FillerSpeech: Towards Human-Like Text-to-Speech Synthesis with Filler Insertion and Filler Style Control

Seung-Bin Kim[1], Jun-Hyeok Cha[1], Hyung-Seok Oh[1], Heejin Choi[2], Seong-Whan Lee[1*]
[1]Korea University, [2]Samsung Research

Abstract

Recent advancements in speech synthesis have significantly improved the audio quality and pronunciation of synthesized speech. To further advance toward human-like conversational speech synthesis, this paper presents FillerSpeech, a novel speech synthesis framework that enables natural filler insertion and control over filler style. To address this, we construct a filler-inclusive speech data, derived from the open-source large-scale speech corpus. This data includes fillers with pitch and duration information. For the generation and style control of natural fillers, we propose a method that tokenizes the filler style and utilizes cross-attention with the input text. Furthermore, we introduce a large language model-based filler prediction method that enables natural insertion of fillers even when only text input is provided. The experimental results demonstrate that the constructed dataset is valid and that our proposed methods for filler style control and filler prediction are effective.

1 Introduction

Text-to-Speech (TTS) synthesis systems (Lee et al., 2022; Li et al., 2024; Peng et al., 2024; Wang et al., 2025) have seen remarkable advancements in recent years, particularly in achieving high-quality audio generation (Kim et al., 2024; Lee et al., 2025a,b) and natural pronunciation (Ju et al., 2024). These improvements have paved the way for applications in various domains, such as virtual assistants, audiobooks, and human-computer interaction. Despite these advancements, achieving human-like conversational speech remains an open challenge.

Fillers, such as "um", "uh", or "well", are an integral part of natural human conversation (Zhu et al., 2022; Dinkar et al., 2022). They serve various functions, including signaling hesitation, buying time for thought formulation, or maintaining the flow of dialogue. When these elements are missing from synthesized speech, the speech can sound unnatural, reducing its effectiveness in applications that require natural human interaction.

Previous research has attempted to address filler speech synthesis. (Éva Székely et al., 2019a) focused on training fillers as separate acoustic models to generate natural speech, while (Éva Székely et al., 2019b) learned fillers as tokens from a spontaneous conversational speech dataset. However, these models were limited to a narrow range of filler types (e.g., "uh" and "um"), limiting their ability to handle diverse speaking styles. (Yan et al., 2021) introduced an adaptive text-to-speech model to capture spontaneous speaking styles but did not explicitly focus on modeling fillers as non-verbal components of speech. (Fernandez et al., 2022) proposed a method to incorporate conversational style, including interjections, but struggled to generate and control fillers in a natural and seamless manner. (Wang et al., 2022) adopted a sampling-based approach for filler insertion but relied heavily on statistical methods, which often lacked coherence with the given textual context. These studies, while pioneering, highlighted challenges in modeling diverse filler styles and achieving precise text-based control for natural synthesis.

To tackle this challenge, we propose FillerSpeech, a novel framework for text-to-speech synthesis with filler insertion and filler style control. We first construct a filler-inclusive speech dataset derived from the large-scale LibriHeavy corpus (Kang et al., 2024) using an automated method to label fillers with pitch and duration information, thereby eliminating the need for manual annotation. To achieve speech synthesis with controllable filler style, we tokenize filler style attributes and use cross-attention to incorporate these word-level style tokens. Additionally, we integrate a pitch predictor into the text encoder to enhance the overall quality

[*]Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 34097–34114
November 4-9, 2025

of the synthesized speech and improve control over filler style. While filler selection can be performed manually, we additionally introduce a large language model (LLM)-based filler prediction method that automatically inserts fillers based solely on the input text. Experimental results validate the effectiveness of our approach, demonstrating that FillerSpeech synthesizes natural and controllable speech, thereby enhancing the realism of conversational speech applications. Our demo is available at `https://fillerspeech.github.io/main`.

2 Related Work

2.1 Flow Matching in Speech Synthesis

Flow matching (Lipman et al., 2023) has emerged as a powerful technique for speech synthesis, offering advantages in both output quality and efficiency compared to traditional diffusion-based approaches (Popov et al., 2021; Kim et al., 2022).

Several recent works have explored the application of flow matching to various aspects of speech synthesis. (Velugoti et al., 2023) builds upon the flow-matching framework by introducing a rectified flow approach that improves synthesis efficiency while maintaining high-quality audio generation. (Le et al., 2023) further advances the field by adopting a versatile, non-autoregressive approach. This model not only generates Mel-spectrograms but also supports speech inpainting and style transfer, showcasing robustness in both seen and unseen scenarios. (Kim et al., 2023) proposes a data-efficient zero-shot TTS method that leverages a speech-prompted text encoder combined with flow matching. (Mehta et al., 2024) leverages optimal-transport conditional flow matching (OT-CFM) to generate high-quality speech with only a few synthesis steps. More recently, (Wu et al., 2024) takes flow-matching-based synthesis a step further by incorporating dynamic emotional control, enabling the generation of speech with time-varying emotional expressions. Similarly, (Kanda et al., 2024) focuses on fine-grained emotional control, specifically targeting laughter synthesis. This approach offers a highly expressive and adaptable speech synthesis framework.

Flow matching enables the generation of diverse and natural outputs with high computational efficiency (Yun et al., 2025). Building on this property, we adopt flow matching to synthesize expressive, filler-inclusive speech with both high quality and fine-grained controllability.

2.2 Large Language Models

Large language models such as GPT-3 (Brown et al., 2020) and PaLM (Chowdhery et al., 2023) have demonstrated strong capabilities in reasoning and context modeling, which are essential for predicting fillers in conversational speech. However, these proprietary models are not publicly available for fine-tuning, limiting their direct applicability to our task.

The LLaMA family of models (Touvron et al., 2023a,b) provides open-source, efficient architectures pretrained on diverse corpora, balancing performance with accessibility for research. Building on this foundation, Vicuna (Chiang et al., 2023) enhances conversational ability through fine-tuning on dialogue datasets. Vicuna-7B, in particular, specializes in context-aware and coherent dialogue generation, making it well-suited for tasks that require inserting contextually appropriate fillers.

We therefore adopt Vicuna-7B as the backbone of our filler prediction module, leveraging its open-source availability, efficiency, strong conversational performance, and compatibility with parameter-efficient fine-tuning methods such as low-rank adaptation (LoRA) (Hu et al., 2022; Cha et al., 2025).

3 Dataset Construction

3.1 Filler-Inclusive Data Collection

To train FillerSpeech, we construct a dataset comprising speech samples, each of which contains at least one filler. Fillers can generally be categorized into lexical fillers (e.g., "like", "you know") and non-lexical fillers (e.g., "uh", "um"). In this work, we focus on non-lexical fillers, as they are more universally applicable and less dependent on linguistic context. Based on previous studies (Ward, 2006; Wang et al., 2022), we curate a list of common fillers: "ah", "aha", "eh", "ha", "hm", "huh", "oh", "uh", "um", "well", "yeah", "ya". Using a large-scale corpus of high-quality speech, we identify and extract speech samples that contain these fillers. This process enables us to build a comprehensive and diverse dataset tailored to the specific needs of filler-inclusive speech synthesis.

