Beyond Fresh Starts: Stateful Inference for Streaming ASR in Conversational Voice Agents

Sameep Chattopadhyay[†] **Alexander Erdmann**[‡] **Mari Ostendorf**[†]
[†] University of Washington [‡] SRI International
sameepch@uw.edu alex.erdmann@sri.com ostendor@uw.edu

Abstract

Modern voice-agent systems rely on streaming speech recognition models that operate under stringent latency constraints. This study shows that, due to the limited memory constraints of real-time processing, these systems are adversely impacted by conversational phenomena such as long silences and backchannels. While many agentic pipelines mitigate this by resetting state at each turn, this approach discards vital context and impairs performance at turn onsets. We propose two state-management strategies that preserve cross-utterance context to reduce onset errors. In experiments with two state-of-the-art streaming models on two spoken dialogue benchmarks, our best method yields an average of 15–21% relative WER reduction at utterance onsets.

1 Introduction

The past few years have seen a rapid expansion in conversational voice agents. In 2026, the vast majority of deployed voice agents follow a cascaded speech-to-text, language model, and text-to-speech pipeline (Hamming AI, 2026). To mimic human turn-taking dynamics, these systems have to operate under tight latency budgets, with response delays below 500 ms (Stivers et al., 2009). Industry benchmarks suggest that ASR alone should contribute no more than 200 ms to the overall pipeline (Hamming AI, 2026), thus requiring streaming ASR, i.e., models that transcribe concurrently as the speaker talks.

Unlike offline systems that can utilize bidirectional global contexts, streaming ASR models must operate with very limited acoustic memory, often in the range of a few seconds (Rekesh et al., 2023; Défossez et al., 2024) to support real-time processing. The long silences and frequent backchannels (*e.g.*, "mm-hm", "uh-huh") in conversational speech pose a challenge to such low-latency models, by displacing crucial acoustic information from the prior utterance with more recent but task-irrelevant content (Hsiao et al., 2020). Most agentic pipelines handle this by simply resetting the state at each utterance turn, giving the model a fresh start. Useful prior context is then lost, sometimes causing the ASR model to falter on tokens at the beginning of a turn. Such errors may also affect downstream language-model components, as prior work suggests that LLMs often place disproportionate weight on early input tokens during inference (Xiao et al., 2024).

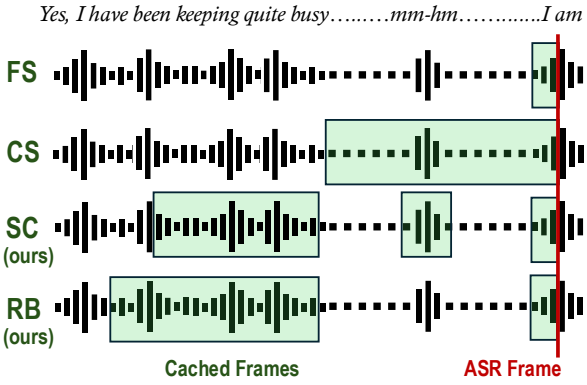

Figure 1: **Stateful inference.** Cached acoustic memory state for streaming ASR models under the different state-management strategies defined in Section 3.3.

In this work, we focus on the FastConformer RNN-Transducer (RNNT) architecture (Noroozi et al., 2024), a widely adopted architecture for low-latency streaming ASR. We first show that these models are susceptible to performance degradation caused by long pauses, backchannels, and state resets at turn boundaries. We then propose a set of simple state-management strategies[1] to mitigate these effects while preserving prior utterance context across turns. Evaluating on two of the best-performing low-latency streaming ASR models across two telephonic dialogue datasets, we achieve a 15–21% relative WER reduction at turn onsets over the standard state-resetting baselines.

[1]Code: https://github.com/Sameep-c/stateful-streaming-asr.git

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 38802–38811
October 24-29, 2026 ©2026 Association for Computational Linguistics

2 Streaming ASR

Deployed streaming ASR systems broadly fall into two inference paradigms. Live captioning and real-time transcription services generally process audio as a continuous stream from start to finish, relying on the model's memory to carry context across the entire recording (Macháček et al., 2023; Raj et al., 2022). In contrast, conversational voice agents used in dialogue settings pair streaming ASR with voice activity detection (VAD) or turn detection and decode each turn independently. In other words, the model state is reset at every utterance boundary (Huang et al., 2022; Narayanan et al., 2019).

The two frameworks have distinct failure modes for conversational speech. Continuous streaming accumulates information across the entire recording, including extended silences and non-lexical backchannels, thereby corrupting acoustic representations of subsequent speech (Hsiao et al., 2020; Choi et al., 2024). Dialogue agents avoid this problem by restarting the model at each turn, but at the cost of lost acoustic and linguistic context established in earlier turns. Previous work has shown that this particularly degrades performance in long-form RNN-T decoding (Narayanan et al., 2019; Ming et al., 2023; Hou et al., 2022).

