PSLM: Parallel Generation of Text and Speech with LLMs for Low-Latency Spoken Dialogue Systems

Kentaro Mitsui, Koh Mitsuda, Toshiaki Wakatsuki, Yukiya Hono, Kei Sawada

rinna Co., Ltd., Tokyo, Japan
{kemits,kohmi,towaka,yuhono,keisawada}@rinna.co.jp

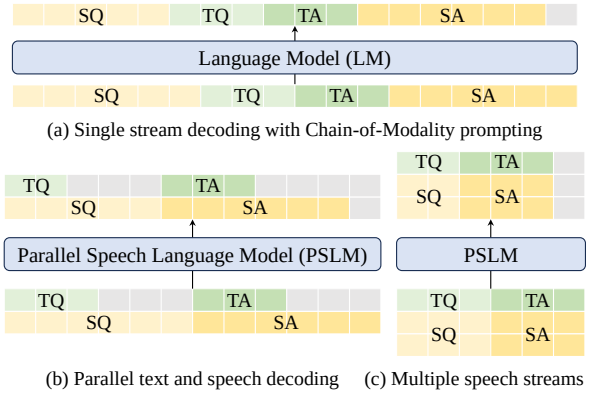
Abstract

Multimodal language models that process both text and speech have a potential for applications in spoken dialogue systems. However, current models face two major challenges in response generation latency: (1) generating a spoken response requires the prior generation of a written response, and (2) speech sequences are significantly longer than text sequences. This study addresses these issues by extending the input and output sequences of the language model to support the parallel generation of text and speech. Our experiments on spoken question answering tasks demonstrate that our approach improves latency while maintaining the quality of response content. Additionally, we show that latency can be further reduced by generating speech in multiple sequences. Demo samples are available at https://rinnakk.github.io/research/publications/PSLM.

1 Introduction

Spoken dialogue systems have been developed for many years to achieve natural human-computer interaction (McTear, 2002; Jokinen and McTear, 2009; Chen et al., 2017). Traditionally, these systems consist of several components: Automatic Speech Recognition (ASR), Response Generation (RG), and Text-to-Speech (TTS). Various methods for RG have been proposed with the advancements in Large Language Models (LLMs) (Wang et al., 2023a; Yi et al., 2024). More recently, the application of LLMs to ASR (e.g., Wang et al. 2023b; Hono et al. 2024; Fathullah et al. 2024) and TTS (Wang et al., 2023b; Hao et al., 2023) has attracted much attention, leading to the development of multimodal LLMs capable of end-to-end spoken language communication (Zhang et al., 2023; Nachmani et al., 2024).

Zhang et al. (2023) proposed SpeechGPT, an LLM that receives speech questions (SQ) as speech tokens, which are discrete representations extracted from raw waveforms, and sequentially generates text questions (TQ), text answers (TA), and speech answers (SA). Figure 1 (a) illustrates their approach called Chain-of-Modality (CoM) prompting. Spectron (Nachmani et al., 2024) follows this prompting style but directly handles speech spectrograms. Although these methods can generate high-quality responses, they face two major challenges in terms of response latency. First, generating SA requires the prior generation of TQ and TA. Second, speech sequences are much longer than text sequences[1].



(a) Single stream decoding with Chain-of-Modality prompting

(b) Parallel text and speech decoding (c) Multiple speech streams

Figure 1: (a) Chain-of-Modality prompting necessitates generating text questions (TQ) and text answers (TA) from speech questions (SQ) before producing speech answers (SA). (b) Our Parallel Speech Language Model (PSLM) enables the parallel decoding of TA and SA, reducing overall latency. (c) Introducing multiple speech streams further accelerates the generation of SA.

In this study, we propose Parallel Speech Language Model (PSLM), an LLM with multiple input-output sequences to handle both text and speech tokens, enabling their parallel generation. To emphasize their parallel processing capabilities, we will refer to these sequences as "streams". As described in Figure 1 (b), PSLM begins to generate SA immediately after the end of SQ tokens,

[1]Actual sequence lengths are provided in Appendix A.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 2692–2700

which can reduce overall latency. This leads to our first research question (**RQ1**): Can PSLM improve latency while maintaining the response quality achieved by CoM prompting? Additionally, we address the second challenge by introducing multiple speech streams to decode multiple speech tokens in a single step, as described in Figure 1 (c). This brings us to the second research question (**RQ2**): Do multiple speech streams sacrifice the response quality? Addressing these questions will pave the way for more advanced and responsive applications of spoken dialogue systems.