3.2 Style Labeling for Fillers

In addition to collecting filler-inclusive data, we also label each filler occurrence with style information to facilitate controllable generation. The style labels focus on two key attributes: pitch (F0)

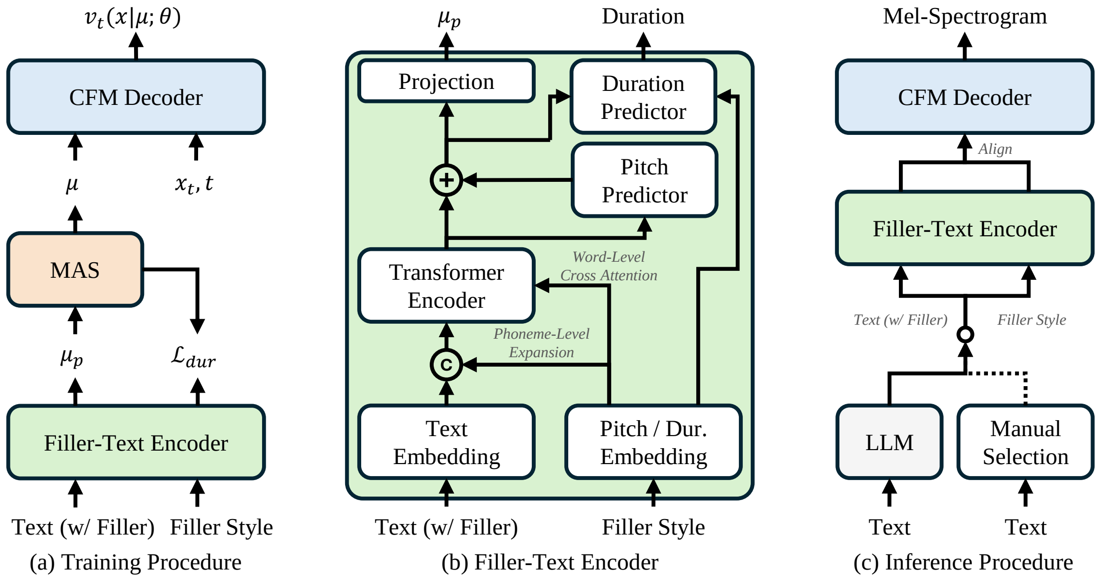


Figure 1: Overview of FillerSpeech. During inference, our method leverages a fine-tuned LLM to predict filler attributes from the input text, or alternatively, users can manually select the desired filler details.

and duration. To accurately identify the locations of fillers within the audio samples, we first align the speech with its corresponding text using an external aligner. This alignment step provides precise segmentation of fillers, which is essential for subsequent style labeling.

For pitch labeling, we first extract F0 values from the audio and then label each filler as high, medium, or low based on its average pitch. We compute these averages in two ways: one method calculates the average pitch for each filler type, and the other computes the average pitch of words within an utterance. Details are provided in Appendix D.1. Since pitch characteristics differ significantly by gender, we label male and female speakers separately. To determine the pitch height, we use semitone differences as a threshold. Specifically, we label a filler as high or low if its pitch deviates by more than four semitones from the reference.

For duration labeling, we calculate the average duration of each filler type and categorize each instance as long or short. We label fillers in the top 25% of the duration distribution as long, while those in the bottom 25% are labeled as short. By labeling both pitch and duration, our dataset captures the prosodic and temporal characteristics of fillers, providing detailed annotations for precise control in filler synthesis.

4 Method

We present a speech synthesis method that inserts fillers and enables control over their style. Our model leverages filler style tokens in conjunction with a pitch predictor and cross-attention to modulate the speaking style of synthesized speech. In addition to directly manipulating filler style, we propose a fine-tuning approach for LLMs to predict filler insertions along with their style attributes, thereby enabling filler-inclusive speech synthesis from text alone. Detailed descriptions of each component are provided in the following subsections.

4.1 Tokenization of Filler

To effectively incorporate fillers into speech synthesis, we adopt a phoneme-based tokenization approach for fillers rather than treating each filler word as an indivisible token. Since there are far fewer distinct filler words than phonemes, using phoneme-level tokenization ensures smooth integration of fillers with regular words in synthesized speech.

For filler style, we tokenize pitch and duration using discrete labels. Regular words, which lack pitch and duration labels, are assigned null labels. Since pitch is strongly correlated with speaker gender, we further tokenize pitch labels based on gender.

4.2 Filler Style Control

To condition the encoder on pitch of filler at the phoneme level, we first embed the pitch tokens, then expand them to match phoneme-level resolution, and finally concatenate them with the phoneme embeddings of the text. To further enhance the integration of fillers into synthesized speech, the encoder computes cross-attention between phoneme-level text representations and word-level pitch embeddings.

For duration control, we embed the duration tokens and use them together with the text representations as input to the duration predictor. To enable more precise filler style control, we explicitly incorporate pitch information during training via a dedicated pitch predictor that estimates appropriate pitch values from the text representation. By modeling pitch explicitly during training, the text encoder learns to better capture the prosodic characteristics necessary for natural filler generation and effective style control.

4.3 Prior Loss for Filler Representation

We compute a prior loss between the encoder outputs and the target Mel-spectrograms. Unlike conventional methods (Popov et al., 2021; Mehta et al., 2024) that compute prior loss on a sampled subset of encoder outputs for training efficiency, we compute the loss using all encoder outputs. Since fillers constitute only a small fraction of the text and style tokens appear exclusively in filler-containing segments, sampling could exclude them and thereby diminish filler style controllability. After computing the prior loss, we follow standard practice by sampling a subset of encoder outputs as input to the decoder to improve training efficiency.

4.4 Flow Matching Decoder

Our decoder is built on the flow matching framework, a generative diffusion model that employs optimal transport conditional flow matching for efficient and probabilistic data transformation. The flow matching process models a probability path that connects a simple prior distribution p_0, such as Gaussian noise, to a complex data distribution $q(x)$, such as a Mel-spectrogram. This is achieved by defining a vector field $\boldsymbol{v}_t(x)$ that governs the transformation of samples over time t through an ODE as follows:

$$\frac{d}{dt}\phi_t(x) = \boldsymbol{v}_t(\phi_t(\boldsymbol{x})); \qquad \phi_0(\boldsymbol{x}) = \boldsymbol{x}. \tag{1}$$

Here, $\phi_t(x)$ represents the trajectory of a sample from the prior distribution to the target distribution. In OT-CFM, the training objective minimizes the difference between the predicted vector field $\boldsymbol{v}_t(x)$ and the ideal vector field $\boldsymbol{u}_t(x)$ as follows:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,x_0,x_1}\|\boldsymbol{u}_t^{\mathrm{OT}}(\phi_t^{\mathrm{OT}}(\boldsymbol{x})|\boldsymbol{x}_1) - \boldsymbol{v}_t(\phi_t^{\mathrm{OT}}(\boldsymbol{x})|\boldsymbol{\mu};\theta)\|^2. \tag{2}$$

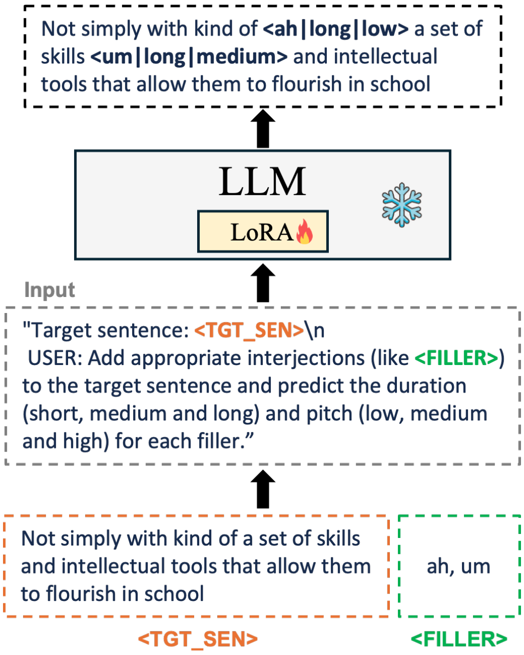


Figure 2: Overview of the LoRA fine-tuning for filler prediction in the LLM.