Approaches for utilizing prior context have primarily focused on injecting text-based dialogue history into the prediction network (Hou et al., 2022) or using audio context from prior segments to train the encoder (Gómez Alanís et al., 2022; Schwarz et al., 2021). The closest work to ours is Narayanan et al. (2019), who show that carrying LSTM encoder states across utterances during training helps models generalize to long-form speech, serving as a workaround for the practical constraint of training on short utterances. In contrast, our methods do not require any training or fine-tuning. Instead, we approach state management as an inference-time decision problem, using streaming models already trained on long-form audio.

Among open-source streaming ASR models, the cache-aware FastConformer-RNNT (Noroozi et al., 2024) has become the go-to architecture for sub-500 ms latency applications. Streaming is enabled via cache-aware attention, where each encoder frame attends to a fixed left-context window of L frames. These key-value (KV) pairs are stored in a rolling cache alongside a configurable right-context of R lookahead frames. In addition, each encoder layer maintains a localized convolution cache to mitigate potential artifacts at chunk boundaries. The decoder follows the RNN-Transducer formulation (Graves, 2012), wherein the prediction network is a single-layer LSTM operating over the sequence of previously emitted non-blank tokens with its hidden and cell states acting as the decoder's linguistic memory (Noroozi et al., 2024).

3 Study Design

This section presents the experimental setup, including the streaming ASR models, datasets, and evaluation protocol. It also formalizes the proposed state-management strategies, State Carry-over and State Rollback, along with the baseline methods used for comparison.

3.1 Models

We evaluate the following pre-trained streaming ASR models: `fastconformer-114m` (Noroozi et al., 2024) and `nemotron-streaming-0.6b` (NVIDIA, 2024). Both models share a common cache-aware FastConformer-RNNT architecture, with model state given by a set of six components distributed between the encoder and the decoder. The FastConformer state comprises the encoder attention KV cache, a per-layer convolution cache, and the corresponding valid cache length. Similarly, the RNNT state consists of the accumulated decoder hypothesis, the last predicted token, and the number of tokens emitted. Further architectural details are provided in Appendix A.2.

3.2 Datasets

We evaluate our framework on the test sets of two standard telephonic dialogue benchmarks: CallHome (Canavan et al., 1997) and Switchboard (Godfrey et al., 1992; Deshmukh et al., 1998) (test split as in Romana et al. (2024)). For both datasets, we use individual (unmixed) channels, which contain backchannels, extended periods of silence, and occasional crosstalk. The datasets provide utterance-level timestamps for each speaker. Consecutive utterances from the same speaker separated by less than 1 s of silence are treated as a single turn, consistent with end-of-turn detection heuristics employed by commercial voice agents (Hamming AI, 2026). Both models are pre-trained on the Switchboard training set and have not been exposed to CallHome during training (Noroozi et al., 2024; NVIDIA, 2024). Additional dataset details are provided in Appendix A.3.

3.3 State Management

We implement and compare the four state management strategies visualized in Figure 1:

- *Fresh Start (FS):* The model's internal state is fully reset at the start of each turn, causing each utterance to be decoded independently.
- *Continuous State (CS):* Streaming proceeds continuously without any manipulation. The acoustic memory always consists of the last L frames from the speaker channel, regardless of content.
- *State Carry-over (SC):* The state at the end of a turn is utilized as the starting state for the next one, allowing the model to retain context from previous turns while bypassing long silences.
- *State Rollback (RB):* An extension of state carry-over in which the model rolls back to the state saved at the end of the last substantial turn when the previous turn is too short ($< k$ *tokens*) in order to drop backchannel context.

Algorithm 1 formalizes the three state management strategies (FS, SC, RB) as a single state-update rule over speaker turns. CS, which streams continuously through inter-turn silence without any manipulation, is excluded from the algorithm.

Algorithm 1 State Management Strategies for Streaming ASR

Require: Audio stream with turns $\{u_1, \ldots, u_T\}$; mode $m \in \{FS, SC, RB\}$; threshold k (for RB)
Ensure: Hypotheses $\{h_1, \ldots, h_T\}$, where h_i is the decoded token sequence for turn u_i and $|h_i|$ is its token length

$S \leftarrow \varnothing$ ▷ Model state
$S^* \leftarrow \varnothing$ ▷ Last substantial-turn state (RB)
for $i = 1$ to T **do**
  $h_i, S_i \leftarrow \textsc{Infer}(u_i, S)$
  if $m = FS$ **then**
    $S \leftarrow \varnothing$ ▷ State reset
  else if $m = SC$ **then**
    $S \leftarrow S_i$ ▷ State carry-over
  else if $m = RB$ **then**
    if $|h_i| \geq k$ **then**
      $S \leftarrow S_i$
      $S^* \leftarrow S_i$
    else
      $S \leftarrow S^*$ ▷ State rollback
    end if
  end if
end for
return $\{h_1, \ldots, h_T\}$

Most commercially deployed voice agents operate in a manner similar to FS, while for other streaming ASR applications, like live meeting captioning, streaming happens without any state manipulation like CS. As described in Section 2, both FS and CS have shortcomings, which the SC and RB strategies are designed to address.