2 PSLM

2.1 Speech Discretization

Speech Tokenization Extracting discrete speech tokens from raw waveforms enables language models to handle speech in the same manner as text tokens. Self-supervised learning has been widely used for speech tokenization due to its ability to extract spoken content from raw waveforms (e.g., Rubenstein et al. 2023; Chou et al. 2023; Hassid et al. 2023). Following Zhang et al. (2023), we employ Hidden-Unit BERT (HuBERT) (Hsu et al., 2021) for speech tokenization.

Speech Detokenization In contrast to text tokenization, which is uniquely recoverable, speech tokenization largely discards the information of raw waveforms. Two major approaches have been proposed to solve this problem. The first approach uses a neural vocoder for directly reconstructing raw waveforms from speech tokens (e.g., Zhang et al. 2023; Chou et al. 2023; Hassid et al. 2023). The second approach uses a pretrained neural audio codec, which requires an additional module to predict the codec's tokens (e.g., Rubenstein et al. 2023; Zhang et al. 2024). We adopt the first approach to reduce overall latency using HiFi-GAN (Kong et al., 2020), a non-autoregressive neural vocoder that efficiently generates high-fidelity waveforms.

2.2 Integrating LMs with a Speech Stream

PSLM is built on top of a pretrained decoder-only Transformer (Vaswani et al., 2017). An overview of the PSLM architecture is provided in Figure 2. We add new input embedding and output projection layers to process speech tokens, while the structure of the intermediate Transformer layers remains unchanged. The embeddings of text and speech tokens are summed before being fed to the Transformer layers. The hidden features from the final Transformer layer are passed to two output projection layers to calculate the logits of the next text and speech tokens. We randomly initialize the weights of new embedding and projection layers.

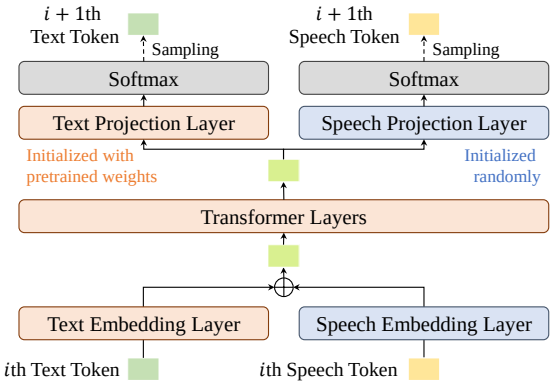


Figure 2: Architecture of PSLM.

A challenge of joint text-speech modeling lies in the mismatch in their lengths. In this study, we simply right-pad TQ and TA sequences with special `[TEXT-PAD]` tokens to align their lengths with those of the SQ and SA sequences, respectively. In a preliminary experiment on the CoM-based architecture, we attempted to generate text tokens and their corresponding speech tokens alternatively in a similar manner to ELLA-V (Song et al., 2024); however, this approach led to frequent mispronunciation. This is mainly because, in our case, the text is represented by tokens rather than phonemes; in some languages, the pronunciation of a character often changes according to subsequent characters, and a certain amount of lookahead is necessary to achieve accurate pronunciation. In contrast, our alignment strategy allows the model to focus on text token generation initially and then refer to the generated text when producing the majority of speech tokens, leading to more accurate pronunciation.

Our PSLM is trained by minimizing the sum of cross entropy losses for each stream. We include prompt tokens, comprising TQ and SQ, in the loss calculation. During inference, PSLM receives these prompt tokens and generates TA and SA in parallel. Text and speech tokens are sampled independently from their respective distributions.

2.3 Introducing Multiple Speech Streams

For further acceleration, we introduce multiple speech streams to PSLM. Assume that PSLM has $1+S$ streams, one for text tokens and S for speech tokens. Given the original speech token sequence of length N, the s-th speech stream consists of the

speech tokens with indices $s, s+S, s+2S, ..., s+MS$, where $s \in \{1, \ldots, S\}$ and $M = \lfloor N/S \rfloor - 1$. Compared to simply increasing the batch size, where the system's throughput improves but the latency for each instance remains unchanged, our approach reduces the sequence length handled by the Transformer layers to $1/S$, leading to an approximate S-fold speedup even in the single-instance scenario.