This formulation ensures that the decoder learns an efficient and smooth transformation from latent noise to Mel-spectrograms. The simplicity of the vector field $\boldsymbol{u}_t(x)$, which changes linearly along the trajectory, reduces the number of required synthesis steps compared to traditional diffusion models, thereby significantly improving speed and accuracy.

4.5 LLM-based Filler Prediction

To insert fillers naturally based on the input text, we propose an LLM-based filler prediction module. To leverage the reasoning capabilities of LLMs (Wei et al., 2022) while mitigating catastrophic forgetting, we fine-tune an LLM using a LoRA (Hu et al., 2022; Cha et al., 2025). Our method predicts both the position and type of each filler, as well as an appropriate duration and pitch for each, by considering the surrounding context in the input text.

To allow a single model to handle a variety of filler prediction scenarios, we design a set of instruction prompts with different levels of specification. These prompts cover cases where the filler type is specified; where both the filler type and position are given; where a set of potential filler candidates is provided; or where only the filler position is specified. In each scenario, the model is trained to predict the remaining characteristics such as duration, pitch, and type that are not provided.

The LLM is trained separately from the TTS model, and during speech synthesis inference, the LLM-based filler prediction can be utilized as needed. Detailed information on the prompt design is provided in Appendix A.

5 Experiments

5.1 Dataset

Using the method described in Section 3, we constructed a filler-inclusive speech dataset from the large-scale Libriheavy corpus, which comprises 50,000 hours of speech data. The resulting dataset comprises 4,460 speakers and a total of 2,116 hours of speech. Each sentence contains at least one filler, and each filler instance is annotated with pitch and duration information. For the validation and test sets, we selected 50 speakers and obtained 2,707 and 2,966 samples, respectively.

5.2 Implementation Details

5.2.1 Data Construction

In constructing the filler-inclusive dataset, we employed the Montreal Forced Aligner[1] (MFA) as the external aligner and Parselmouth[2] for pitch extraction.

5.2.2 Speech Synthesis

The flow-matching decoder in our model was implemented using a Transformer-based U-Net architecture, ensuring efficient and high-quality Mel-spectrogram generation. For speaker information extraction, we employed the style encoder from the Meta-StyleSpeech model (Min et al., 2021). We integrated the pitch predictor into our framework to achieve accurate and controllable style generation, adopting the structure proposed in (Ren et al., 2021). To compute the alignment between the encoder outputs and the Mel-spectrogram, we employed super monotonic alignment search (MAS) (Lee and Kim, 2024).

For training, we used two NVIDIA RTX A6000 GPUs with a batch size of 32 per GPU. The model was trained for one million steps (approximately 83 hours). Our model contains 60.11M parameters, and additional hyperparameter details are provided in Table 4. We used BigVGAN (Lee et al., 2023) as the vocoder for waveform generation.

[1] https://montreal-forced-aligner.readthedocs.io

[2] https://github.com/YannickJadoul/Parselmouth

5.3 LLM-based Filler Prediction

We employed Vicuna-7B (Chiang et al., 2023) as the base LLM for the filler prediction task. We fine-tuned this model using a LoRA adapter, which reduced the number of trainable parameters to only 4.20M. Additionally, we evaluated several other instruction-tuned LLMs from the LLaMA (Touvron et al., 2023a) and Qwen (Yang et al., 2024) families as alternative bases for filler prediction. For each of these models, we froze the model weights and fine-tuned a LoRA adapter on our filler prediction task. Their performance is compared in Table 3. In particular, we implemented a variant termed Vicuna w/ SFI (sampling-based filler insertion, following (Wang et al., 2022)) by adding an additional output branch to Vicuna-7B following the SFI approach, with a LoRA adapter integrated during fine-tuning. This branch consists of a single 13-way softmax layer that predicts one of 13 filler words or no filler insertion.

5.4 Evaluation Metrics

5.4.1 Speech Synthesis

We evaluated the performance of the synthesized speech using both subjective and objective metrics. To evaluate the naturalness of the synthesized speech and the similarity to the target speaker, we conducted a mean opinion score (MOS) test and a similarity mean opinion score (sMOS) test. In the MOS test, evaluators rated the naturalness of the speech on a 5-point scale (1 to 5), while in the sMOS test, they assessed how similar the synthesized speech was to the target speech on the same scale. We also conducted filler naturalness MOS (fMOS) and filler style controllability MOS (cMOS) evaluations to assess the naturalness of filler segments in speech and the effectiveness of filler style control. Detailed information on the MOS evaluation can be found in Appendix B.1.

We employed the UTMOS (Saeki et al., 2022) model to automatically predict MOS scores, providing an objective measure of speech quality. To evaluate the pronunciation accuracy of synthesized speech, we used automatic speech recognition (ASR) models, specifically Whisper (Radford et al., 2023) and wav2vec 2.0 (Baevski et al., 2020), to calculate word error rate (WER) and phoneme error rate (PER). To verify how well the synthesized speech matched the target speaker's voice, we extracted speaker embeddings using WavLM-

Table 1: Experimental results of the proposed method. Con, PP, and CA indicate style controllability, pitch predictor, and cross-attention, respectively. The MOS results are reported with a 95% confidence interval.

Method	Con	Token	PP	CA	MOS (↑)	sMOS (↑)	fMOS (↑)	cMOS (↑)	UTMOS (↑)	WER (↓)	PER (↓)	SIM (↑)
GT	-	-	-	-	4.02 ± 0.05	4.22 ± 0.05	4.04 ± 0.06	3.91 ± 0.07	3.6038	5.64	14.27	-
Vocoded	-	-	-	-	3.99 ± 0.05	4.16 ± 0.05	3.98 ± 0.07	3.85 ± 0.06	3.4116	5.64	14.48	-
Matcha-TTS	✗	✗	✗	✗	3.62 ± 0.07	3.47 ± 0.07	3.86 ± 0.07	3.69 ± 0.07	3.3536	**4.56**	11.11	0.9269
FillerSpeech	✓	✓	✗	✗	3.21 ± 0.07	3.39 ± 0.07	3.54 ± 0.08	3.36 ± 0.08	3.2020	4.60	**11.05**	0.9280
	✓	✓	✓	✗	3.80 ± 0.06	$\mathbf{3.53 \pm 0.07}$	3.92 ± 0.07	3.81 ± 0.07	3.8307	9.36	14.28	0.9293
	✓	✓	✓	✓	$\mathbf{3.84 \pm 0.06}$	3.50 ± 0.07	$\mathbf{3.96 \pm 0.07}$	$\mathbf{3.82 \pm 0.07}$	**3.8780**	6.33	12.10	**0.9309**

base-plus for speaker verification[3] and computed speaker embedding similarity (SIM) with the reconstructed waveform.