3.4 Evaluation

We evaluate our state-management strategies using the standard Word Error Rate (WER) measure. For all experiments, the models perform greedy decoding with a batch size of 1, operating with a left-context window of $L = 70$ frames (5.6 s of history) and a right-context of $R = 1$ frame (160 ms latency), following the configuration described in Noroozi et al. (2024). This places the models within the latency requirements of real-time voice agents. For RB, we experiment with token thresholds $k \in \{1, 5, 10\}$, and present results using the hyperparameters that achieve the best validation-set performance: $k = 5$ for `fastconformer-114m` and $k = 10$ for `nemotron-streaming-0.6b`.

For the key experiments reported in Section 4, we use the ground-truth timestamps provided by each dataset to identify spoken regions and filter out tokens generated outside these intervals, such as those arising from crosstalk. To accommodate late token emissions that may occur past the utterance boundary, we extend each spoken region by 500 ms beyond the annotated end time for all our experiments. We further evaluate our strategies in a more realistic deployment setting using turn boundaries generated by Silero-VAD (Silero Team, 2021), a voice activity detection model widely used in commercial voice agents (Hamming AI, 2026). Implementation details including VAD hyperparameters are provided in Appendix A.7.

3.5 Memory Overhead

As described in Algorithm 1, SC incurs no additional memory overhead beyond standard streaming inference, as it simply carries forward the model state produced at the end of the previous turn. In contrast, RB needs to store an additional state (S^*) from the last substantial turn. For the default configuration described above, this amounts to approximately 2.6 MB for `fastconformer-114m` and 7.32 MB for `nemotron-streaming-0.6b` (see Appendix A.2). This corresponds to just 0.6% and 0.3% of the respective model sizes, making the additional memory overhead of RB negligible.

4 Results and Analysis

We first conduct a set of experiments to characterize the failure modes of different state-management strategies under varying conversational conditions. Specifically, we measure WER as a function of (i) the first n words in the turn (for turns with $\geq n$

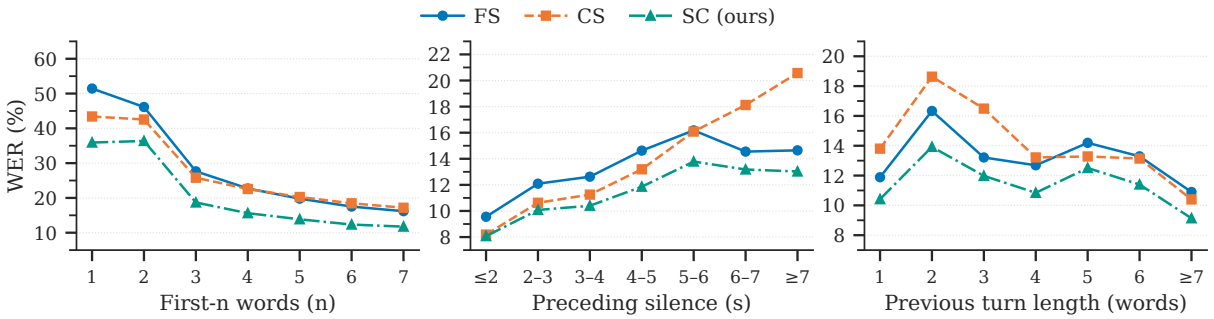

Figure 2: Word Error Rates for different state-management strategies with `fastconformer-114m` on Switchboard.

Model	Category	CallHome				Switchboard			
		FS	CS	SC	RB	FS	CS	SC	RB
`fastconformer-114m`	Overall	15.43	15.18	14.02	**13.83**	11.97	12.36	10.26	**10.20**
	First Word	48.10	41.40	36.24	**35.03**	51.44	44.32	35.92	**34.78**
	After Long Pause	17.17	20.66	16.15	**15.70**	14.73	20.06	13.08	**12.99**
	After Short Turn	18.52	20.20	17.17	**16.61**	13.93	15.94	12.03	**11.94**
`nemotron-streaming-0.6b`	Overall	13.82	14.29	13.86	**13.52**	10.90	11.40	10.85	**10.62**
	First Word	32.28	32.61	32.36	**27.85**	32.50	34.40	31.69	**29.46**
	After Long Pause	14.79	16.64	15.62	**14.68**	13.25	14.37	13.11	**12.83**
	After Short Turn	16.64	18.31	17.05	**15.66**	12.51	13.62	12.62	**12.15**

Table 1: Detailed WER (%) analysis. "First Word" refers to the WER of the initial word in a turn; "Long Pause" denotes silences exceeding the acoustic memory length of 5.6 s; "Short Turn" denotes turns shorter than 3 words.

words); (ii) the duration of silence preceding the turn; and (iii) the length of the preceding turn. The first analysis evaluates performance at turn onsets, while the latter two assess the robustness of the strategies to conversational noise.