During training, simply summing the cross entropy losses for each stream makes the loss of text tokens less dominant, leading to poor text generation quality. Therefore, we introduce a weighted loss, where we multiply the loss for speech streams by $1/S$ to balance the weight of losses for text and speech streams.

2.4 Streaming Inference with HiFi-GAN

Following Chen et al. (2022), we use HiFi-GAN for streaming inference; specifically, we provide partial speech tokens to generate waveform fragments. In this study, we use non-causal convolution to maintain high speech quality. Therefore, the first speech fragment can be generated once $N_{\text{offset}} = \lfloor R/2 \rfloor + 1$ tokens are decoded, where R denotes the receptive field of HiFi-GAN. Implementation details can be found in Appendix B.

2.5 Overall Latency

We define latency as the delay between the end of the user's utterance and the system's initial response. The latency of conventional CoM-based systems L_{CoM} can be represented as follows:

$$L_{\text{CoM}} = D_{\text{s2t}} + D_{\text{SQ}} + \frac{N_{\text{dec}}}{P} + D_{\text{t2s}} \quad (1)$$

$$N_{\text{dec}} = N_{\text{TQ}} + N_{\text{TA}} + N_{\text{offset}} \quad (2)$$

where D_{s2t}, D_{SQ}, and D_{t2s} denote the delays of speech tokenization, the prefill phase in LMs, and speech detokenization, respectively; N_{TQ} and N_{TA} denote the number of tokens in TQ and TA, respectively; and P denotes the tokens per second (TPS) during the decode phase in LMs.

Our PSLM eliminates the need for generating TQ and TA beforehand, although it requires to run external ASR to obtain TQ. Hence, its latency L_{PSLM} can be represented as follows:

$$L_{\text{PSLM}} = D_{\text{ASR}} + D_{\text{SQ}} + \frac{N_{\text{offset}}}{P \cdot S} + D_{\text{t2s}} \quad (3)$$

where D_{ASR} denotes the ASR delay. Here D_{s2t} is omitted because speech tokenization can be performed in parallel with ASR.

3 Experimental Setup

3.1 Dataset

We used an internal dataset comprising 1.8M written QA pairs for training all models. Since some of these samples, which were primarily crawled from the internet, were deemed unsuitable for evaluation, we used a publicly available Japanese dataset (Hayashibe, 2023) for evaluation. This dataset was manually reviewed and consists of 669 diverse written QA pairs. We further filtered the evaluation set by excluding samples whose TQ or TA exceeded 140 characters, the maximum number of characters observed in the training set. The final evaluation set contained 396 samples. For both the training and evaluation sets, we constructed a spoken question answering (SQA) dataset by synthesizing SQ and SA using a well-trained single-speaker TTS system based on VITS (Kim et al., 2021).

3.2 Configuration

Tokenization and Detokenization For text tokenization, we used the tokenizer with a vocabulary size of 151,936 from rinna/nekomata-7b[2]. For speech tokenization, we applied k-means clustering with $k = 512$ to 12-th layer features from rinna/japanese-hubert-base[3] (Sawada et al., 2024), obtaining 50 speech tokens per second. For speech detokenization, we trained discrete unit-based HiFi-GAN (Polyak et al., 2021) using pairs of synthesized speech waveforms of SQ and SA and their corresponding speech tokens. For ASR, Whisper large-v3 (Radford et al., 2023) with faster-whisper[4] was used throughout our experiments.

Language Modeling We used rinna/nekomata-7b, a 32-layer 4096-hidden-size Transformer LM that was continuously pretrained from Qwen-7B (Bai et al., 2023) on Japanese text, as the backbone of our models. We implemented our models using the GPT-NeoX library (Andonian et al., 2023). Unless otherwise noted, models were trained for 50k steps with a batch size of 16 on 8 NVIDIA A100 GPUs using an Adam optimizer (Kingma and Ba, 2015) with a peak learning rate set to 1e-5. During inference, we set the temperature to 0.8 and applied top-k and top-p sampling with $k = 60$ and $p = 0.8$.

[2] https://huggingface.co/rinna/nekomata-7b

[3] https://huggingface.co/rinna/japanese-hubert-base

[4] https://github.com/SYSTRAN/faster-whisper

3.3 Baselines

We involved three CoM-based baselines, which share the model weights but differ in their prompts during decoding: (1) CoM-SQ receives only SQ, (2) CoM-ASR receives SQ and transcribed TQ, and (3) CoM receives SQ and gold TQ. In our preliminary experiments, the three-stage training (Zhang et al., 2023) was not effective in our configuration; thus, we trained the model using the same configuration as described in Section 3.2.