5.4.2 Filler Prediction

To evaluate the performance of the language model's filler prediction, we calculate the accuracy of filler position, type, duration, and pitch by comparing the model's outputs to the ground truth, with results reported as percentages. This evaluation measures the degree of agreement between the predicted and ground truth labels for each aspect. Note that the accuracy for filler type, duration, and pitch is computed only for those instances where the predicted filler is inserted at the correct position as specified in the ground truth.

In addition to quantitative accuracy, qualitative evaluation is conducted using GPT-4o (OpenAI, 2024), which assigns scores to the model's performance in two filler prediction tasks: position (Score-P) and type (Score-T). Scores range from 1 to 5, where a score of 1 indicates poor performance and a score of 5 indicates excellent performance. Given the inherent variability of natural speech, multiple filler placements may appear natural within a sentence. Therefore, qualitative evaluation is crucial to capture these nuances, which is why we leverage GPT-4o for this assessment.

The scoring process for all tasks takes into account factors such as naturalness, contextual relevance, fluency, and overall suitability for speech interaction, providing a comprehensive assessment of the model's performance. For more details on the evaluation process, please refer to Appendix B.3.

[3] https://huggingface.co/microsoft/wavlm-base-plus-sv/

6 Results

6.1 Speech Synthesis with Filler Insertion

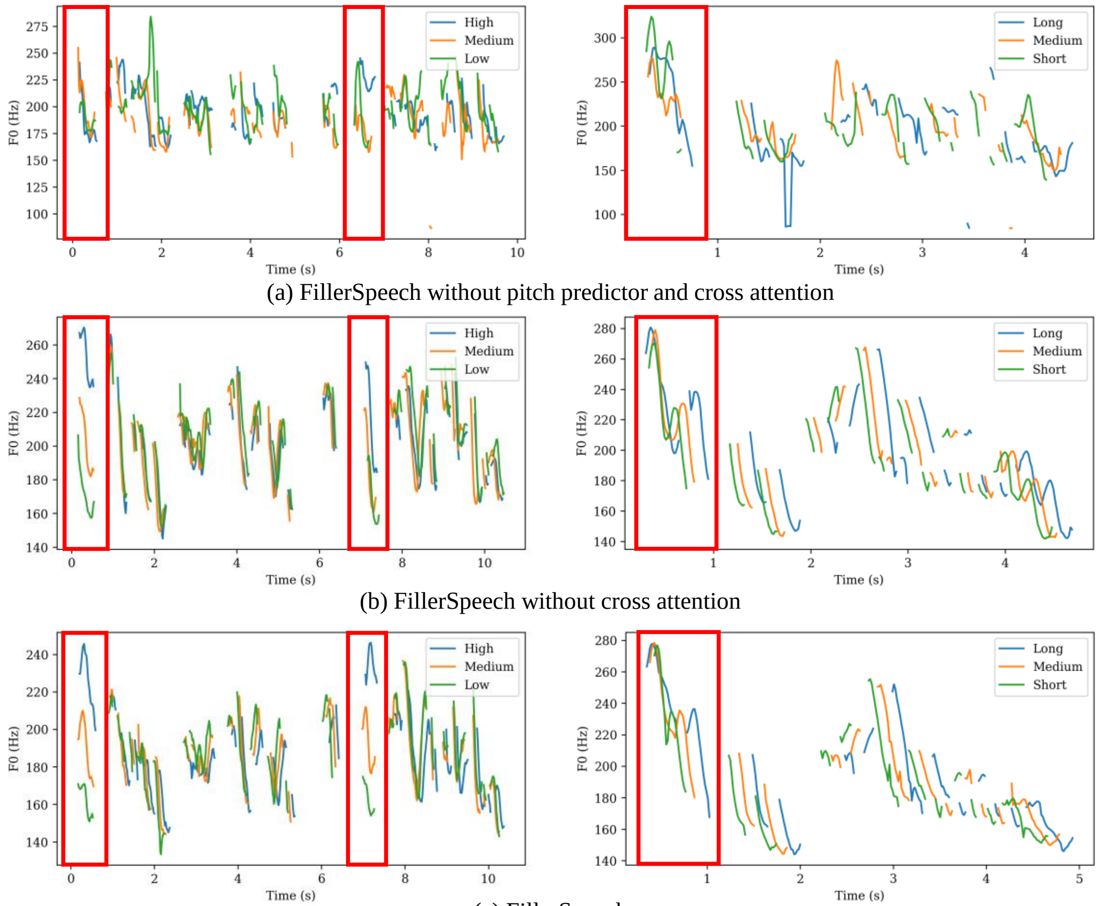
Table 1 shows the subjective and objective evaluation results. The proposed method successfully synthesizes speech with natural-sounding, contextually appropriate filler insertion. Through both subjective MOS tests and objective metrics such as UTMOS and SIM, the synthesized speech demonstrated high naturalness, even when fillers were inserted at various positions in the text.

Notably, the inclusion of pitch information led to a significant improvement in UTMOS, indicating enhanced speech naturalness. However, a decline in pronunciation accuracy was observed, as reflected by increased WER and PER values. This suggests that incorporating pitch information into the prior loss computation may cause the text encoder to focus more on acoustic features at the expense of text-based representations. Consequently, the encoder's ability to represent phonetic information accurately could be negatively impacted, leading to reduced pronunciation accuracy. Nevertheless, incorporating cross-attention helped improve both pronunciation accuracy and speaker similarity.

To specifically evaluate the naturalness of the filler segments in speech, we additionally conducted an fMOS evaluation. As shown in Table 1, using filler tokenization alone generally resulted in reduced naturalness. However, by incorporating a pitch predictor and cross-attention mechanism, our method improved the naturalness of the filler segments.

6.2 Filler Style Control

Our method provides precise control over filler style, allowing the pitch and duration of fillers to be modulated as desired. To validate this capability, we conducted an experiment using the same input text, where a particular filler word was synthesized under different style conditions. Figure 3 presents pitch track plots, showing that the gen-



(a) FillerSpeech without pitch predictor and cross attention

(b) FillerSpeech without cross attention

(c) FillerSpeech

Figure 3: Pitch track visualization of synthesized speech with different filler styles. The red boxes highlight regions where filler words occur. The left column shows varying pitch control, while the right column shows varying duration control.

erated pitch contours vary according to the specified style tokens. For example, when a higher-pitch style was applied, the filler's pitch was consistently higher than in other conditions, demonstrating the method's robustness in controlling prosodic attributes. This highlights the model's ability to adapt fillers dynamically based on stylistic requirements, a critical feature for expressive and context-aware speech synthesis.

Comparisons between FillerSpeech and ablated versions (with certain modules removed) reveal that both the pitch predictor and cross-attention significantly affect the filler's pitch control. Notably, while the pitch predictor slightly degrades pronunciation accuracy, it markedly improves control over the filler's pitch. Similarly, cross-attention—by incorporating word-level pitch conditioning—adds stability to pitch control. In contrast, duration control remains largely unchanged, indicating that pitch information does not substantially contribute to predicting duration.