Figure 2 illustrates the performance of the FS, CS, and SC strategies for these failure modes using `fastconformer-114m` on Switchboard. The first plot highlights that resetting the state results in significant performance degradation in the absence of any prior context for the first few words. The second plot shows how the preceding silence can degrade the state. While CS is generally better than FS for turns following shorter silences, CS performance degrades when the silence length exceeds the acoustic memory length of 5.6 s. The third plot suggests that short preceding turns, which often correspond to backchannels, degrade CS performance on the subsequent utterance relative to FS. In all contexts, the proposed SC approach outperforms existing baselines, demonstrating how state management through SC can effectively preserve relevant cross-utterance context.

Figure 3 shows the WER of SC compared to RB for utterances after turns of different lengths, for both models on Switchboard. For `fastconformer-114m`, RB and SC perform similarly. However, RB provides noticeably larger improvements for `nemotron-streaming-0.6b`, suggesting that its model state is more sensitive to shorter turns and backchannels.

All four strategies are compared in Table 1 for all model and dataset pairs. The key findings from the plots in Figure 2 are captured in the following WER metrics: i) *Overall:* the primary WER. ii) *First Word:* WER comparing the first word in the reference turn and the ASR hypothesis. iii) *After Long Pause:* WER of turns preceded by pauses ≥ 5.6 s. iv) *After Short Turn:* WER of utterances following short turns with ≤ 2 words (note that the vast majority of these short turns are backchannels).

Table 1 shows that RB achieves the lowest overall WER across all models and datasets, demonstrating the benefits of selectively preserving cross-utterance context. In contrast, naive CS performs worst in three of four settings, largely due to errors following long pauses and backchannels. While FS mitigates these effects by resetting state at each turn, it incurs a substantial turn-onset penalty, particularly for `fastconformer-114m`, where first-word WER degrades despite improvements in overall WER. Preserving prior context via SC provides clear gains for `fastconformer-114m`, but not for `nemotron-streaming-0.6b`, where utterances following backchannels often suffer degradation. RB addresses this issue by selectively discarding harmful context, consistently outperforming both FS and SC across all evaluated metrics.

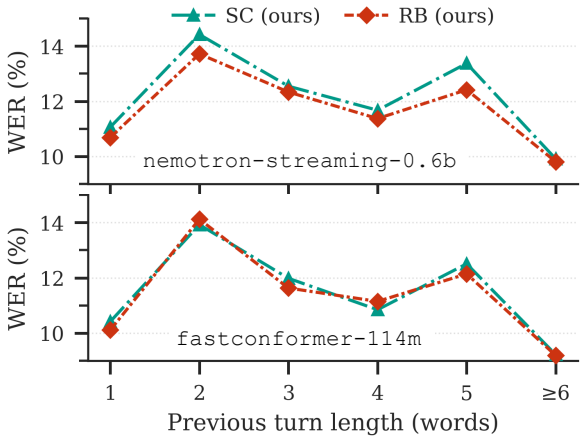

Figure 3: Comparison of SC and RB for Switchboard.

Statistical significance. To verify that the improvements from our methods are statistically significant, we compute "Overall WERs" for FS and RB and compare them using the MAPSSWE test, following standard practice in the ASR literature (Gillick and Cox, 1989). Our analysis reveals that, across all four model-dataset combinations, RB's improvement over FS is statistically significant, with $p < 10^{-3}$ for three setups and $p < 0.05$ for the remaining one (Appendix A.4). While the overall gains appear modest, this is expected: the benefits from our strategies are concentrated in the first few tokens of each turn and are averaged out over the complete utterances. The first-word WER results in Table 1 directly capture this effect.

Rollback threshold. We experiment with coarse thresholds $k \in \{1, 5, 10\}$ for RB, and present results with the best hyperparameters, i.e., $k = 5$ for `fastconformer-114m` and $k = 10$ for `nemotron-streaming-0.6b` in Table 1. Additional results with the other thresholds are provided in Appendix A.5. Based on these results, we observe that RB consistently outperforms SC across all tested values of k.

Variable latency. The performance trends of the state-management strategies remain largely unchanged even with different latencies, with RB consistently achieving the best results at all supported latencies below 250 ms (Appendix A.6). Reducing latency from 560 ms to 160 ms to satisfy voice-agent requirements results in only a modest degradation in performance. For `fastconformer-114m`, RB at 160 ms achieves a lower WER than FS at 560 ms. With RB, across both models, the WER increase relative to the best-performing strategy at 560 ms remains below 6%.

Without ground-truth timings. Finally, we evaluate our methods using turn boundaries generated by Silero-VAD (Silero Team, 2021), to better approximate real-world deployment conditions. The results (Appendix A.7) show that our strategies continue to outperform existing approaches without access to ground-truth timings, with the improvements being statistically significant. At 160 ms latency, RB achieves a 20.6% relative reduction in first-word WER over FS using VAD-obtained boundaries for Switchboard, closely matching the 20.9% reduction observed in Table 1. This analysis is restricted to the Switchboard dataset, as it requires word-level timing annotations for turn attribution, which are not available in CallHome.