3.4 Evaluation Metrics

ChatGPT Scores We used OpenAI's GPT-3.5 Turbo API to evaluate response quality on a 5-point scale from 1 (bad) to 5 (excellent). The prompt is described in Appendix C. We report the scores for TA and the transcription of SA as T-score and S-score, respectively.

Character Error Rate (CER) We calculated the character error rate between the generated TA and the transcription of SA to assess their alignment.

Failure Rate (FR) We counted failure cases such as (1) no `[EOS]` token was generated before the total sequence length reached 2048, or (2) tokens were generated in the wrong modality, i.e., speech tokens in TQ and TA, or text tokens in SA.

Latency We simulated latency according to Equations 2 and 3 for each sample in the evaluation set, and reported the median values. We set $D_{\text{s2t}} = 0.05$, $D_{\text{SQ}} = 0.05$, $D_{\text{ASR}} = 0.2$, and $D_{\text{t2s}} = 0.01$ based on measurements taken on a single NVIDIA A100 GPU. For the TPS value P, the actual TPS varies depending on computing resources and optimization; 70 TPS was achieved with vLLM (Kwon et al., 2023) optimization, and 25 TPS without it. Meanwhile, for streaming inference with HiFi-GAN, LMs need to generate 50 speech tokens per second. Therefore, we set P to 50 in our simulations to match this requirement.

Human Rating We also conducted two subjective evaluations: one for text and the other for speech. In the text evaluation, we presented pairs of gold TQ and generated TA, and raters evaluated the naturalness of TA based on the same criteria used in the ChatGPT-based evaluation (Text Naturalness). In the speech evaluation, we presented gold SQ and generated SA successively, along with their TQ and TA, and asked the raters to evaluate (1) how natural the SA is as the speech of the TA (Speech Naturalness), and (2) whether the response is fast enough (Speed Score). For better reproducibility, we provide the actual instruction used for speech evaluation in Appendix D. The duration of silence between SQ and SA was simulated in the manner described in Section 2.5, except for the Ground Truth where the silence duration was set to 200ms, the average turn-taking gap in human conversation (Levinson and Torreira, 2015). Scores were rated on a 5-point scale. Fifty samples were randomly chosen from the evaluation set, and twenty in-house workers rated twenty samples each.

4 Results and Discussion

4.1 Automatic Evaluation

Comparison with Baselines To answer **RQ1**, we compared the proposed method in two conditions, PSLM and PSLM-ASR, with the baselines described in Section 3.3. PSLM receives SQ and gold TQ, while PSLM-ASR receives SQ and transcribed TQ. Table 1 summarizes the results. When gold TQ was given, PSLM achieved comparable scores to CoM and significantly improved latency. A similar trend was observed under more practical conditions where gold TQ was not available (PSLM-ASR vs. CoM-ASR). However, their scores were lower than those with gold TQ, and CoM-SQ faced greater degradation. These results suggest that ASR performance is crucial for response quality, and CoM-SQ seems to have produced more ASR errors than Whisper. Nevertheless, we conclude that PSLM maintains the response quality of CoM (**RQ1**). We also found that PSLM-based methods achieved lower FRs than CoM-based ones. Each stream of PSLM is dedicated to a single modality, which could have reduced the failures in generation. Furthermore, methods other than CoM-SQ marked lower CERs than Ground Truth. From this result, we confirmed that both CoM and PSLM can generate appropriate SA corresponding to TA.

Multiple Speech Streams To answer **RQ2**, we trained PSLM variants with two (-2x) or three (-3x) speech streams[5]. PSLM-2x achieved comparable scores to PSLM, whereas PSLM-3x demonstrated significant degradation. From these results, we conclude that speech tokens can be decoded in up to two streams without quality degradation (**RQ2**). An ablation study can be found in Appendix E.

[5]PSLM-3x was trained with a batch size of 4 due to the increased number of parameters.

Table 1: Automatic evaluation results. T-score and S-score represent the ChatGPT-based score for TA and transcribed SA, respectively. FR denotes the failure rate. Latency values in parentheses represent inputs involving gold TQ.

