

Speculative End-Turn Detector for Efficient Speech Chatbot Assistant

Hyunjong Ok^{1,2} Suho Yoo^{2,3} Jaeho Lee¹

¹POSTECH ²HJ AILAB ³KAIST

{hyunjong.ok, uso7d0}@gmail.com, jaeho.lee@postech.ac.kr

Abstract

Spoken dialogue systems powered by large language models have demonstrated remarkable abilities in understanding human speech and generating appropriate spoken responses. However, these systems struggle with end-turn detection (ETD)—the ability to distinguish between user turn completion and hesitation. This limitation often leads to premature or delayed responses, disrupting the flow of spoken conversations. In this paper, we introduce the OpenETD Dataset, the first public dataset for end-turn detection. The OpenETD dataset consists of both synthetic speech data generated with text-to-speech models and real-world speech data collected from web sources. We also propose SpeculativeETD, a novel collaborative inference framework that balances efficiency and accuracy to improve real-time ETD in resource-constrained environments. Our approach jointly employs a lightweight GRU-based model, which rapidly detects the non-speaking units in real-time on local devices, and a high-performance Wav2vec-based model running on the server to make a more challenging classification of distinguishing turn ends from mere pauses. Experiments demonstrate that the proposed SpeculativeETD significantly improves ETD accuracy while keeping the required computations low.

1 Introduction

Recent advancements in large language models (LLMs) have spurred extensive research into various LLM-based agents (Song et al., 2023; Shao et al., 2024; Zhang et al., 2024; Xie et al., 2024). Many studies focus on integrating LLMs with spoken dialogue systems to enable more engaging human-computer interaction (Mitsui et al., 2023; Yan et al., 2024; Ma et al., 2024; Veluri et al., 2024). However, LLM-based spoken dialogue systems face a critical challenge: when the user suddenly stops speaking, the systems often fail to discern

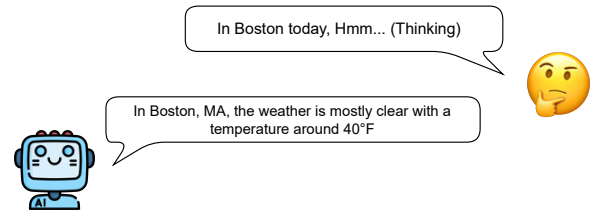


Figure 1: **End-turn detection failure example.** An example of a failure case of end-turn detection during a voice chat with GPT-4o. When the user pauses to think, the system incorrectly interprets this as the end of their turn.

whether the user has completed their turn or is merely pausing to think during a conversation (Lin et al., 2022; Umair et al., 2024). For instance, as illustrated in Figure 1, when a user momentarily pauses while contemplating a specific question, the LLM may erroneously interpret this pause as the end of the turn, leading to premature and off-target responses. This task, called end-turn detection (ETD), is a challenging task that requires nuanced mechanisms based on a firm understanding of the content of the speech; naïve cues, such as the pause duration, fails to provide meaningful information to distinguish between the turn completion and user hesitation (Ten Bosch et al., 2005).

Recent research has explored techniques to improve turn-taking in dialogue systems to address this issue (Chang et al., 2022; Veluri et al., 2024). However, a major bottleneck is the lack of publicly available datasets. Existing datasets, such as the Fisher corpus (Cieri et al., 2004), are now paid for utilizing, while studies like Chang et al. (2022); Veluri et al. (2024); Ma et al. (2024) rely on private, inaccessible data.

To tackle this bottleneck, we construct and release the OpenETD dataset, an open-to-public dataset that is specifically designed for training and evaluating end-turn detection models. The OpenETD dataset consists of more than 120k sam-

Model	Param.	Acc.
GRU	1M	79.7
Wav2vec 2.0	94M	99.3

Table 1: **Model accuracy vs. size trade-off.** Experimental results on our OpenETD dataset. The 1M GRU model can achieve real-time on-device but exhibits lower accuracy.

ples with over 300 hours of conversational data. In particular, the dataset consists of both synthetic spoken conversation generated via applying text-to-speech (TTS) generation on text dialogue dataset and real-world dialogues collected from web sources.

Beyond dataset construction, we propose an efficient end-turn detection algorithm tailored for real-time chatbot interactions under resource-constrained environments. While transformer-based audio processing models achieve strong performance, their high computational cost makes them impractical to be used in highly repetitive tasks with real-time applications, such as ETD. In contrast, models under 1M parameters can run efficiently on local devices (Qi et al., 2023; Wu and Chen, 2023) but suffer from degraded accuracy (see Table 1). To address this gap, we introduce SpeculativeETD, a novel collaborative inference framework that balances efficiency and accuracy. In this framework approach, we adopt both the lightweight GRU-based model (Chung et al., 2014) and the high-performance Wav2vec-based model (Baevski et al., 2020). Here, the lightweight model operates on local devices in real time and keeps track of whether the speaker is speaking or not (either as a pause or end-of-turn). Whenever the model detects the silence, the model queries the high-performance model to make a more challenging prediction, *i.e.*, telling whether the silence marks the end of the speaker’s turn or not. This framework removes the need for the high-performance model to be running in real-time, dramatically saving the amount of required computations with minimal degradation in accuracy.

Our key contributions can be summarized as follows:

- We release the first open-source dataset for end-turn detection, incorporating synthetic and real-world speech data.
- We propose a novel method, SpeculativeETD, for efficient and accurate end-turn detection, leverag-

ing a lightweight on-device model and a server-side model.

2 Related work

Turn-taking. Turn-taking plays a crucial role in human communication, allowing speakers to transition smoothly without interruptions. Its detection is guided by linguistic, prosodic, and non-verbal cues such as pitch variation, speech rhythm, and gaze direction (Gravano and Hirschberg, 2011; Levinson and Torreira, 2015). Early turn-taking models have used finite-state machines to predict turn timing and duration (Raux and Eskenazi, 2009).

In spoken dialogue systems, turn-taking ensures fluid human-computer interactions. Various approaches have been explored for turn boundary prediction, including end-to-end models based on automatic speech recognition (ASR) (Chang et al., 2022) and transformer-based models incorporating contextual and pragmatic cues (Ekstedt and Skantze, 2020). More recently, full-duplex LLM-based agents have been developed to enable simultaneous listening and responding (Défossez et al., 2024). However, existing models remain computationally expensive and rely on private datasets, limiting real-time applicability and reproducibility. Our approach addresses these challenges by introducing an efficient turn-detection method and an open-source dataset.