To summarize, our analysis yields the following key findings:

- Although FS outperforms CS in most contexts, it consistently incurs a substantial penalty on the initial words at each turn.
- CS performance degrades as the preceding silence duration increases.
- SC substantially outperforms CS on both models and benchmarks. The gains from SC over FS are mainly for `fastconformer-114m`.
- RB is the best-performing state-management strategy across all evaluated models and datasets.
- These trends are consistent across all latencies below 250 ms and hold even with VAD-obtained turn boundaries for Switchboard.

5 Conclusions

This work provides a detailed empirical analysis of how long silences and backchannels degrade low-latency streaming ASR. We demonstrate that the state-resetting approach used by most voice agents discards useful cross-utterance context, causing disproportionate performance degradation at utterance onsets where errors are most costly for the downstream language model. To address this, we propose two simple state-management strategies, State Carry-over (SC) and State Rollback (RB), that preserve prior context across turns while remaining robust to conversational noise. Experiments across two streaming ASR models and two conversational benchmarks confirm that our strategies improve overall recognition accuracy, especially at turn onsets. These improvements can be particularly important for subsequent downstream processing by voice agents, since early ASR errors can propagate to and compound in downstream components.

6 Limitations

Our experiments are limited to benchmarks consisting primarily of American English telephonic conversations; whether the observed trends generalize to other languages and conversational settings remains an open question. The proposed rollback strategy relies on an empirically tuned threshold. While we demonstrate consistent gains across varied settings, adapting this threshold dynamically to conversational context may yield further improvements. Furthermore, this work focuses on a specific set of conversational failure modes: state resets, long pauses, and backchannels. It would be useful to extend the analysis to additional challenges, such as noisy acoustic conditions, code-switching, and multi-speaker settings. Finally, this work focuses on improving ASR performance rather than evaluating downstream voice-agent behavior. Although improved recognition may benefit subsequent language-model components, we do not directly measure its impact on downstream tasks such as intent prediction, dialogue quality, or task success. Investigating ASR error propagation in LLM-based voice-agent pipelines remains an important direction for future work.

Acknowledgments

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number 140D0424C0070. The United States Government is authorized to reproduce and distribute reprints for governmental purposes, notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

Alexandra Canavan, David Graff, and George Zipperlen. 1997. CALLHOME american english speech. Linguistic Data Consortium, Philadelphia. LDC97S42.

Yong-Seok Choi, Jeong-Uk Bang, and Seung Hi Kim. 2024. Joint streaming model for backchannel prediction and automatic speech recognition. *ETRI Journal*, 46(1):118–126.

Neeraj Deshmukh, Aravind Ganapathiraju, Andi Gleeson, Jonathan Hamaker, and Joseph W. Picone. 1998. Resegmentation of switchboard. *5th International Conference on Spoken Language Processing (ICSLP 1998)*.

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. Technical report.

Laurence Gillick and Stephen J. Cox. 1989. Some statistical issues in the comparison of speech recognition algorithms. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 532–535, Glasgow, Scotland. IEEE.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. Switchboard: telephone speech corpus for research and development. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ICASSP'92, page 517–520, USA. IEEE Computer Society.

Alejandro Gómez Alanís, Lukas Drude, Andreas Schwarz, Rupak Vignesh Swaminathan, and Simon Wiesler. 2022. Contextual-utterance training for automatic speech recognition. In *Proceedings of the IberSPEECH Conference*, pages 26–30.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Hamming AI. 2026. Voice AI Latency: What's Fast, What's Slow, and How to Fix It. `https://hamming.ai/resources/voice-ai-latency-whats-fast-whats-slow-how-to-fix-it`. Accessed: 2026.

Junfeng Hou, Jinkun Chen, Wanyu Li, Yufeng Tang, Jun Zhang, and Zejun Ma. 2022. Bring dialogue-context into RNN-T for streaming ASR. In *Proc. Interspeech*, pages 2048–2052.

Roger Hsiao, Dogan Can, Tim Ng, Ruchir Travadi, and Arnab Ghoshal. 2020. Online automatic speech recognition with listen, attend and spell model. *IEEE Signal Processing Letters*, 27:1889–1893.

W. Ronny Huang, Shuo-Yiin Chang, David Rybach, Rohit Prabhavalkar, Tara N. Sainath, Cyril Allauzen, Cal Peyser, and Zhiyun Lu. 2022. E2E segmenter: Joint segmenting and decoding for long-form ASR. In *Proc. Interspeech*, pages 4995–4999.

Dominik Macháček, Raj Dabre, and Ondřej Bojar. 2023. Turning whisper into real-time transcription system. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 17–24, Bali, Indonesia. Association for Computational Linguistics.

Yue Ming, Boyang Lyu, and Zerui Li. 2023. Cast: Context-association architecture with simulated long-utterance training for mandarin speech recognition. *Speech Communication*, 155:102985.

Arun Narayanan, Rohit Prabhavalkar, Chung-Cheng Chiu, David Rybach, Tara N. Sainath, and Trevor Strohman. 2019. Recognizing long-form speech using streaming end-to-end models. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 920–927.