We also conducted a subjective evaluation cMOS to measure how well the fillers in the generated speech matched the intended style labels. As shown in Table 1, simply adding filler tokenization (without our other modules) degraded performance relative to the baseline. However, when tokenization was combined with the pitch predictor and cross-attention, the filler style controllability improved beyond the baseline.

6.3 LLM-based Filler Prediction

Table 2 shows that our proposed method outperforms all baseline models in the filler prediction task. Specifically, the Vicuna w/o FT model fails to predict fillers accurately. Although the Vicuna w/ SFI model predicts only filler position and type (not duration or pitch), its performance is significantly inferior to that of our model.

To verify that Vicuna-7B was the optimal choice for the LLM component, we compared its performance with other instruction-tuned LLMs. All models were trained under identical conditions, with only the LLM component varied. The results, presented in Table 2, show that Vicuna-7B (fine-tuned on GPT-human dialogue data) outperforms the other models.

6.3.1 Accuracy Evaluation

As shown in Table 3, the Qwen series shows consistent improvements as the model size increases, but overall remains weaker than the LLaMA family. Even the smallest LLaMA model surpasses all Qwen variants, highlighting the stronger generalization ability of the LLaMA backbone. Within the LLaMA series, different model sizes provide complementary strengths, with some models favoring duration while others perform better on pitch.

Vicuna-7B achieves the best overall performance, particularly in position and type prediction. Its dialogue-oriented pretraining allows it to capture contextual cues more effectively than other instruction-tuned LLMs, making it the most reliable model for filler prediction across the evaluated metrics.

6.3.2 LLM-based Evaluation

Table 3 also presents the GPT-based evaluation of naturalness. The trend largely mirrors the accuracy results: Qwen models lag behind, whereas the LLaMA family achieves stronger scores. Within LLaMA, performance improves steadily with scale, reflecting better contextual modeling.

Vicuna-7B again demonstrates the most human-like behavior, receiving the highest evaluations for both filler placement and type. This confirms that its dialogue-focused adaptation not only improves discrete accuracy but also translates into more natural and contextually appropriate filler generation.

6.4 Addressing Potential Hallucination

To address concerns about potential hallucinations in filler prediction, we conducted experiments using a random sampling baseline. The random sampling method generates fillers based on the distribution observed in our training dataset. As shown in Table 2, our proposed Vicuna w/ LoRA approach significantly outperforms the random sampling method. These findings support our claim that the superior performance of our model is due to meaningful filler predictions appropriate to the input text, rather than mere hallucinations.

Table 2: Comparison of filler prediction performance. The Vicuna w/ SFI model only predicts filler position and type.

Method	Accuracy				GPT Scores	
	Position	Type	Duration	Pitch	Position	Type
GT	-	-	-	-	3.25	3.30
Random Sampling	4.32	28.87	-	-	2.80	2.77
Vicuna w/o FT	1.35	13.33	46.67	24.44	2.44	2.47
Vicuna w/ SFI	59.67	38.14	-	-	2.41	2.81
Vicuna w/ LoRA	**82.56**	**78.44**	**52.46**	**63.27**	**3.31**	**3.27**

Table 3: Comparison of LoRA-based fine-tuning results for filler prediction across instruction-tuned LLMs.

Method	Accuracy				GPT Scores	
	Position	Type	Duration	Pitch	Position	Type
GT	-	-	-	-	3.25	3.30
Qwen-1.5B	69.65	60.15	49.87	61.95	3.10	3.13
Qwen-3B	73.59	57.66	49.03	61.19	3.23	3.14
Qwen-7B	75.20	59.76	51.43	62.02	3.23	3.19
LLaMA-1B	81.11	72.85	**52.54**	62.36	3.25	3.22
LLaMA-3B	80.13	73.78	50.28	62.68	3.27	3.20
LLaMA-8B	81.65	76.43	50.55	**63.71**	3.29	3.22
Vicuna-7B	**82.56**	**78.44**	52.46	63.27	**3.31**	**3.27**

7 Conclusion

In this paper, we introduced FillerSpeech, a novel speech synthesis framework that integrates filler insertion with filler style control. We constructed a filler-inclusive speech dataset from a large-scale speech corpus by leveraging an automated method to label fillers with pitch and duration information, thereby eliminating the need for manual annotation. Our approach employs cross-attention mechanisms and a pitch predictor to condition the model on filler style, which enhances control over prosody, especially pitch. While fillers can be manually adjusted to achieve a desired style, we further propose an LLM-based filler prediction method that enables natural filler insertion given only text input. Experimental results demonstrate that our methods substantially improve both speech quality and style control, and that the LLM-based filler prediction effectively predicts filler attributes from text.

8 Limitations

In this work, we employed Vicuna-7B to achieve the best possible performance in filler prediction. However, such a large model can be inefficient in terms of inference speed and resource usage. Nevertheless, our LoRA-based fine-tuning approach is also applicable to smaller LLMs, and we demonstrate that even a 1B-parameter model can outperform existing baselines. This suggests that using a smaller model is a viable alternative when computational efficiency is a priority.

While our discrete labeling approach allows for effective control within a predefined label space, it limits the model's ability to achieve very fine-grained or extreme stylistic control. In future work, we aim to explore more expressive speech synthesis and investigate control methods based on continuous values rather than categorical labels.

9 Acknowledgments

This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) under the artificial intelligence graduate school program (Korea University) (No. RS-2019-II190079) and artificial intelligence star fellowship support program to nurture the best talents (IITP-2025-RS-2025-02304828) grant funded by the Korea government (MSIT).

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jun-Hyeok Cha, Seung-Bin Kim, Hyung-Seok Oh, and Seong-Whan Lee. 2025. Jelly: Joint emotion recognition and context reasoning with llms for conversational speech synthesis. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631. Association for Computational Linguistics.

Wei-Lin Chiang et al. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Tanvi Dinkar, Chloé Clavel, and Ioana Vasilescu. 2022. Fillers in spoken language understanding: Computational and psycholinguistic perspectives. In *Traitement Automatique des Langues, Volume 63, Numéro 3 : Etats de l'art en TAL [Review articles in NLP]*, pages 37–62, France. ATALA (Association pour le Traitement Automatique des Langues).

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2025. LLaMA-omni: Seamless speech interaction with large language models. In *The Thirteenth International Conference on Learning Representations*.

Raul Fernandez, David Haws, Guy Lorberbom, Slava Shechtman, and Alexander Sorin. 2022. Transplantation of conversational speaking style with interjections in sequence-to-sequence speech synthesis. In *Interspeech 2022*, pages 5488–5492.

Edward J Hu et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proc. Int. Conf. Learn. Represent. (ICLR)*.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Eric Liu, Yichong Leng, Kaitao Song, Siliang Tang, Zhizheng Wu, Tao Qin, Xiangyang Li, Wei Ye, Shikun Zhang, Jiang Bian, Lei He, Jinyu Li, and sheng zhao. 2024. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *Forty-first International Conference on Machine Learning*.

Naoyuki Kanda, Xiaofei Wang, Sefik Emre Eskimez, Manthan Thakker, Hemin Yang, Zirun Zhu, Min Tang, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Yufei Xia, Jinzhu Li, Yanqing Liu, Sheng Zhao, and Michael Zeng. 2024. Making flow-matching-based zero-shot text-to-speech laugh as you like. *Preprint*, arXiv:2402.07383.

Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey. 2024. Libriheavy: A 50,000 hours asr corpus with punctuation casing and context. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10991–10995.

Heeseung Kim, Sungwon Kim, and Sungroh Yoon. 2022. Guided-TTS: A diffusion model for text-to-speech via classifier guidance. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11119–11133. PMLR.

Seung-Bin Kim, Sang-Hoon Lee, Ha-Yeong Choi, and Seong-Whan Lee. 2024. Audio super-resolution with robust speech representation learning of masked autoencoder. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:1012–1022.

Sungwon Kim, Kevin J. Shih, Rohan Badlani, Joao Felipe Santos, Evelina Bakhturina, Mikyas T. Desta, Rafael Valle, Sungroh Yoon, and Bryan Catanzaro. 2023. P-flow: A fast and data-efficient zero-shot TTS through speech prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Junhyeok Lee and Hyeongju Kim. 2024. Super Monotonic Alignment Search. *arXiv preprint arXiv:2409.07704*.

Sang-Gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. 2023. BigVGAN: A universal neural vocoder with large-scale training. In *The Eleventh International Conference on Learning Representations*.

Sang-Hoon Lee, Ha-Yeong Choi, Seung-Bin Kim, and Seong-Whan Lee. 2025a. Hierspeech++: Bridging the gap between semantic and acoustic representation of speech by hierarchical variational inference for zero-shot speech synthesis. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15.

Sang-Hoon Lee, Ha-Yeong Choi, and Seong-Whan Lee. 2025b. Periodwave: Multi-period flow matching for high-fidelity waveform generation. In *The Thirteenth International Conference on Learning Representations*.

Sang-Hoon Lee, Seung-Bin Kim, Ji-Hyun Lee, Eunwoo Song, Min-Jae Hwang, and Seong-Whan Lee. 2022. Hierspeech: Bridging the gap between text and speech by hierarchical variational inference using self-supervised representations for speech synthesis. In *Advances in Neural Information Processing Systems*, volume 35, pages 16624–16636. Curran Associates, Inc.

Xiang Li, FanBu FanBu, Ambuj Mehrish, Yingting Li, Jiale Han, Bo Cheng, and Soujanya Poria. 2024. CM-TTS: Enhancing real time text-to-speech synthesis efficiency through weighted samplers and consistency models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3777–3794, Mexico City, Mexico. Association for Computational Linguistics.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. 2023. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

Shivam Mehta, Ruibo Tu, Jonas Beskow, Éva Székely, and Gustav Eje Henter. 2024. Matcha-tts: A fast tts architecture with conditional flow matching. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11341–11345.

Dongchan Min, Dong Bok Lee, Eunho Yang, and Sung Ju Hwang. 2021. Meta-stylespeech : Multi-speaker adaptive text-to-speech generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7748–7759. PMLR.

OpenAI. 2024. Hello gpt-4o.

Puyuan Peng, Po-Yao Huang, Shang-Wen Li, Abdelrahman Mohamed, and David Harwath. 2024. VoiceCraft: Zero-shot speech editing and text-to-speech in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12442–12462, Bangkok, Thailand. Association for Computational Linguistics.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. 2021. Grad-tts: A diffusion probabilistic model for text-to-speech. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8599–8608. PMLR.

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*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2021. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *International Conference on Learning Representations*.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022. In *Proceedings of the Interspeech*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tejasri Velugoti, L. Hemanth Kumar, Koneru Vinay, and M. Vanitha. 2023. Voice flow control using artificial intelligence. In *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, pages 493–496.

Siyang Wang, Joakim Gustafson, and Éva Székely. 2022. Evaluating sampling-based filler insertion with spontaneous TTS. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1960–1969, Marseille, France. European Language Resources Association.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. 2025. MaskGCT: Zero-shot text-to-speech with masked generative codec transformer. In *The Thirteenth International Conference on Learning Representations*.

Nigel Ward. 2006. Non-lexical conversational sounds in american english. *Pragmatics & Cognition*, 14(1):129–182.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Haibin Wu, Xiaofei Wang, Sefik Emre Eskimez, Manthan Thakker, Daniel Tompkins, Chung-Hsien Tsai, Canrun Li, Zhen Xiao, Sheng Zhao, Jinyu Li, and Naoyuki Kanda. 2024. Laugh now cry later: Controlling time-varying emotional states of flow-matching-based zero-shot text-to-speech. *Preprint*, arXiv:2407.12229.

Yuzi Yan, Xu Tan, Bohan Li, Guangyan Zhang, Tao Qin, Sheng Zhao, Yuan Shen, Wei-Qiang Zhang, and Tie-Yan Liu. 2021. Adaptive text to speech for spontaneous style. In *Interspeech 2021*, pages 4668–4672.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Jun-Hak Yun, Seung-Bin Kim, and Seong-Whan Lee. 2025. Flowhigh: Towards efficient and high-quality audio super-resolution with single-step flow matching. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Ge Zhu, Juan-Pablo Caceres, and Justin Salamon. 2022. Filler word detection and classification: A dataset and benchmark. In *Interspeech 2022*, pages 3769–3773.

Éva Székely, Gustav Eje Henter, Jonas Beskow, and Joakim Gustafson. 2019a. How to train your fillers: uh and um in spontaneous speech synthesis. In *10th ISCA Workshop on Speech Synthesis (SSW 10)*, pages 245–250.

Éva Székely, Gustav Eje Henter, Jonas Beskow, and Joakim Gustafson. 2019b. Spontaneous conversational speech synthesis from found data. In *Interspeech 2019*, pages 4435–4439.

Table 4: Hyperparameters of FillerSpeech.

Module	Hyperparameter	FillerSpeech
Embedding	Text	192
	Speaker	64
	Pitch	64
	Duration	64
Encoder	Prenet Conv. Hidden Dim.	192
	Prenet Conv. Layers	3
	Prenet Conv. Kernel Size	5
	Prenet Dropout	0.5
	Transformer Hidden Dim.	320
	Transformer FFN Filter Channels	768
	Transformer Layers	6
	Transformer Kernel Size	3
	Transformer Attention Heads	2
	Transformer Dropout	0.1
	Projection Hidden Dim.	320
	Projection Layers	2
	Projection Kernel Size	3
	Projection Dropout	0.5
Pitch predictor	Conv. Hidden Dim.	192
	Conv. Layers	5
	Conv. Kernel Size	5
	Conv. Dropout	0.5
Duration predictor	Conv. Hidden Dim.	384
	Conv. Layers	2
	Conv. Kernel Size	3
	Conv. Dropout	0.1
CFM decoder	Channels	[512, 512]
	Dropout	0.05
	Blocks	1
	Mid Blocks	2
	Attention Heads	2
	Activation	snakebeta
	Solver	euler
	Sigma min	1e-4
Optimizer	Optimizer	Adam
	Learning Rate	0.0001
	Beta	[0.9, 0.98]

A Prompt for Filler Prediction

To train our LLM to predict the appropriate position, type, duration, and pitch of fillers, as shown in Figure 4, we employed four different types of prompts.