Disfluency Detection. Disfluencies, such as interruptions, filler words, or self-corrections, occur frequently in spontaneous speech and can disrupt the natural flow of conversations. Identifying and handling these phenomena is crucial for maintaining coherent interactions in spoken dialogue systems. Traditional approaches primarily rely on text-based detection, analyzing transcriptions with span classification models to improve accuracy (Ghosh et al., 2022). Although effective, these methods depend on ASR, which introduces latency and errors. Recent studies have explored direct speech-based disfluency detection (Zhou et al., 2024) and integrated frameworks that combine ASR with disfluency removal for smoother real-time processing (Lou and Johnson, 2020). However, these methods often struggle with spontaneous speech variations and real-time efficiency.

Chatbot Agents. LLMs have advanced chatbot systems, enabling more natural spoken dialogue interactions (Lin et al., 2022; Rubenstein et al., 2023).

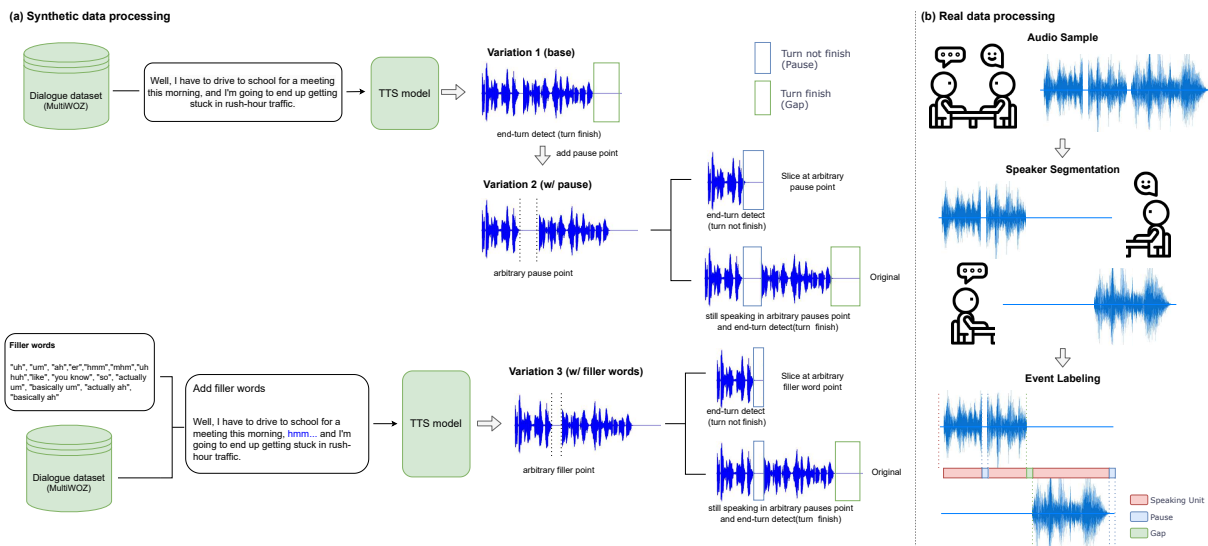


Figure 2: **Data generation methodology overview.** (a) The synthetic data pipeline converts text-based dialogue data into three types of speech variations using a text-to-speech (TTS) system. (b) The real data processing pipeline involves collecting and processing speech data from online sources.

Traditional chatbots rely on turn-based exchanges, whereas recent models explore real-time processing to handle overlapping speech, interruptions, and backchanneling (Veluri et al., 2024; Ma et al., 2024).

However, LLM-driven agents often suffer from high computational costs due to their autoregressive inference paradigm, leading to slow response times. Additionally, while some models aim to adaptively handle turn-taking, they frequently miss fine-grained conversational cues like hesitations and mid-turn pauses, resulting in suboptimal interactions (Umair et al., 2024). Another challenge is the reliance on proprietary datasets, restricting accessibility for further improvements (Cieri et al., 2004). To address these challenges, we incorporate speculative inference for efficient real-time processing and introduce an open-source dataset for end-turn detection, fostering further research in spoken dialogue systems.

3 OpenETD: End-Turn Detection Dataset

Before introducing the dataset, we first formally describe the task of end-turn detection (ETD). The goal of ETD is to infer from the given speech whether the speaker has finished speaking or not. More precisely, at each time t , the speaker is at one of the three states. (1) *Speaking unit (SU)*: The speaker is in speech. (2) *Pause*: The speaker is not in speech but is only at a brief pause and intends to keep on talking. (3) *Gap*: The speaker has finished

talking, marking the end of his or her turn.

The OpenETD dataset consists of multiple audio speech files labeled with the ternary segmentation information. That is, the label of each audio file is a sequence of the (state, starting time) pair of each same-state segment, *e.g.*,

$$\left((SU, 0s), (Gap, 12.3s), (SU, 13.7s), (Gap, 15.0s), \dots \right). \quad (1)$$

We construct the OpenETD dataset using both synthetic (for both training and evaluation) and real (for evaluation only) conversational audio segments.

3.1 Synthetic data generation

We have constructed the synthetic audio dataset of multi-turn conversation by applying the TTS (text-to-speech) model to the existing text dialogue dataset. In particular, we have used the MultiWOZ corpus (Budzianowski et al., 2018) as the base dataset. As the TTS model, we have used Google Cloud Text-to-Speech.

An important thing to note here is that a naïve text-to-speech translation of text dialogues will generate data consisting only of speaking units and gaps, as the textual cues for the “pause” are typically missing in a text dialogue. Thus, we artificially insert the pauses to the conversation in three different styles (Figure 2(a)):

- **Base**: We do not insert any pause.

	All	V1 (base)	V2 (w/ pause)	V3 (w/ filler words)
Total				
# Samples	122,481	28,368	47,076	47,037
Total duration (h)	148.26	28.66	69.60	49.99
Average duration (s)	4.36	3.64	5.32	3.83
Train				
# Samples	96,773	22,321	37,246	37,206
Total duration (h)	116.83	22.49	54.87	39.47
Average duration (s)	4.35	3.63	5.30	3.82
Dev				
# Samples	12,840	3,008	4,916	4,916
Total duration (h)	15.75	3.06	7.37	5.32
Average duration (s)	4.42	3.66	5.40	3.90
Test				
# Samples	12,868	3,039	4,914	4,915
Total duration (h)	15.68	3.11	7.36	5.20
Average duration (s)	4.39	3.69	5.39	3.81

Table 2: **OpenETD synthetic data spans 148 h across three pause variations and disjoint train/dev/test splits.** Statistics of the synthetic data grouped by pause variation: V1 is the base dialogue without explicit pauses, V2 injects pause silences, and V3 additionally inserts filler words before the pause. Splits follow the MultiWOZ partitioning of source dialogues (no speaker overlap across splits). “V” denotes different variations of pauses.