Vahid Noroozi, Somshubra Majumdar, Ankur Kumar, Jagadeesh Balam, and Boris Ginsburg. 2024. Stateful conformer with cache-based inference for streaming automatic speech recognition. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12041–12045.

NVIDIA. 2024. Nemotron-Speech Streaming EN 0.6B. `https://huggingface.co/nvidia/nemotron-speech-streaming-en-0.6b`. Accessed: 2026-05-20.

Desh Raj, Liang Lu, Zhuo Chen, Yashesh Gaur, and Jinyu Li. 2022. Continuous streaming multi-talker asr with dual-path transducers. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7317–7321.

Dima Rekesh, Nithin Rao Koluguri, Samuel Kriman, Somshubra Majumdar, Vahid Noroozi, He Huang, Oleksii Hrinchuk, Krishna Puvvada, Ankur Kumar, Jagadeesh Balam, and Boris Ginsburg. 2023. Fast Conformer with linearly scalable attention for efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Amrit Romana, Kazuhito Koishida, and Emily Mower Provost. 2024. Automatic disfluency detection from untranscribed speech. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 32:4727–4740.

Andreas Schwarz, Ilya Sklyar, and Simon Wiesler. 2021. Improving rnn-t asr accuracy using context audio. In *Proc. Interspeech 2021*, pages 1792–1796.

Silero Team. 2021. Silero VAD: Pre-trained enterprise-grade voice activity detector, number detector and language classifier. `https://github.com/snakers4/silero-vad`.

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.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *Proceedings ICLR*.

A Appendix

A.1 Inference Setup

All experiments were conducted using Python 3.10.19 and the NeMo ASR toolkit version 2.7.0, with all model checkpoints obtained from Hugging Face. Inference was performed on NVIDIA L40 GPUs. To emulate real-world streaming deployment conditions, all evaluations were carried out independently with a batch size of 1. Both streaming ASR models employ deterministic greedy decoding during inference; consequently, all reported results are obtained from a single run.

A.2 Architecture Details

Both models operate on 16 kHz audio and produce subword token sequences. The smaller `fastconformer-114m` model consists of a 17-layer FastConformer encoder with a hidden dimension of 512, resulting in approximately 114M parameters and a model size of 435 MB. The model is trained using a hybrid RNNT-CTC objective on the datasets described in Noroozi et al. (2024); throughout our experiments, we use the RNNT decoder for inference. The `nemotron-streaming-0.6b` model employs a larger 24-layer FastConformer encoder with a hidden dimension of 1024, yielding approximately 0.6B parameters across the encoder and decoder and a memory footprint of approximately 2.25 GB. Unlike `fastconformer-114m`, it is trained exclusively with an RNNT objective on the data described in NVIDIA (2024).

Model state. For both the streaming models, the model state consists of a tuple of six components, which can be further divided into two groups: encoder (FastConformer) state components and decoder (RNNT) state components. The encoder state comprises two continuous cache tensors: (i) the attention KV cache (shape: `[n_layers, batch_size, left_context_len, hidden_dim]`) and (ii) the convolution cache (shape: `[n_layers, batch_size, hidden_dim, conv_subsample]`) for cache-aware streaming, plus (iii) a small counter tracking valid cache length. The decoder state holds (iv) the RNNT predictor's recurrent LSTM states, (v) its last output token, and (vi) the decoded length for the token sequence emitted so far. Table 2 provides the shapes and memory footprint for each of the state components for both the streaming models.

		fastconformer-114m		nemotron-streaming-0.6b	
		Shape	Memory	Shape	Memory
Encoder State	Attention KV Cache	$(17, 1, 70, 512)$	2.32 MB	$(24, 1, 70, 1024)$	6.56 MB
	Convolution Cache	$(17, 1, 512, 8)$	0.27 MB	$(24, 1, 1024, 8)$	0.75 MB
	Valid Cache Length	$(1,)$	8 B	$(1,)$	8 B
Decoder State	LSTM Recurrent State	$(3, 640)$	7.5 KB	$(10, 640)$	25 KB
	Last Token	scalar	8 B	scalar	8 B
	Decoded Length	scalar	8 B	scalar	8 B
Total			2.59 MB		7.32 MB

Table 2: Model state components with the corresponding tensor shapes and memory footprint.

Dataset	**Model**	**Δ% (FS $\rightarrow$ RB)**	**MAPSSWE**	**Wilcoxon**
Switchboard	nemotron-streaming-0.6b	2.47	**	**
	fastconformer-114m	14.36	**	**
CallHome	nemotron-streaming-0.6b	2.17	*	**
	fastconformer-114m	9.91	**	**

Table 3: Relative WER reduction (%) from FS to RB; statistical significance is assessed using the MAPSSWE and per-utterance Wilcoxon signed-rank tests. * indicates $p < 0.05$ and ** indicates $p < 0.001$.