In the first prompt type, the desired filler type is explicitly specified for prediction. In this case, **<TGT_SEN>** denotes the sentence into which the filler will be inserted, and **<FILLER>** indicates the desired filler type.

The second prompt type involves specifying both the desired filler type and the insertion position within the sentence. Here, **<TGT_SEN>** represents the sentence for filler insertion, **<FILLER>** stands for the desired filler type, and **<TGT_POS>** indicates the token position within **<TGT_SEN>** where the filler should be inserted.

For the third prompt type, a set of filler type options is provided, and the LLM selects the most appropriate filler from these options to insert into **<TGT_SEN>**.

In the fourth prompt type, similar to the third, a set of filler type options is given. However, in this case, the LLM not only selects the appropriate filler but also inserts it at the specified token position **<TGT_POS>** within **<TGT_SEN>**.

Across all prompt types, the predicted duration for each filler is classified as either short, medium, or long, while the predicted pitch is categorized as low, medium, or high.

B Details of Evaluation Metrics

B.1 Mean Opinion Score Test

For the subjective evaluation, we conducted both MOS and sMOS tests using Amazon Mechanical Turk, recruiting 20 evaluators for each test. For the evaluations, 50 utterances were randomly sampled from the test set. Additionally, we interspersed fake samples among the test utterances. We filtered out ratings from workers who gave scores to fake samples to exclude unreliable participants.

B.2 Automatic Speech Recognition for Filler-inclusive Speech

In typical TTS tasks, ASR used for pronunciation evaluation employs a text normalization process that includes the removal of filler words from the ASR output. However, because our approach intentionally synthesizes speech with fillers, we deliberately bypass the removal of filler words during text normalization. This allows us to directly assess the performance of our system in generating filler-inclusive speech.

B.3 GPT Score

Building on the studies (Chiang and Lee, 2023; Chiang et al., 2023; Zheng et al., 2023; Fang et al., 2025) that use LLM models to evaluate model outputs, we employ GPT-4o (OpenAI, 2024) to assess the filler prediction ability of our fine-tuned LLM. In this evaluation, GPT-4o examines two key aspects: the prediction of filler positions and the prediction of filler types.

For the filler position, GPT-4o assigns a score ranging from 1 to 5, where a higher score indicates better performance (1: Poor, 2: Below Average, 3: Neutral, 4: Good, 5: Excellent). The evaluation of filler types is carried out in the same manner, with GPT-4o using the identical 1 to 5 scoring scale. Detailed information on the evaluation prompt can be found in Figure 5. Here, the term **{sentence}** refers to the sentence into which the predicted filler is inserted.

Type 1: Prompt template for filler prediction

Target sentence: **<TGT_SEN>**
USER: Add the specified fillers (like **<FILLER>**) to the target sentence to make it sound more natural. For each filler, also specify its duration (short, medium, long) and pitch (low, medium, high) that sound contextually appropriate and natural.
ASSISTANT:

Type 2: Prompt template for filler prediction

Target sentence: **<TGT_SEN>**
USER: Add the specified fillers (like **<FILLER>**) at target positions **<TGT_POS>** in the target sentence. For each filler, also specify its duration (short, medium, long) and pitch (low, medium, high) that sound contextually appropriate and natural.
ASSISTANT:

Type 3: Prompt template for filler prediction

Target sentence**: <TGT_SEN>**
Filler word options: oh, ah, ha, eh, aha, huh, hm, uh, yeah, mm, um, ya, well
USER: Add contextually appropriate fillers to the target sentence. For each filler, also specify its duration (short, medium, long) and pitch (low, medium, high) that sound contextually appropriate and natural."
ASSISTANT:

Type 4: Prompt template for filler prediction

Target sentence: **<TGT_SEN>**
Filler word options: oh, ah, ha, eh, aha, huh, hm, uh, yeah, mm, um, ya, well
USER: Add contextually appropriate fillers at target positions **<TGT_POS>** in the target sentence. For each filler, also specify its duration (short, medium, long) and pitch (low, medium, high) that sound contextually appropriate and natural."
ASSISTANT:

Figure 4: Sample templates for filler prediction (Type 1, 2, 3, 4).

C Analysis on Sampling Steps

In our analysis of the CFM decoder during inference, we evaluated the effect of varying the number of sampling steps by measuring the real time factor (RTF), UTMOS, WER, and speaker embedding cosing similarity (SECS) using Resemblyzer[4] for efficient experiments. As shown in Tables 5 and 6, our model achieves rapid performance improvements even with fewer sampling steps. This improvement is attributed to the use of a pitch predictor, which enables the decoder to condition on encoder outputs that include pitch information. Conversely, as the number of sampling steps increases, we observed a decline in UTMOS and WER, indicating that the pitch information employed for enhanced pitch style control does not necessarily improve pronunciation accuracy. Moreover, with additional sampling steps, SECS increases. This can be explained by the fact that our model's encoder outputs combine text, filler pitch style, and speaker representations, thereby reducing the relative influence of speaker information. Since the sampling process further conditions on the speaker information with encoder outputs, speaker similarity improves with more sampling iterations.

Table 5: Inference performance of the CFM decoder with pitch predictor.

# Steps	RTF ($\downarrow$)	UTMOS ($\uparrow$)	WER ($\downarrow$)	SECS ($\uparrow$)
1	**0.0206**	3.7519	6.41	0.7507
2	0.0213	**3.8945**	**5.59**	0.7600
4	0.0227	3.8780	6.33	0.7736
8	0.0259	3.8260	7.03	**0.7779**

Table 6: Inference performance of the CFM decoder without pitch predictor.

# Steps	RTF ($\downarrow$)	UTMOS ($\uparrow$)	WER ($\downarrow$)	SECS ($\uparrow$)
1	**0.0201**	2.0691	**3.56**	0.6925
2	0.0210	2.6805	3.66	0.7323
4	0.0224	3.0658	4.18	0.7578
8	0.0255	**3.2020**	4.60	**0.7698**

[4] https://github.com/resemble-ai/Resemblyzer

D Analysis on Constructed Data

D.1 Comparison between Pitch labeling Method

We employ two complementary strategies for annotating filler pitch, each designed to capture different aspects of prosodic variation. First, we extract F0 values using Parselmouth and identify filler regions with the MFA. Based on these boundaries, we compute two sets of average F0 values: one for the filler segments and one for the entire utterance.

Our first labeling strategy focuses on comparing F0 values across fillers, independent of their utterance context. For each filler type, we calculate the median F0 separately for male and female speakers to reduce the impact of outliers and account for gender-specific pitch differences. We use XLSR-52-based gender recognition model[5]. Each filler instance is then labeled as low, medium, or high based on whether its F0 is at least four semitones below or above the gender-specific median. The threshold is defined as:

$$\text{threshold}_{\pm} = \text{median} \times 2^{\pm\frac{4}{12}}. \quad (3)$$

The second strategy normalizes filler pitch relative to the overall utterance. Here, we compare the F0 of filler regions to the average F0 of the entire sentence. Fillers whose F0 deviates by at least four semitones from the utterance average are labeled as low or high. If the proportion of fillers labeled as low or high is below 15% when using a four-semitone threshold, a three-semitone threshold is applied instead. As with the first method, these calculations are performed separately for male and female speakers to accommodate gender-specific pitch characteristics. The F0 ratio is computed as:

$$\text{F0 ratio} = \frac{\text{F0\_mean}}{\text{sentence\_F0\_mean}}, \quad (4)$$

with the threshold given by:

$$\text{threshold}_{\pm} = 2^{\pm\frac{4}{12}}. \quad (5)$$

These two methods provide complementary perspectives on pitch variation: one capturing filler-specific deviations across speakers and the other contextualizing filler pitch within each utterance. We evaluated both labeling strategies in our experiments and, as shown in Table 7, found that the first method yields superior performance in speech synthesis.