- **w/ Pause:** We randomly select tiny hesitations—typically inserted by TTS models to mimic human breathing (Braunschweiler and Buchholz, 2011; Hwang et al., 2023; Yang et al., 2024)—and extend them into pauses. Here, we sample each pause duration X from an Erlang distribution $\text{Erlang}(k, \lambda)$ with shape $k=3$ and rate $\lambda=4.29$, truncated to $[0.1, 3.0]$ s. This yields a right-skewed distribution with mean 0.70 s and peak around 0.4–0.6 s, matching the typical pattern of conversational pauses (many short, few long). Statistical validation against real data is provided in Section A. For some randomly picked samples, we also remove the segments that follow the pause, generating samples without a gap at the end of the audio file.
- **w/ Filler words:** We randomly inject the filler words, *e.g.*, “um” or “uh,” at arbitrary locations in the text dialogue. After the TTS generation, we add a pause after the filler words. Again, we randomly select a fraction of samples and remove the segments after the pause to generate a sample that does not end with a gap.

3.2 Real data generation

We have also collected real spoken conversation data from YouTube and Buckeye speech corpus (Pitt et al., 2005); we have selected these sources as

they contain many spontaneous, natural dialogues between two speakers with frequent turn-takings. For YouTube, we used broad queries (*e.g.*, “*english conversation podcast*”, “*podcast*”) via yt-dlp, retrieving up to 1,000 relevance-ranked results per query, and kept only two-speaker conversations after diarization (monologues and multi-party discussions were filtered out). To ensure that the collected samples involve exactly two speakers, we have applied a pretrained speaker diarization toolkit (Bredin, 2023) to each audio file and filtered out those with more than two identified speakers.

Given conversation data, we again utilized the speaker diarization tool to generate segmentations of the speech of two different speakers (Figure 2 (b)). Any silent interval between speech segments that exceed 200ms is labeled as either pause or gap, depending on whether the neighboring speech segments belong to the same speaker or not; here, the 200ms criterion follows the commonly used thresholds in the turn-taking literature (Stivers et al., 2009; Heldner and Edlund, 2010; Nguyen et al., 2022; Li et al., 2022). Similarly to the synthetic dataset, we have randomly selected some samples and removed the audio segment that follows a randomly chosen gap or pause.

	All	{SU,Pause}	{SU,Gap}	{SU,Pause,Gap}	Total Pause	Total Gap
Total						
# Samples	8,987	3,756	649	4,582	-	-
Duration (h)	165.75	43.98	8.08	113.69	7.78	3.47
YouTube						
# Samples	6,089	2,643	514	2,932	-	-
Duration (h)	133.56	35.58	6.96	90.97	4.82	2.19
Buckeye						
# Samples	2,898	1,113	135	1,650	-	-
Duration (h)	32.21	8.40	1.13	22.72	2.96	1.27
Train						
# Samples	6,290	2,624	449	3,217	-	-
Duration (h)	117.16	31.12	5.47	80.58	5.54	2.46
Dev						
# Samples	899	363	67	469	-	-
Duration (h)	16.23	3.60	0.91	11.72	0.74	0.34
Test						
# Samples	1,798	769	133	896	-	-
Duration (h)	32.36	9.26	1.70	21.40	1.50	0.67

Table 3: **OpenETD real data covers 166 h of YouTube and Buckeye conversations with disjoint train/dev/test splits.** Statistics of the real data stratified by source platform and segment composition (whether an audio file contains speaking units with Pause, Gap, or both). “Total Pause” and “Total Gap” report the aggregate duration of the respective silence intervals obtained from the diarized timestamps.

3.3 Other details

Splits and statistics. The default train/dev/test split of the synthetic data follows that of the MultiWOZ dataset (Budzianowski et al., 2018). For the real data, we partition the collected YouTube and Buckeye conversations into disjoint train/dev/test splits at the file level, so that no source conversation appears in more than one split. The statistics of the synthetic and real datasets are given in Table 2 and Table 3, respectively.

Human validation. To quantify the label noise introduced by the automatic diarization + 200 ms rule, we conducted a human validation study on a subset of real data (three annotators, 96 clips). Annotators reached 85.4% agreement with the automatic labels and rated diarization quality at 4.17/5.0 on average; full results are in Section C.

Files. For real data crawled from YouTube, we do not directly share the audio files to avoid license issues. Instead, we release the code and public URL to download the audio files.

4 Method

In this section, we describe *SpeculativeETD*, a framework that achieves a favorable tradeoff of computational efficiency and accuracy for end-turn detection.

For a smooth real-time conversation, one needs to run the end-turn detector repetitively at a rapid frequency. Thus, it is desirable to deploy an ETD model with an extremely small computational footprint. However, lightweight models—*e.g.*, a recurrent model with a single GRU layer—tend to suffer from a low detection accuracy. On the other hand, heavier transformer-based models typically require a large computational cost despite being highly accurate in the end-turn detection (see Section 6).

To enjoy the best of both worlds, the *SpeculativeETD* is designed as a collaborative inference framework that utilizes both a lightweight model (say, GRU), which runs on-device, and a heavier model (say, Wav2vec 2.0), which runs on the server. In a nutshell, the proposed *SpeculativeETD* operates in two stages (Figure 3):

1. **On-device:** The lightweight recurrent model processes the streaming audio signal frame-by-frame (or chunk-by-chunk) and detects whether the user’s speech has finished. In other words, the model conducts a binary classification between the Speaking Unit (SU) and non-SU (Gap and Pause). As this task is relatively easier than distinguishing between gaps and pauses, we can achieve high accuracy with tiny computational costs and latency.

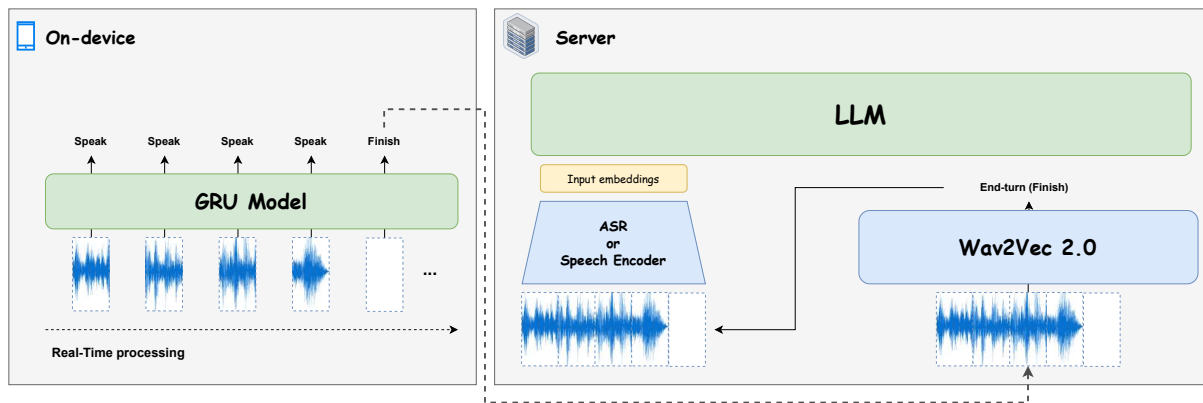


Figure 3: **SpeculativeETD framework overview.** A lightweight model (a 1M-parameter GRU in this example) operates as the on-device model, enabling real-time processing. A high-performance model (a 94M-parameter Wav2Vec 2.0 base model in this case) serves as the server-side model, verifying the predictions of the on-device model.