A.3 Dataset Statistics

Switchboard (Godfrey et al., 1992; Deshmukh et al., 1998) and CallHome (Canavan et al., 1997) are two of the most commonly used telephonic dialogue benchmarks. Switchboard consists of telephone conversations between strangers discussing assigned topics, resulting in relatively structured dialogues with clearer turn-taking behavior. In contrast, CallHome contains conversations between friends and family members, making the speech more spontaneous, informal, and conversationally dense, with more frequent backchannels, interruptions, and topic shifts. Consequently, CallHome is generally considered the more challenging benchmark for ASR systems and provides a useful testbed for evaluating robustness to natural conversational phenomena.

To approximate the end-of-turn detection behavior employed in commercial voice agents (Hamming AI, 2026), we combine consecutive utterances from the same speaker separated by less than 1 s of silence and treat them as a single conversational turn. The CallHome test set comprises 40 channels containing 1,335 turns, whereas the Switchboard test set (Romana et al., 2024) contains 214 channels and 5,822 turns; each channel corresponds to approximately 5 minutes of conversational audio. The average turn length is 14.2 words for the CallHome corpus and 16.5 words for the Switchboard corpus.

A.4 Statistical Significance Testing

We assess the statistical significance of the difference between FS and RB in *Overall WER* using the Matched-Pair Sentence-Segment Word Error (MAPSSWE) test, with the Wilcoxon signed-rank test on utterance-level WER differences serving as an additional confirmation (Gillick and Cox, 1989). MAPSSWE pools error counts across all segments, with utterances weighted by their number of reference words. In contrast, the Wilcoxon test ranks signed per-utterance WER differences, assigning equal weight to each turn. As shown in Table 3, improvements from our methods are statistically significant under both tests, with $p < 10^{-3}$ for all but one result. The results for nemotron-streaming-0.6b with the CallHome dataset are statistically significant with $p < 0.05$ for the MAPSSWE test and $p < 10^{-3}$ for the per-utterance Wilcoxon test. Notably, the Wilcoxon test, which considers individual utterance-level WER improvements, consistently yields higher statistical significance than the MAPSSWE test.

Finally, our primary contribution is improving recognition at turn onsets rather than overall utterance-level WER. We therefore view these significance tests as secondary evidence that our state-management strategies improve, rather than degrade, overall transcription quality, while the turn-onset analyses presented in Section 4 provide the primary evidence for our contribution.

Model	Dataset	SC	RB ($k = 1$)	RB ($k = 5$)	RB ($k = 10$)
`fastconformer-114m`	CallHome (val)	13.94	13.82	**13.70**	13.74
	Switchboard (val)	9.95	9.90	**9.87**	9.91
	CallHome (test)	14.02	13.92	**13.83**	13.90
	Switchboard (test)	10.26	10.21	**10.20**	10.25
`nemotron-streaming-0.6b`	CallHome (val)	13.83	13.75	13.66	**13.57**
	Switchboard (val)	10.37	10.30	10.05	**10.02**
	CallHome (test)	13.86	13.71	13.62	**13.52**
	Switchboard (test)	10.85	10.80	10.68	**10.62**

Table 4: Comparison of overall WER (%) across different token thresholds for RB.

Model	**Latency**	**CallHome**				**Switchboard**			
		FS	CS	SC	RB	FS	CS	SC	RB
`fastconformer-114m`	80 ms	15.52	15.64	14.47	**14.19**	12.21	12.81	10.57	**10.49**
	160 ms	15.43	15.18	14.02	**13.83**	11.97	12.36	10.26	**10.20**
	240 ms	15.45	15.37	14.06	**13.98**	12.04	12.66	10.28	**10.27**
	560 ms	14.75	15.04	13.49	**13.44**	11.44	12.06	**10.04**	**10.04**
`nemotron-streaming-0.6b`	80 ms	14.51	14.86	14.35	**14.16**	11.44	11.88	11.5	**11.17**
	160 ms	13.82	14.29	13.86	**13.52**	10.90	11.40	10.85	**10.62**
	240 ms	13.97	14.67	13.83	**13.47**	10.97	11.61	11.01	**10.73**
	560 ms	**12.82**	14.15	13.77	13.33	**10.08**	11.19	10.86	10.67

Table 5: Comparison of overall WER (%) across different state-management strategies and latency settings.

A.5 Rollback Threshold

The State Rollback (RB) strategy described in Section 3.3 uses an empirically tuned threshold k. Specifically, if the preceding turn contains less than k tokens, the model rolls back to the state saved after the most recent substantial utterance, defined as one for which the ASR model emitted at least k tokens. This allows RB to discard context from short turns, such as backchannels, while preserving context from longer utterances. In this study, we evaluate coarse thresholds $k \in \{1, 5, 10\}$ and select the configuration with the best overall validation WER for the results presented in Section 4. As shown in Table 4, this corresponds to $k = 5$ for `fastconformer-114m` and $k = 10$ for `nemotron-streaming-0.6b`. The same optimal thresholds are observed across both datasets and their respective validation and test splits, suggesting that an appropriate value of k can be selected through validation-set tuning for real-world deployment. These results are consistent with our earlier observation that `nemotron-streaming-0.6b` is more sensitive to performance degradation following short turns and backchannels under SC, motivating a higher rollback threshold. Finally, we observe that RB consistently outperforms SC across all settings for all the evaluated thresholds.