Method	Input modality	Output Modality	T-score↑	S-score↑	FR↓	CER↓	Latency [s]↓
Ground Truth	—	—	4.00±0.02	3.58±0.06	—	7.35	—
CoM	SQ → TQ (Gold)	TA → SA	3.50±0.09	3.27±0.09	12.12	6.28	(0.67)
PSLM	SQ, TQ (Gold)	TA, SA	3.50±0.08	3.22±0.09	5.05	5.25	(0.34)
CoM-SQ	SQ	TQ → TA → SA	3.12±0.11	2.94±0.10	15.91	7.83	1.03
CoM-ASR	SQ → TQ (ASR)	TA → SA	3.27±0.10	3.07±0.09	13.13	6.18	0.92
PSLM-ASR	SQ, TQ (ASR)	TA, SA	3.34±0.09	3.05±0.10	6.31	6.05	0.54
PSLM-2x	SQ, TQ (Gold)	TA, SA	3.50±0.08	3.20±0.09	4.29	6.39	(0.20)
PSLM-3x	SQ, TQ (Gold)	TA, SA	3.28±0.10	2.99±0.10	7.07	6.09	(0.15)

Table 2: Human evaluation results.

Method	Text↑	Speech↑	Speed↑
Ground Truth	4.08±0.26	3.74±0.19	4.73±0.11
CoM-SQ	2.44±0.29	4.04±0.20	4.07±0.23
CoM-ASR	2.90±0.30	3.94±0.20	4.17±0.22
PSLM-ASR	3.08±0.27	4.08±0.20	4.57±0.13

4.2 Human Evaluation

Considering practical applicability to SQA, we manually evaluated three methods: CoM-SQ, CoM-ASR, and PSLM-ASR, which do not rely on gold TQ, along with Ground-Truth. Table 2 shows the results. The text response naturalness of PSLM-ASR was comparable to CoM-ASR and higher than CoM-SQ, which is consistent with the automatic evaluation results. For speech naturalness, all methods achieved higher scores than Ground-Truth. This result can be attributed to two reasons: (1) SA of Ground-Truth are synthetic speech, which may include errors in pronunciation, intonation, and pauses, and (2) SA of Ground-Truth are typically longer than those of other methods, incurring that one or two unnatural parts lowered the entire score. Nevertheless, we confirmed that our approach can generate natural and faithful speech responses. For response speed evaluation, PSLM-ASR achieved a significantly higher score than CoM-ASR and CoM-SQ. This finding verifies that the proposed method reduces latency both numerically and perceptibly. Detailed analysis can be found in the next subsection.

4.3 Detailed Latency Analysis

The sequence length of TA, or N_{TA}, is the most influential factor in overall latency of CoM-based systems, as TA must be generated before SA. Thus, we investigated the overall latency by varying N_{TA}. Figure 3 shows the results. Due to the need for prior generation of TA, the latency of CoM-SQ and CoM-ASR increases linearly as TA length increases. In contrast, the latency of PSLM-ASR is constant because Equation 3 does not include N_{TA}, and PSLM-2x-ASR further reduces the latency. The gap between CoM-based and PSLM-based systems is remarkable when generating long TA, highlighting the effectiveness of generating text and speech tokens in parallel.

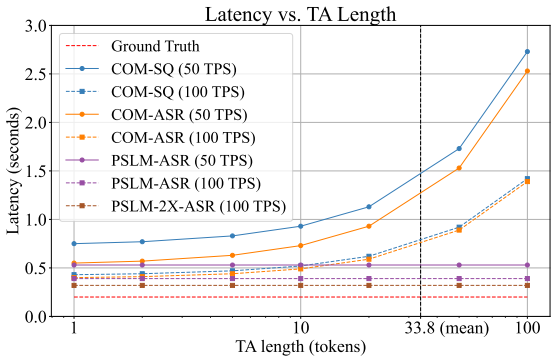


Figure 3: Latency vs. TA length for different methods and tokens per second (TPS). PSLM-2x-ASR (50 TPS) is omitted because its latency is identical to PSLM-ASR (100 TPS).

5 Conclusion

In this study, we proposed the Parallel Speech Language Model (PSLM), an LLM capable of generating text and speech tokens in parallel with multiple input-output streams, and investigated its impact on response quality and overall latency. The experimental evaluations on spoken question answering demonstrated that the proposed method significantly reduces latency compared to existing methods while maintaining response quality. Future work includes verifying the effectiveness of the proposed method on larger datasets and real speech data. Additionally, extending the proposed method to multi-turn dialogues is an important research direction.