2. **Server-side:** Given a speech segment that ended with a silence, the heavier transformer-based model (e.g., a Wav2Vec 2.0) predicts whether the silence indicates the gap or the pause. Whenever the gap is detected, the language model will be called for generating an appropriate response to the speaker.

The name of the method—SpeculativeETD—comes from its structural resemblance to the *speculative decoding* (Leviathan et al., 2023), a popular decoding algorithm for large language models that use a smaller and faster drafter model for autoregressive generation and larger models for parallel verification. We note, however, that our method critically differs from the speculative decoding in the sense that the small model and large model do not predict the same output classes in our framework; rather than playing the role of a verifier, the large model is used for making a more difficult fine-grained prediction, conditioned on the predictions of the small model.

Advantages. The key advantages of this framework are twofold: (1) The trickier decision that requires a large computation (*i.e.*, telling “Gap” vs. “Pause”) can happen only once per each consecutive segment of silence, not at every time frame. As we will see in Section 6, this helps save the computation more than $10\times$ with only a small degradation prediction accuracy. (2) The two-stage pipeline allows us to utilize on-device computation by placing the lightweight model on the edge. By doing so, we no longer need the communication between the edge device and the server to be done in a continuous manner; instead, we can communicate once

every silence.

Inference Protocol. During real-time operation, the on-device model processes each new 100 ms chunk in constant time. The server-side Wav2vec 2.0 is triggered once the on-device GRU predicts non-SU (silence) for at least 200 ms, *i.e.*, two consecutive 100 ms chunks. Only the accumulated audio from the start of that silence segment is sent to the server. The server model then classifies the segment as Gap or Pause, determining whether an end-turn event has occurred or if streaming should continue. The 200 ms trigger follows the standard turn-taking threshold (Stivers et al., 2009; Heldner and Edlund, 2010).

5 Experimental Setup

We now empirically validate the usefulness of the constructed OpenETD dataset (Section 3) and the effectiveness of the proposed method, SpeculativeETD (Section 4).

Task. We consider two tasks for the OpenETD:

- *Binary classification:* When a speech segment comes as an input, the detector reads the whole audio and determines whether the speech segment has ended (Gap) or not (Pause).
- *Real-time audio segmentation:* The speech arrives in real-time, and at each interval (*e.g.*, every 100ms), the detector determines whether the current state of the speaker is SU, Pause, or Gap. That is, we are conducting a ternary segmentation in a sequential manner.

Models. We evaluate a total of four models for end-turn detection.

- VAP (Ekstedt and Skantze, 2022): A popular open-source pretrained turn-taking model, which can also be employed for end-turn detection tasks. We use the pretrained encoder as frozen and train the predictor module of the model from scratch using the OpenETD dataset. We have used the “comparative” predictor head, which fits the ETD task.
- GRU (Chung et al., 2014): We construct a lightweight GRU model with about 202K parameters, consisting of a Conv2D frontend (2 layers) and a single GRU layer on top of a 40-dim log-mel input computed on 100 ms chunks (full input specification in Section B), and train it from scratch on the OpenETD dataset.
- Wav2vec 2.0 (Baevski et al., 2020): A relatively large transformer-based model with 94M parameters, which achieves high performance in many speech processing tasks. We fine-tune the full model using the OpenETD dataset.
- SpeculativeETD (Ours): We construct an ETD model using the GRU and Wav2vec 2.0 model following the same settings as described above. Note that, as this method assumes a real-time scenario, we evaluate this method only in real-time audio segmentation scenarios and not in binary classification.

Evaluation. For the binary classification task, performance is quantified using Precision, Recall, F1-score, and accuracy metrics. In the real-time segmentation task, we assess the F1-score at 100ms intervals and evaluate segmentation quality using the Intersection over Union (IoU) metric for the average of each of the three classes (Speaking Unit, Gap, Pause).

Datasets for training/evaluation. In all experiments, we train the models on a mixture of the training splits of the synthetic and real data in the OpenETD dataset; the synthetic split provides controllable pause/gap patterns, while the real split grounds the models in natural conversational speech. For evaluation, we use the held-out test splits of both the synthetic and real datasets, with no overlap with the training or development sets.

Optimization. We have used the AdamW optimizer (Loshchilov and Hutter, 2019) for all tasks and models and trained for 10 epochs. The learning rate has been selected with a random search over

the interval $[3 \times 10^{-6}, 3 \times 10^{-4}]$, using the validation split of the synthetic OpenETD dataset. The weight decay values are also randomly searched in the range $[0.01, 2.00]$. The batch size has been tuned over $\{128, 256, 512\}$ for GRU, $\{8, 16, 32\}$ for VAP, and $\{4, 8, 16\}$ for Wav2vec 2.0. The searched hyperparameters are provided in Section H.

Hardware. All experiments have taken place on a single GPU; some experiments have taken place on an NVIDIA L40S, while some have been done on NVIDIA RTX 6000 Ada. To measure the inference latency of ETD models, we used the iPhone 12 mini; the computing took place on the A14 bionic chipset with Hexa-core CPUs.

6 Results

6.1 Binary classification

In Table 4, we report the binary classification results of ETD models. From the table, we observe that the GRU models—which are much lighter than other models—achieve a significantly lower accuracy than other methods, with around 20%p gap in both synthetic and real datasets. This result motivates us to develop a computationally efficient framework that can achieve both efficiency and accuracy.

6.2 Real-time audio segmentation

In Table 5, we compare the real-time audio segmentation performances of the proposed SpeculativeETD against three baseline methods. The results demonstrate that the SpeculativeETD achieves a performance that is comparable to a larger-scale model, namely Wav2vec 2.0, on the synthetic set. The proposed method also outperforms VAP, which requires much more computation.

6.3 Efficiency of SpeculativeETD

FLOPs comparison. We evaluate the computational cost of our method in terms of FLOPs using 100 synthetic samples. By leveraging the server-side Wav2vec 2.0 model only when vital, SpeculativeETD achieves comparable performance to Wav2vec 2.0 with substantially lower FLOPs. Moreover, latency remains unaffected as Wav2vec 2.0 runs on a server GPU rather than on-device.