A.6 Latency Analysis

Both streaming ASR models are trained with a left-context window of $L = 70$ frames, corresponding to 5.6 s of acoustic memory, and lookahead values of $R \in \{0, 1, 6, 13\}$, corresponding to latencies of 80 ms, 160 ms, 560 ms, and 1120 ms, respectively (NVIDIA, 2024). Although inference is possible with arbitrary positive values of $\{L, R\}$, we observe the best performance when using the configurations seen during training. All the results mentioned in Section 4 use $L = 70, R = 1$, which is the best-performing model training configuration for conversational voice agent applications that need to transcribe with a latency budget below 250 ms (Hamming AI, 2026). To evaluate the robustness of our state-management strategies to latency constraints, we test them at target latencies of 80 ms and 240 ms. Table 5 reports the performance of all strategies across latency settings for both models with ground-truth timings. We additionally include results at 560 ms, a latency setting used during model training. While unsuitable for real-time voice agents, this setting serves as a useful reference for understanding the impact of increased lookahead on performance.

The results in Table 5 show that the trends observed at 160 ms latency remain consistent across

Model	Category	FS	CS	SC	RB
`fastconformer-114m`	Overall	14.71	15.98	13.84	**13.59**
	First Word	49.15	45.92	35.05	**32.50**
	After Long Pause	18.17	23.33	17.13	**16.97**
	After Short Turn	16.61	19.41	15.63	**15.39**
`nemotron-streaming-0.6b`	Overall	14.79	15.74	14.66	**14.28**
	First Word	33.15	37.72	32.95	**30.72**
	After Long Pause	17.71	19.68	17.75	**17.19**
	After Short Turn	16.58	18.71	16.92	**16.01**

Table 6: Performance comparison of different state-management strategies on Switchboard with 160 ms latency when turn boundaries are derived from Silero-VAD. Metric definitions are identical to those used in Table 1.

other latency settings, with RB consistently outperforming the alternative strategies across all models, datasets, and voice-agent-compatible latencies. Comparison with the reference reveals that reducing the latency from 560 ms to 160 ms results in less than a 6% increase in overall WER for both models under RB. For `fastconformer-114m`, overall WER for RB with a latency budget of 160 ms is actually lower than that of FS with 560 ms latency for both datasets. The larger context window available at 560 ms latency reduces the need for prior-turn contextual information, allowing FS to outperform the other state-management strategies for `nemotron-streaming-0.6b`.

A.7 VAD-derived Turn Boundaries

To mimic realistic deployment settings, we use Silero-VAD to detect the turn onsets and ends. Consistent with the ground-truth timing setup, the VAD uses a minimum pause threshold of 1 s and extends the turn ends by 500 ms to accommodate delayed token emissions. Any tokens produced outside the resulting speech regions are excluded from evaluation. Most Silero-VAD parameters were kept at their default settings (Silero Team, 2021), with some modification: the minimum turn length was reduced to 100 ms to better capture short turns.

To reduce crosstalk in the VAD-derived speech regions, we compare speech segments across opposite channels and remove long overlapping regions (>500 ms) from the channel with lower signal energy. This heuristic effectively filters out crosstalk while preserving short backchannels and genuine overlaps. With these settings, VAD detects speech activity covering approximately 98.8% of the words in Switchboard. We further observe that the VAD-derived boundaries are slightly more permissive than the ground-truth annotations, increasing the average detected speech duration from 139 s to 147 s per channel, potentially exposing the ASR model to residual crosstalk and telephony-channel noise despite our filtering procedure.

Under this setup, each VAD-detected speech segment is treated as a separate turn. Each reference word is attributed to the turn containing its temporal midpoint, with words falling outside all detected segments assigned to the nearest turn. For WER calculation, we consider only turns containing at least one attributed reference word. This analysis is limited to Switchboard, as CallHome does not provide the word-level timing annotations required for turn attribution (Canavan et al., 1997).

The results for Switchboard with VAD timings for a setup with 160 ms latency are provided in Table 6. The overall WER with VAD-derived timings is consistently higher than with ground-truth timings. This degradation is likely caused by a combination of residual crosstalk leakage as well as missed words and speech segments resulting from imperfect VAD detection. The overall trends are quite similar to the experiments using ground-truth timings, with our strategies outperforming the existing methods for both the models.

A.8 Use of Existing Artifacts

This work exclusively uses publicly available datasets and open-source models. The artifacts were employed in a manner consistent with their intended use. To support reproducibility, we have released the software and implementation details associated with this work.

A.9 AI Assistant Usage

The authors used generative AI tools to assist with grammar correction and improving the readability of the manuscript. The AI tools were not used to generate research ideas, conduct experiments, analyze results, or draw scientific conclusions. All technical content and final manuscript revisions were reviewed and approved by the authors.