6 Limitations

We recognize several limitations of this study. First, PSLM sacrifices ASR capability for faster response, requiring an external ASR module to serve as a spoken dialogue system. Although this dependency can complicate the system structure, it does not degrade the system's performance, provided that an appropriate ASR module is selected. This is supported by the fact that CoM-ASR outperformed CoM-SQ, as described in Section 4.1. Nevertheless, enabling ASR with the PSLM architecture can be an interesting research direction. Second, we used single-speaker synthetic speech for SQ and SA, which lacks diversity in several aspects of speech such as accent, rhythm, emotion, and timbre. Practical applications may require to accept voices of arbitrary speakers, which we will address in future work. Finally, multi-turn dialogue settings were not investigated in our experiments. While SpeechGPT (Zhang et al., 2023) was not applied to multi-turn dialogue due to sequence length limitations, our models with multiple speech streams have the potential to perform multi-turn dialogue.

References

Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Jason Phang, Shivanshu Purohit, Hailey Schoelkopf, Dashiell Stander, Tri Songz, Curt Tigges, Benjamin Thérien, Phil Wang, and Samuel Weinbach. 2023. GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *Computing Research Repository*, arxiv:2309.16609.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD Explorations Newsletter*, 19(2):25–35.

Ziyi Chen, Haoran Miao, and Pengyuan Zhang. 2022. Streaming non-autoregressive model for any-to-many voice conversion. *Computing Research Repository*, arXiv:2206.07288.

Ju-Chieh Chou, Chung-Ming Chien, Wei-Ning Hsu, Karen Livescu, Arun Babu, Alexis Conneau, Alexei Baevski, and Michael Auli. 2023. Toward joint language modeling for speech units and text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6582–6593, Singapore. Association for Computational Linguistics.

Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Junteng Jia, Yuan Shangguan, Ke Li, Jinxi Guo, Wenhan Xiong, Jay Mahadeokar, Ozlem Kalinli, Christian Fuegen, and Mike Seltzer. 2024. Prompting large language models with speech recognition abilities. In *Proceedings of the 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 13351–13355, Seoul, Korea.

Hongkun Hao, Long Zhou, Shujie Liu, Jinyu Li, Shujie Hu, Rui Wang, and Furu Wei. 2023. Boosting large language model for speech synthesis: An empirical study. *Computing Research Repository*, arXiv:2401.00246.

Michael Hassid, Tal Remez, Tu Anh Nguyen, Itai Gat, Alexis Conneau, Felix Kreuk, Jade Copet, Alexandre Defossez, Gabriel Synnaeve, Emmanuel Dupoux, Roy Schwartz, and Yossi Adi. 2023. Textually pretrained speech language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 63483–63501, New Orleans, LA, U.S.A.

Yuta Hayashibe. 2023. megagonlabs/instruction_ja: Japanese instructions data for LLM.

Yukiya Hono, Koh Mitsuda, Tianyu Zhao, Kentaro Mitsui, Toshiaki Wakatsuki, and Kei Sawada. 2024. Integrating pre-trained speech and language models for end-to-end speech recognition. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13289–13305, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Kristiina Jokinen and Michael McTear. 2009. *Spoken Dialogue Systems*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool publishers, U.S.A.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *Proceedings of the 38th International Conference on Machine Learning*, pages 5530–5540, online.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings*

of the 3rd International Conference on Learning Representations, San Diego, CA, U.S.A.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 17022–17033, online.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, New York, NY, U.S.A.

Stephen C Levinson and Francisco Torreira. 2015. Timing in turn-taking and its implications for processing models of language. *Frontiers in psychology*, 6.

Michael F. McTear. 2002. Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys*, 34(1):90–169.

Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich. 2024. Spoken question answering and speech continuation using spectrogram-powered LLM. In *In Proceedings of the 12th International Conference on Learning Representations*, Vienna, Austria.

Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. Speech resynthesis from discrete disentangled self-supervised representations. In *Proceedings of INTERSPEECH 2021*, pages 3615–3619, online.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, pages 28492–28518, Honolulu, Hawaii, U.S.A.