Latency. We verify whether the latency is suitable for real-time on-device deployment. We mea-

Methods	Synthetic data				Real data			
	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.
VAP (Ekstedt and Skantze, 2022)	91.5	93.5	92.1	92.3	59.8	58.7	59.1	69.6
GRU (Chung et al., 2014)	78.8	77.6	78.1	79.7	52.7	51.4	49.8	69.0
wav2vec 2.0 (Baevski et al., 2020)	99.3	99.2	99.2	99.3	76.3	74.3	75.2	81.2

Table 4: **Binary classification results.** Performance of various models on the synthetic and real datasets of the OpenETD dataset.

Methods	Synthetic data		Real data	
	F1	IoU	F1	IoU
VAP (Ekstedt and Skantze, 2022)	90.6	84.8	33.2	25.9
GRU (Chung et al., 2014)	58.0	52.2	34.2	31.7
Wav2Vec2 (Baevski et al., 2020)	94.7	90.2	58.4	46.2
SpeculativeETD (Ours)	<u>94.0</u>	<u>88.9</u>	<u>45.6</u>	<u>37.8</u>

Table 5: **SpeculativeETD matches Wav2Vec2 on synthetic data and substantially outperforms VAP and GRU baselines on real data.** Real-time 3-class segmentation (Gap/SU/Pause) results on the synthetic and real test sets of the OpenETD dataset. F1 and IoU are macro-averaged across the three states. The best results are marked with **bold**, and the runner-up is underlined. SpeculativeETD retains Wav2Vec2-level synthetic performance while achieving 26.7× fewer W2V calls on real audio.

Methods	Compute (MFLOPs)
VAP	10,354.98
GRU	45.34
wav2vec 2.0	34,971.68
SpeculativeETD	$45.34 + 874.30 = 919.64$

Table 6: **FLOPs comparison for real-time processing.** Computational cost measured in MFLOPs for processing 100 samples. SpeculativeETD achieves comparable performance to wav2vec 2.0 with 38× fewer FLOPs.

sure the latency via LiteRT¹ on iPhone mini 12 (iOS 16.3.1). We check model loading, initialization, and execution time, which is shown in Table 7. We averaged three runs of latency measurement, and the execution time was the average of 50 iterations of inference time when the model performed inference every 100 ms on a 5-second speech audio.

Our experiments demonstrate that SpeculativeETD, which utilizes GRU on the on-device side, achieves latency below 1 ms per 100 ms interval. As illustrated in Fig. 5, the inference latency of Wav2vec 2.0 significantly increases with input length due to the computational overhead of transformer architectures, while GRU latency remains

¹<https://ai.google.dev/edge/litert>

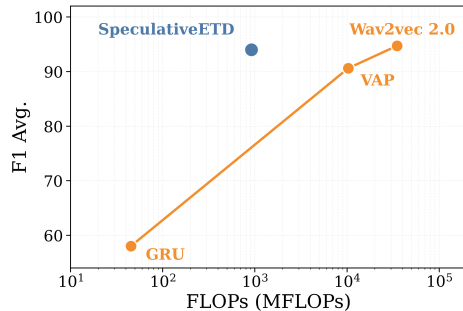


Figure 4: **FLOPs vs. performance comparison.** Visualization of computational cost (FLOPs) and performance for each model. SpeculativeETD achieves high accuracy with substantially lower computational requirements compared to Wav2vec 2.0.

Methods	Latency (ms)		
	Load	Init	Execute
wav2vec 2.0	874.06	17.89	1500.32
GRU (SpeculativeETD)	1.16	3.85	0.26

Table 7: **Latency analysis in a real-time on-device setting.** The ‘Execute’ latency is the average inference time when the model performs inference every 100 ms on a 5-second speech audio. SpeculativeETD with GRU achieves sub-millisecond execution latency.

stable. On top of on-device inference, we also measured end-to-end audio-transfer latency over 5G and Wi-Fi; round-trip times stay within 106–140 ms across payload sizes up to 10 s, well inside the 200 ms turn-taking threshold (see Section E). These results confirm that SpeculativeETD is an excellent option for serving real-time on-device.

7 Conclusion

We introduce the OpenETD dataset, a novel publicly available dataset for end-turn detection in spoken dialogue systems, and propose SpeculativeETD, an efficient method for real-time end-turn detection. These contributions address critical challenges in developing more natural and responsive spoken dialogue agents, particularly those powered

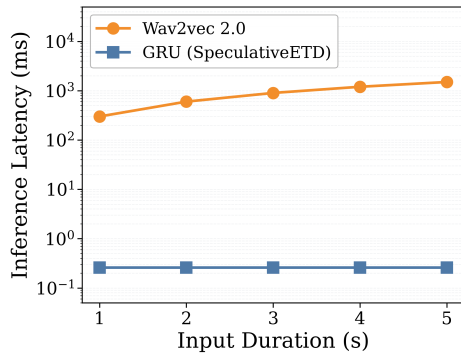


Figure 5: **Inference latency comparison.** Visualization of inference latency on real-time on-device settings. As input length increases, Wav2vec 2.0 latency grows significantly due to transformer overhead, while GRU maintains consistently low latency, confirming SpeculativeETD’s suitability for real-time deployment.

by large language models. The OpenETD dataset comprises synthetic and real data and is the first public dataset for end-turn detection research. This dataset enables more comprehensive training and evaluation of end-turn detection models by offering a diverse range of conversational scenarios, including artificially extended pauses and injected filler words. Also, real data further enhances the dataset’s utility in assessing model generalizability in real-world conversational settings. SpeculativeETD, represents a significant advancement in balancing efficiency and accuracy for real-time end-turn detection. Combining a lightweight GRU on-device model for on-device processing with a high-performance Wav2Vec 2.0 server-side model on the server improves accuracy while maintaining low latency. This two-stage framework effectively addresses the computational constraints of on-devices while leveraging the superior performance of larger models.

Limitations

Our work has several limitations. First, the OpenETD dataset primarily focuses on English conversations, which may limit its applicability to other languages with different turn-taking patterns. Second, while SpeculativeETD demonstrates strong performance, the reliance on server-side computation for Gap/Pause classification introduces network latency considerations not addressed in our current evaluation. Third, our synthetic data generation, while effective, may not fully capture the complexity of spontaneous human speech patterns.

Ethics Statement

All datasets used in this research are used exclusively for academic purposes. We release open-source processing code and, for data that cannot be directly shared due to licensing constraints, provide reproducible download scripts. We build on four data sources with distinct terms: MultiWOZ (Budzianowski et al., 2018) (Apache 2.0), Google Cloud Text-to-Speech (used under the Google Cloud Terms of Service), YouTube audio (redistributed only as URLs with download scripts; we do not rehost audio files), and the Buckeye Corpus (Pitt et al., 2005) (used under its Academic License). Full licensing details are provided in Section F.

