can be considered as a document-level model, since it cannot process the conversation cross utterances.

### A.4.3 Models Trained on Different Conversation Rounds

The performance of Round Specific Training models represents results from ten separate models, each trained specifically for a fixed number of dialogue rounds(from 1 to 10). In contrast, the results of 5 Rounds Training model shows the performance of a single model trained only on 5-turn conversations and then evaluated across all turn lengths. The results are shown in figure 10.

This highlights a key strength of our recurrent architecture that it can generalize well across variable-length dialogues without needing to train and store multiple models. Notably, the 5-round model achieves competitive performance even on dialogues of varying lengths (especially >5), showcasing strong generalization and significant efficiency benefits in both training and deployment.

Method	GPT-4o	LLaMA3.1	Qwen2.5	Gemini2.0	DeepSeek-v3
DetectGPT (U.)	48.91 / 53.90	51.80 / 63.82	52.65 / 56.23	47.94 / 53.27	54.86 / 58.31
Fast-DetectGPT (U.)	49.09 / 55.02	76.35 / 85.83	66.31 / 79.82	76.04 / 79.46	66.18 / 75.69
DetectGPT (D.)	49.07 / 56.75	61.71 / 69.73	49.91 / 56.37	51.92 / 55.19	57.22 / 60.81
Fast-DetectGPT (D.)	70.89 / 80.92	96.29 / 99.61	96.71 / <b>99.98</b>	<b>93.22</b> / 99.38	90.28 / 99.32
SeqXGPT (U.)	96.58 / 99.81	84.95 / 99.15	82.80 / 98.91	85.32 / 99.19	82.66 / 98.92
<b>Our</b>	<b>96.62 / 99.83</b>	<b>97.44 / 99.87</b>	<b>97.69</b> / 99.89	92.22 / <b>99.53</b>	<b>93.86 / 99.66</b>

Table 8: Performance comparison (Recall / Precision) of baselines and our proposed method on five representative LLMs. Bold values denote the best results.

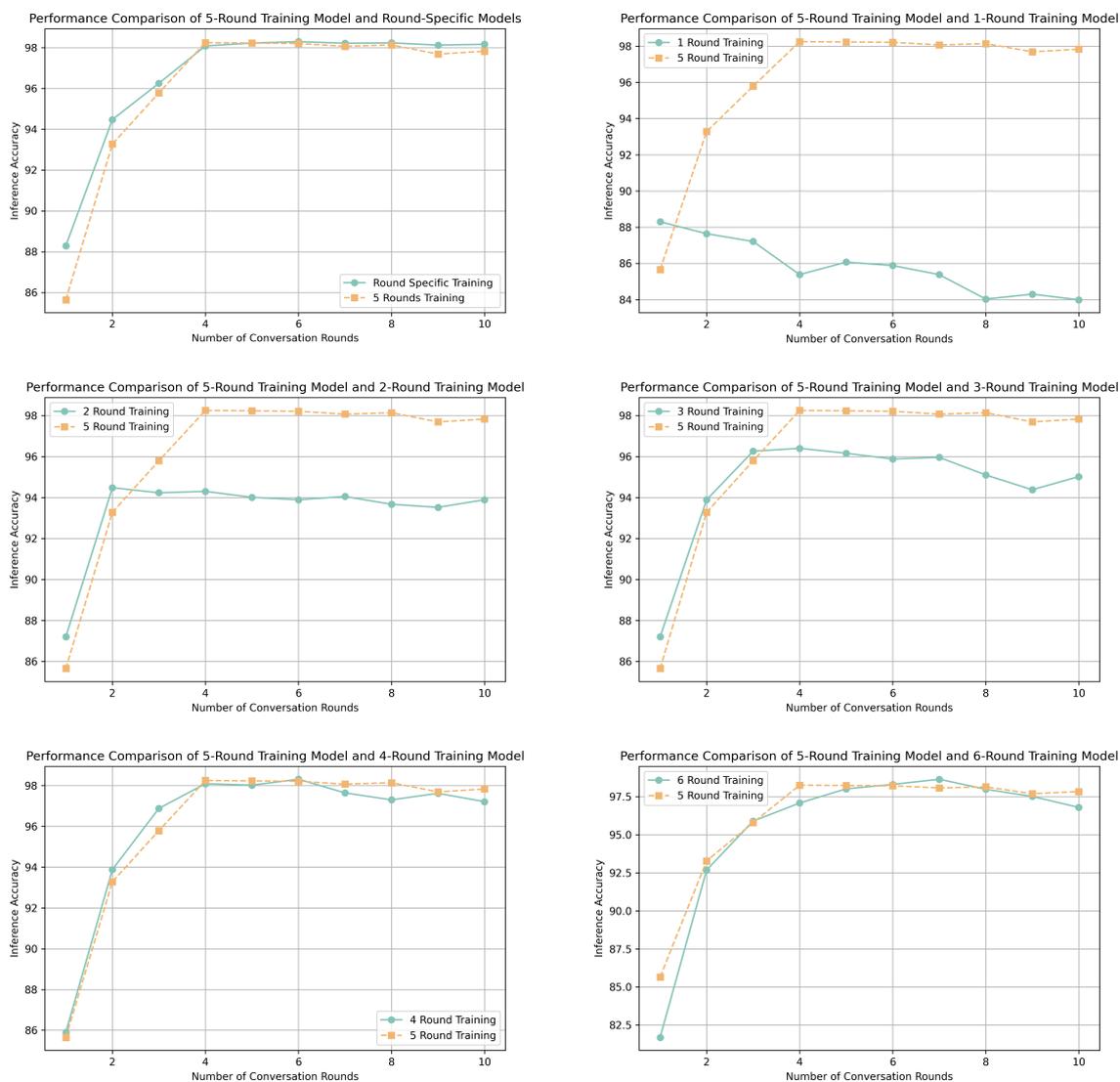


Figure 10: **Performance Comparison of 5-Round Training Model and Round-Specific Models and Performance Comparison of 5-Round Training Model and 1,2,3,4,6-Round Training Models.** (a) presents the performance of models trained on dialogue datasets with specific numbers of conversation rounds (1 to 10) compared to a model trained on 5-round dialogues, evaluated on GPT-4o generated data. (b) presents the performance of models trained on 1 conversation round dialogue datasets compared to a model trained on 5-round dialogues, evaluated on GPT-4o generated data. (c) presents the performance of models trained on 2 conversation round dialogue datasets compared to a model trained on 5-round dialogues, evaluated on GPT-4o generated data. (d) presents the performance of models trained on 3 conversation round dialogue datasets compared to a model trained on 5-round dialogues, evaluated on GPT-4o generated data. (e) presents the performance of models trained on 4 conversation round dialogue datasets compared to a model trained on 5-round dialogues, evaluated on GPT-4o generated data. (f) presents the performance of models trained on 6 conversation round dialogue datasets compared to a model trained on 5-round dialogues, evaluated on GPT-4o generated data.