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, Hannah Muckenhirn, Dirk Padfield, James Qin, Danny Rozenberg, Tara Sainath, Johan Schalkwyk, Matt Sharifi, Michelle Tadmor Ramanovich, Marco Tagliasacchi, Alexandru Tudor, Mihajlo Velimirović, Damien Vincent, Jiahui Yu, Yongqiang Wang, Vicky Zayats, Neil Zeghidour, Yu Zhang, Zhishuai Zhang, Lukas Zilka, and Christian Frank. 2023. AudioPaLM: A large language model that can speak and listen. *Computing Research Repository*, arXiv:2306.12925.

Kei Sawada, Tianyu Zhao, Makoto Shing, Kentaro Mitsui, Akio Kaga, Yukiya Hono, Toshiaki Wakatsuki, and Koh Mitsuda. 2024. Release of pre-trained models for the Japanese language. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 13898–13905, Torino, Italia. ELRA and ICCL.

Yakun Song, Zhuo Chen, Xiaofei Wang, Ziyang Ma, and Xie Chen. 2024. ELLA-V: Stable neural codec language modeling with alignment-guided sequence reordering. *Computing Research Repository*, arXiv:2401.07333.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5998–6008, Long Beach, CA, U.S.A.

Hongru Wang, Lingzhi Wang, Yiming Du, Liang Chen, Jingyan Zhou, Yufei Wang, and Kam-Fai Wong. 2023a. A survey of the evolution of language model-based dialogue systems. *Computing Research Repository*, arxiv:2311.16789.

Tianrui Wang, Long Zhou, Ziqiang Zhang, Yu Wu, Shujie Liu, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Furu Wei. 2023b. VioLA: Unified codec language models for speech recognition, synthesis, and translation. *Computing Research Repository*, arXiv:2305.16107.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024. A survey on recent advances in llm-based multi-turn dialogue systems. *Computing Research Repository*, arxiv:2402.18013.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15757–15773, Singapore. Association for Computational Linguistics.

Dong Zhang, Xin Zhang, Jun Zhan, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2024. SpeechGPT-Gen: Scaling chain-of-information speech generation. *Computing Research Repository*, arXiv:2401.13527.

A Sequence Length Distributions

We calculated the sequence length distributions of SQ, TQ, TA, and SA in the training set. The results are listed in Table 3. On average, CoM prompting requires to generate 36.5 (TQ) + 33.8 (TA) $\approx$ 70 text tokens before generating SA. Eliminating the need for generating these tokens can greatly reduce overall latency. In addition, speech tokens are more than 11 times longer than text tokens, highlighting the need for efficient generation of speech tokens.

B Implementation Details of HiFi-GAN

The HiFi-GAN generator comprises convolution layers. Therefore, a waveform fragment corresponding to the i-th token depends only on tokens with indices $[i - \lfloor R/2 \rfloor, i + \lfloor R/2 \rfloor]$. This allows waveform generation to start before the entire SA is generated. As described in Figure 4, HiFi-GAN first generates a waveform fragment once the LM generates $N_{\text{offset}} = \lfloor R/2 \rfloor + 1$ tokens, then generates subsequent fragments by shifting input tokens one by one.

In our experiments, we trained HiFi-GAN to generate 24 kHz waveform from 50Hz tokens, which results in $R = 26$. Following Polyak et al. (2021), we embedded input speech tokens into 256-dimensional features and fed them to HiFi-GAN. We modified the upsampling rates to $[8, 6, 5, 2]$, the number of total iterations to 300k, and kept the other configuration the same as the original work (Kong et al., 2020).

C ChatGPT Evaluation Prompt

We used the prompt in Figure 5 for ChatGPT-based evaluation. The original prompt was written in Japanese, but a translated version is presented here.

D Speech Evaluation Instruction

We used the instruction in Figure 6 for speech evaluation. The original instruction was written in Japanese, but a translated version is presented here.

E Ablation Study

We trained three PSLM variants, one from scratch (-no-pretrain), one without TQ (-no-TQ), and one without SQ (-no-SQ). In addition, we trained PSLM-2x and PSLM-3x without weighted loss (-no-WL). Table 4 shows the automatic evaluation results. PSLM-no-pretrain exhibited significant degradation in all metrics, indicating the necessity of pretrained LM's text capability. PSLM-no-TQ also showed large degradation, highlighting the importance of TQ in response quality. In contrast, PSLM-no-SQ achieved comparable scores to PSLM. This result implies that the speech-specific information such as intonation, rhythm, and emotion is not essential in the current SQA task due to the use of synthetic speech. We also found that PSLM-2x-no-WL achieved almost comparable scores to PSLM, whereas PSLM-3x-no-WL showed significant degradation. From these results, we conclude that the weighted loss is especially effective as the number of speech streams increases.