Acknowledgments

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (Nos. RS-2024-00457882, RS-2019-II191906, 2022-0-00713) and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (Nos. RS-2024-00453301, RS-2025-24873016, RS-2026-25494004).

References

- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.
- Norbert Braunschweiler and Sabine Buchholz. 2011. Automatic sentence selection from speech corpora including diverse speech for improved hmm-tts synthesis quality.
- Hervé Bredin. 2023. pyannotate.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe. In *Proc. INTERSPEECH 2023*.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.
- Shuo-yiin Chang, Bo Li, Tara N Sainath, Chao Zhang, Trevor Strohman, Qiao Liang, and Yanzhang He. 2022. Turn-taking prediction for natural conversational speech. *arXiv preprint arXiv:2208.13321*.

- Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.
- Christopher Cieri, David Miller, and Kevin Walker. 2004. [The fisher corpus: a resource for the next generations of speech-to-text](#). In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).
- Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. [Moshi: a speech-text foundation model for real-time dialogue](#). *Preprint*, arXiv:2410.00037.
- Erik Ekstedt and Gabriel Skantze. 2020. [TurnGPT: a transformer-based language model for predicting turn-taking in spoken dialog](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2981–2990, Online. Association for Computational Linguistics.
- Erik Ekstedt and Gabriel Skantze. 2022. Voice activity projection: Self-supervised learning of turn-taking events. *arXiv preprint arXiv:2205.09812*.
- Sreyan Ghosh, Sonal Kumar, Yaman Kumar Singla, Rajiv Ratn Shah, and S. Umesh. 2022. [Span classification with structured information for disfluency detection in spoken utterances](#). *Preprint*, arXiv:2203.16028.
- Agustín Gravano and Julia Hirschberg. 2011. [Turn-taking cues in task-oriented dialogue](#). *Computer Speech & Language*, 25:601–634.
- Mattias Heldner and Jens Edlund. 2010. [Pauses, gaps and overlaps in conversations](#). *Journal of Phonetics*, 38(4):555–568.
- Ji-Sang Hwang, Sang-Hoon Lee, and Seong-Whan Lee. 2023. Pausespeech: Natural speech synthesis via pre-trained language model and pause-based prosody modeling. In *Asian Conference on Pattern Recognition*, pages 415–427. Springer.
- Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.
- Stephen C. Levinson and Francisco Torreira. 2015. [Timing in turn-taking and its implications for processing models of language](#). *Frontiers in Psychology*, 6:731.
- Siyan Li, Ashwin Paranjape, and Christopher D. Manning. 2022. [When can i speak? predicting initiation points for spoken dialogue agents](#). *Preprint*, arXiv:2208.03812.
- Ting-En Lin, Yuchuan Wu, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2022. Duplex conversation: Towards human-like interaction in spoken dialogue systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3299–3308.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *ICLR*.
- Paria Jamshid Lou and Mark Johnson. 2020. [End-to-end speech recognition and disfluency removal](#). *Preprint*, arXiv:2009.10298.
- Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. 2024. Language model can listen while speaking. *arXiv preprint arXiv:2408.02622*.
- Kentaro Mitsui, Yukiya Hono, and Kei Sawada. 2023. Towards human-like spoken dialogue generation between ai agents from written dialogue. *arXiv preprint arXiv:2310.01088*.
- Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoit Sagot, Abdelrahman Mohamed, and Emmanuel Dupoux. 2022. [Generative spoken dialogue language modeling](#). *Preprint*, arXiv:2203.16502.
- Mark A. Pitt, Keith Johnson, Elizabeth Hume, Scott F. Kiesling, and William D. Raymond. 2005. [The buckeye corpus of conversational speech: labeling conventions and a test of transcriber reliability](#). *Speech Commun.*, 45:89–95.
- Xuan Qi, Zegang Sun, Xue Mei, Ryad Chellali, and Yugen Yi. 2023. [A lightweight binarized convolutional neural network model for small memory and low-cost mobile devices](#). *Mob. Inf. Syst.*, 2023.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. [Robust speech recognition via large-scale weak supervision](#). *Preprint*, arXiv:2212.04356.
- Antoine Raux and Maxine Eskenazi. 2009. [A finite-state turn-taking model for spoken dialog systems](#). In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 629–637, Boulder, Colorado. Association for Computational Linguistics.
- Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, and 1 others. 2023. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*.
- Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. 2024. Assisting in writing wikipedia-like articles from scratch with 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)*, pages 6252–6278.
- Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009.
- Tanya Stivers, N. J. Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter de Ruiter, Kyung-Eun Yoon, and Stephen C. Levinson. 2009. [Universals and cultural variation in turn-taking in conversation](#). *Proceedings of the National Academy of Sciences*, 106(26):10587–10592.
- Louis Ten Bosch, Nelleke Oostdijk, and Lou Boves. 2005. On temporal aspects of turn taking in conversational dialogues. *Speech Communication*, 47(1-2):80–86.
- Muhammad Umair, Vasanth Sarathy, and Jan Ruiter. 2024. Large language models know what to say but not when to speak. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15503–15514.
- Bandhav Veluri, Benjamin Peloquin, Bokai Yu, Hongyu Gong, and Shyamnath Gollakota. 2024. Beyond turn-based interfaces: Synchronous llms as full-duplex dialogue agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21390–21402.
- YiHeng Wu and JianXin Chen. 2023. [A lightweight real-time system for object detection in enterprise information systems for frequency-based feature separation](#). *Int. J. Semant. Web Inf. Syst.*, 19(1):1–18.
- Chengxing Xie, Canyu Chen, Feiran Jia, Ziyu Ye, Shiyang Lai, Kai Shu, Jindong Gu, Adel Bibi, Ziniu Hu, David Jurgens, James Evans, Philip Torr, Bernard Ghanem, and Guohao Li. 2024. [Can large language model agents simulate human trust behavior?](#) In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Haoqiu Yan, Yongxin Zhu, Kai Zheng, Bing Liu, Haoyu Cao, Deqiang Jiang, and Linli Xu. 2024. [Talk with human-like agents: Empathetic dialogue through perceptible acoustic reception and reaction](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15009–15022, Bangkok, Thailand. Association for Computational Linguistics.
- Dong Yang, Tomoki Koriyama, and Yuki Saito. 2024. Frame-wise breath detection with self-training: An exploration of enhancing breath naturalness in text-to-speech. *arXiv preprint arXiv:2402.00288*.
- Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. 2024. [Agent-pro: Learning to evolve via policy-level reflection and optimization](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5348–5375, Bangkok, Thailand. Association for Computational Linguistics.
- Xuanru Zhou, Anshul Kashyap, Steve Li, Ayati Sharma, Brittany Morin, David Baquirin, Jet Vonk, Zoe Ezzes, Zachary Miller, Maria Luisa Gorno Tempini, Jiachen Lian, and Gopala Krishna Anumanchipalli. 2024. [Yolo-stutter: End-to-end region-wise speech dysfluency detection](#). *Preprint*, arXiv:2408.15297.

Comparison	KS stat	Cohen’s d
<i>Duration</i>		
Syn vs. Real gap duration	0.083	0.12
Syn vs. Real pause duration	0.092	0.15
<i>Position within utterance</i>		
Real pause vs. Uniform(0, 1)	0.091	–
Real gap vs. Uniform(0, 1)	0.094	–
Syn vs. Real gap (position)	0.238	0.23
Syn vs. Real pause (position)	0.238	0.27

Table 8: **Synthetic pauses match real duration statistics but not exact positions.** Kolmogorov–Smirnov (KS) two-sample test and Cohen’s d comparing synthetic and real pause/gap distributions. Duration statistics are nearly identical (KS= 0.083), confirming that the Erlang-fitted generator captures real pause lengths. Real-data positions are close to uniform within an utterance, so exact position replication is inherently difficult; synthetic data therefore serves as augmentation rather than perfect replication.

A Synthetic pause generation: statistical validation

Our synthetic pipeline does not insert pauses at random positions. Instead, we detect micro-hesitations naturally produced by the TTS model (Braunschweiler and Buchholz, 2011; Hwang et al., 2023; Yang et al., 2024) and extend their duration using an Erlang distribution fitted to real pause durations.

Duration. Table 8 reports Kolmogorov–Smirnov (KS) two-sample statistics and Cohen’s d comparing synthetic and real gap/pause durations. The KS statistic for gap duration is only 0.083 (Cohen’s $d = 0.12$), indicating that the Erlang-fitted generator closely matches real duration distributions.

Position. In real conversational data, both pause and gap positions within an utterance are close to uniform (KS vs. Uniform(0, 1): pause 0.091, gap 0.094), reflecting the inherent unpredictability of pause placement. Synthetic positions show a larger divergence (KS = 0.238 for both pause and gap). We acknowledge this as a limitation; however, exact position replication is inherently difficult given that real positions themselves are near-uniform.

Augmentation value. The synthetic data therefore functions as augmentation rather than exact replication. Table 9 shows that training SpeculativeETD on the mixture consistently outperforms training on either split alone, confirming the practical value of the synthetic set.

Training mode	Real F1	Real IoU	Δ F1	Δ IoU
Mix (syn + real)	45.6	37.8	–	–
Real only	43.1	36.3	–2.5	–1.5
Synthetic only	44.0	36.7	–1.6	–1.1

Table 9: **Mixed training consistently improves real-data performance, confirming synthetic data as effective augmentation.** SpeculativeETD real-time segmentation performance on the real test set when trained on synthetic only, real only, or the mixture. Mixed training achieves the best F1 and IoU; Δ denotes degradation relative to the Mix row.

B On-device GRU input specification

We detail the input pipeline of the on-device GRU used in SpeculativeETD.

Mel-spectrogram input. Each input chunk corresponds to 100 ms of audio sampled at 16 kHz. We extract a 40-dim log-mel spectrogram with a 25 ms window and a 10 ms hop, yielding 10 frames per chunk, followed by per-utterance z-score normalization. The resulting chunk tensor has shape (40, 10).

Conv2D frontend. Two Conv2D layers downsample the chunk: Conv(1 \rightarrow 16, $k=5, s=2, p=2$) \rightarrow ReLU \rightarrow Conv(16 \rightarrow 32, $k=3, s=2, p=1$) \rightarrow ReLU, producing a 960-dim per-chunk feature vector after flattening.

GRU backbone. A single-layer GRU (hidden size 64) consumes the chunk features autoregressively, and a linear head outputs the binary (SU vs. non-SU) logits. The entire on-device model has approximately 202K parameters.

Filler word vocabulary. During synthetic data generation (Section 3.1), we inject filler words drawn from the following vocabulary: “uh”, “um”, “ah”, “er”, “hmm”, “mhm”, “uh huh”, “like”, “you know”, “so”, “actually um”, and “basically um”. These cover the common English hesitation markers observed in conversational speech.

C Human validation of pause/gap labels

To quantify label noise introduced by the automatic diarization + 200 ms rule, we performed a human validation study. Three annotators each judged 96 randomly sampled real clips (50 Pause, 46 Gap). Each clip was presented with 2 s of audio before the silence and 3 s after, and annotators reported (i) whether the automatic label was correct and (ii) the diarization segmentation quality on a 1–5 scale.

Metric	Pause ($n=50$)	Gap ($n=46$)	Overall ($n=96$)
Human–Auto agreement	94.0%	76.1%	85.4%
Diarization quality (1–5)	4.29 ± 1.00	4.04 ± 1.27	4.17 ± 1.14

Table 10: **Human validation confirms reliable pause labels and moderate gap-label noise.** Three annotators judged label correctness and diarization segmentation quality on 96 randomly sampled real clips (2 s before + silence + 3 s after). Pause labels show 94% human agreement, while Gap boundaries are harder to pinpoint (76.1%) due to ambiguous speaker-change boundaries. Diarization quality averages 4.17/5, indicating the automatic pipeline generally produces clean segmentations.

Table 10 summarizes the results. Pause labels are highly reliable (human–auto agreement 94.0%), while Gap labels are moderately noisier (76.1%) because speaker-change boundaries are intrinsically harder to localize. Overall diarization quality averages 4.17/5.0, supporting the use of the automatic pipeline. The 200 ms threshold follows the widely adopted convention in turn-taking research (Stivers et al., 2009; Heldner and Edlund, 2010).

D Domain analysis: synthetic vs. real

To characterize the mismatch between synthetic and real conversations, we analyzed audio-level and speaker-level statistics (Table 11). Signal-to-noise ratio (SNR) is computed from the raw waveform, Word-Per-Second (WPS) from Whisper-tiny transcripts, gender is derived from metadata, and speaker count reflects the distinct voice identities present.

The two largest domain gaps are SNR (39.7 dB vs. 10.5 dB) and speaking-rate variability (± 0.77 vs. ± 3.30). Real data also exhibits a richer emotion distribution (with a substantially higher share of sad speech) and spans five English accents, whereas synthetic data uses only two TTS voices (EN-US and EN-BR); see Table 11.