Table 3: Sequence length distributions in the training set (in tokens).

	SQ	TQ	TA	SA
Mean	406.6	36.5	33.8	386.5
Min	34	2	1	27
25%	214	19	15	179
50%	354	32	29	340
75%	577	51	50	563
Max	1861	148	147	1697

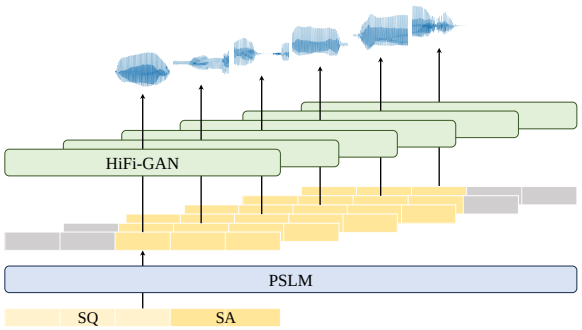


Figure 4: Streaming inference using HiFi-GAN with receptive field size $R = 5$ and SA length $N_{\text{SA}} = 6$. Waveform generation begins once $N_{\text{offset}} = \lfloor R/2 \rfloor + 1 = 3$ tokens are generated. Text tokens are omitted.

A conversation between two individuals will be provided. The conversation follows a format where one asks a question and the other responds. Based on the following evaluation criteria, rate the response quality on a scale from 1 (bad) to 5 (excellent).

[Evaluation Criteria]
1. Bad - The response is completely off-topic and difficult to understand.
2. Poor - The response is somewhat related to the question but contains grammatical or semantic errors, making it somewhat difficult to understand.
3. Fair - The response mostly aligns with the question, with few grammatical or semantic errors, and provides somewhat adequate information.
4. Good - The response aligns with the question, contains few grammatical or semantic errors, and provides adequate information.
5. Excellent - The response aligns with the question, contains almost no grammatical or semantic errors, provides adequate and appropriate information.

Output the evaluation score in the following format:
[Example Evaluation 1]
Question: Can you recommend an easy-to-write pen?
Answer: I recommend Mitsubishi Pencil Jetstream Standard.
Score: 5
[Example Evaluation 2]
Question: What is the highest mountain in the world?
Answer: I guess 3141010059.
Score: 1
[Evaluation Target]
Question: {Question}
Answer: {Answer}
Score:

Figure 5: Prompt for ChatGPT evaluation.

At the top of the screen, you will see the "Reference Text," and at the bottom of the screen, you will hear the audio of a conversation between two people. The conversation is in a QA format, with one person asking questions and the other responds. Please listen to the audio file with a headphone and evaluate it on a scale of 1 (poor) to 5 (good) based on the following two criteria:
(1) How natural is the system's response audio as a reading of the response text (the part after "A:" in the reference text)?
(2) Is the system's response speed sufficiently fast?
Please do not include the naturalness of the text content or the naturalness of the question audio in the score.

Figure 6: Instruction for speech evaluation.

Table 4: Ablation study. The suffix no-WL denotes weighted loss was not applied.

Method	**Input modality**	**Output Modality**	**T-score↑**	**S-score↑**	**FR↓**	**CER↓**
PSLM	SQ, TQ (Gold)	TA, SA	3.50±0.08	3.22±0.09	5.05	5.25
PSLM-2x	SQ, TQ (Gold)	TA, SA	3.50±0.08	3.20±0.09	4.29	6.39
PSLM-3x	SQ, TQ (Gold)	TA, SA	3.28±0.10	2.99±0.10	7.07	6.09
PSLM-no-pretrain	SQ, TQ (Gold)	TA, SA	2.22±0.07	2.12±0.07	18.18	10.13
PSLM-no-TQ	SQ	TA, SA	2.34±0.09	2.19±0.09	8.84	6.38
PSLM-no-SQ	TQ (Gold)	TA, SA	3.54±0.08	3.17±0.09	6.31	8.99
PSLM-2x-no-WL	SQ, TQ (Gold)	TA, SA	3.42±0.08	3.17±0.08	8.84	4.99
PSLM-3x-no-WL	SQ, TQ (Gold)	TA, SA	2.67±0.10	2.46±0.10	11.36	6.94