E End-to-end audio transfer latency

Although SpeculativeETD operates asynchronously—the on-device GRU continues detecting silence while the server classifies in the background—excessively long transfer times could still delay the classification decision. We therefore measured real audio-transfer round-trip time (RTT) from an iPhone 12 mini to a remote server via WebSocket, without model inference, across six payload sizes (100 ms–10 s of 16 kHz audio; 50 trials each).

Dimension	Synthetic	Real
Signal-to-noise ratio (dB)	39.7 ± 4.2	10.5 ± 13.7
Word per second	1.57 ± 0.77	2.87 ± 3.30
Gender (Female %)	66.2	39.4
Speaker count	3 TTS voices	~ 200

(a) Audio / speaker statistics.

	Emotion (%)					Accent, Real (%)				
	Happy	Sad	Angry	Disgust	Other	US	England	Canada	Indian	Australia
Real	33	33	11	5	18	44.0	16.6	16.6	14.6	8.2
Synthetic	30	8	19	30	13	Two TTS voices (EN-US, EN-BR)				

(b) Emotion distribution and real-data accent distribution.

Table 11: **Domain gaps between synthetic and real data: lower SNR, higher speaking-rate variability, richer emotion and accent diversity.**

(a) Audio-level and speaker-level statistics; SNR is computed from the raw waveform, speaking rate from Whisper-tiny transcripts, gender/speaker from metadata. (b) Emotion distribution (wav2vec2-based classifier) and English-accent distribution on the real side (dima806/speech-accent-classification); the synthetic set only contains two TTS accent voices. These gaps motivate real-data training and highlight limitations that synthetic augmentation cannot fully close.

Table 12 reports the results. On 5G, RTT ranges from 106 to 116 ms; on Wi-Fi, 98 to 140 ms—both comfortably inside the 200 ms turn-taking threshold. RTT is dominated by fixed network latency rather than payload size; a $100\times$ payload increase (3.1 KB \rightarrow 312.5 KB) adds only about 10 ms on 5G.

F Licensing of data sources

Table 13 summarizes the licenses of the data sources used to build OpenETD. We release the dataset and processing code in a manner consistent with each underlying license: the MultiWOZ-derived synthetic data inherits Apache 2.0; YouTube audio is not redistributed (we share URLs and download scripts only); Buckeye Corpus is used under its Academic License.

G Data quality improvement

Segment Filtering. We applied a structured segmentation process to better capture conversational dynamics. Consecutive speech fragments from the same speaker were merged into a Speaking Unit (SU) if separated by less than 200 ms of silence. Silences longer than 200 ms were labeled Pause (if within the same speaker) or Gap (if between different speakers), based on established turn-taking criteria (Stivers et al., 2009; Heldner and Edlund,

Audio	Payload	Mean RTT	P50	P95
<i>5G network</i>				
100 ms	3.1 KB	106.0 ms	104.0 ms	115.0 ms
500 ms	15.6 KB	106.4 ms	104.0 ms	119.0 ms
1 s	31.3 KB	106.4 ms	104.0 ms	119.0 ms
3 s	93.8 KB	107.2 ms	103.0 ms	110.0 ms
6 s	187.5 KB	109.9 ms	108.0 ms	128.0 ms
10 s	312.5 KB	116.2 ms	114.0 ms	129.0 ms
<i>Wi-Fi</i>				
100 ms	3.1 KB	100.2 ms	95.0 ms	131.0 ms
500 ms	15.6 KB	101.6 ms	94.0 ms	142.0 ms
1 s	31.3 KB	107.9 ms	99.0 ms	178.0 ms
3 s	93.8 KB	98.8 ms	94.0 ms	121.0 ms
6 s	187.5 KB	107.0 ms	100.0 ms	128.0 ms
10 s	312.5 KB	139.7 ms	115.0 ms	225.0 ms

Table 12: **End-to-end audio transfer stays well within the 200 ms turn-taking threshold across 5G and Wi-Fi.** Round-trip time (RTT) measured from iPhone 12 mini to a remote server via WebSocket without model inference, across six payload sizes (50 trials each). RTT is dominated by fixed network latency rather than payload size: a $100\times$ payload increase adds only ~ 10 ms on 5G.

Dataset / Source	License
MultiWOZ (Budzianowski et al., 2018)	Apache 2.0
Google Cloud Text-to-Speech	Google Cloud Terms of Service
YouTube audio	URLs + download scripts (not redistributed)
Buckeye Corpus (Pitt et al., 2005)	Academic License

Table 13: **Licenses of the data sources used to build OpenETD.** We release the MultiWOZ-derived synthetic data under a compatible license, redistribute only scripts and URLs for YouTube content, and respect the Buckeye Academic License in all downstream use.

2010). Using these labels, we grouped continuous SUs into segments, where each segment represents a stretch of speech bounded by either a Pause or a Gap. To focus on turn-relevant phenomena, we filtered out segments that did not contain any Pause or Gap, retaining only those that reflect actual hesitation or speaker transitions crucial for end-turn detection.

Language Filtering. To ensure the quality and consistency of the real-world speech data, we applied a language filtering step using the Whisper model (Radford et al., 2022). For each audio file, we used Whisper’s language detection capability to identify the spoken language. Since our focus is on English-based dialogue systems, we retained only the samples identified as English and discarded those in other languages. Table 14 summarizes the top 10 most frequent languages and the aggregate of other low-resource languages. Out of a total of 11,127 collected files, 8,022 were classi-

Language	Code	# Files	Included
English	en	8,022	✓
Hindi	hi	674	✗
Romanian	ro	334	✗
Urdu	ur	315	✗
Spanish	es	258	✗
Arabic	ar	179	✗
Swahili	sw	151	✗
Korean	ko	43	✗
Portuguese	pt	41	✗
French	fr	25	✗
Others (40+ languages)	—	1,055	✗

Table 14: **Language distribution before filtering.** Top-10 most frequent languages in the collected data. Only English files were retained in the final dataset.

fied as English and included in the final dataset. We additionally conducted a human evaluation of language identification performance by Whisper, which is recognized well as English. We checked all the data, and the accuracy of identification was 99.07%.

H Experiment details

Model	Hyper-parameter		
	epochs	batch size	lr
VAP	10	32	6×10^{-6}
GRU	10	256	3×10^{-4}
Wav2vec 2.0	10	8	5×10^{-6}

Table 15: **Hyper-parameter settings.** Training configuration used for each model in our experiments.

We detail the hyperparameter settings used for training each model in our experiments. All models were trained for 10 epochs. The VAP model was configured with a batch size of 32 and a learning rate of 6×10^{-6} . For the GRU model, we used a larger batch size of 256 and set the learning rate to 3×10^{-4} . In the case of Wav2vec 2.0, we employed a smaller batch size of 8 with a learning rate of 5×10^{-6} .