Table 7: Performance comparison of pitch labeling strategies.

Method	UTMOS (↑)	WER (↓)	SECS (↑)
First Strategy	**3.8780**	**6.33**	**0.7736**
Second Strategy	3.8240	7.55	0.7631

Figure 6 shows the F0 distributions for fillers computed using the first strategy. For most filler types, the distribution near the median is skewed toward values below the median. However, in general, the proportion of fillers labeled as high tends to be higher than those labeled as low.

E Discussion

E.1 General Word Style Control

Due to our model's design which applies style conditioning at the positions of designated tokens, it is capable of modulating the style not only of these tokens but also of general words. Consequently, we demonstrate that even when only a subset of words in the speech data contains pitch or duration information, our approach enables fine-grained control over the overall speech style.

E.2 Potential Risks

While the advancements in speech synthesis technology offer significant benefits, they also raise concerns about potential malicious uses. The ability to generate highly realistic synthesized speech can be exploited to produce deceptive content, such as deepfakes or misleading information, which may have harmful societal implications. To address these risks, a discussion on synthesized speech detection and watermarking techniques during synthesis is necessary to authenticate and trace speech outputs.

E.3 AI asist

We used GPT-4o for proofreading, including typo and sentence correction.

[5] https://huggingface.co/alefiury/wav2vec2-large-xlsr-53-gender-recognition-librispeech

Prompt for GPT Scores – Filler Position (Model: GPT-4o)

You are an expert evaluator of filler placement.
I need your help to evaluate the performance of a model in a filler prediction scenario.
The model receives a target sentence and generates a response by inserting fillers at specific positions.
Your task is to rate the model's response based only on the correctness of filler positions.
Ignore the content of the fillers themselves and focus strictly on whether **the placement of the fillers** aligns with natural speaking patterns.

Scoring Guidelines (Evaluate only the filler position!)
Provide a **single score** on a scale from **1 to 5**, where:

- **1**: Poor
- Fillers are placed incorrectly, disrupting the sentence's natural flow.
- **2**: Below Average
- Some fillers are misplaced, causing minor disruptions.
- **3**: Neutral
- Fillers are placed in acceptable locations but do not necessarily enhance the sentence.
- **4**: Good
- Fillers are mostly well-placed, making the sentence sound natural.
- **5**: Excellent
- Fillers are placed **perfectly**, improving the conversational tone.

Important: Focus **only** on **filler position** for this evaluation.

After evaluating, output the score **only as a number** (e.g., `4`).
Evaluate the following sentence:\n'**{sentence}**'

Prompt for GPT Scores – Filler Type (Model: GPT-4o)

You are an expert evaluator of filler types in natural speech.
I need your help to evaluate the performance of a model in a filler prediction scenario.
The model receives a target sentence and generates a response by inserting fillers of specific types at particular positions.

Your task is to rate the model's response based only on the naturalness and appropriateness of the filler types used in the sentence.
Consider the following aspects:

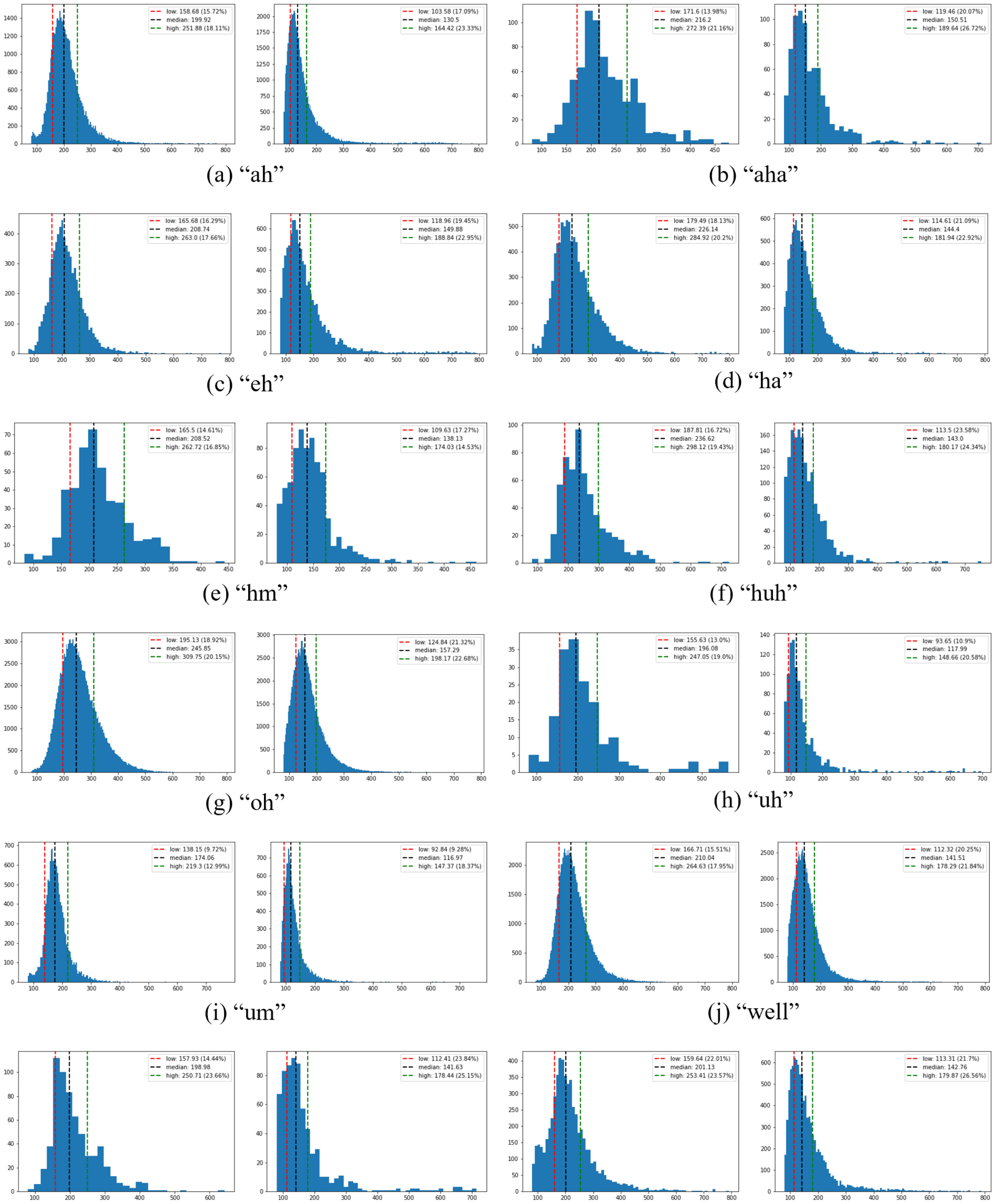
1. **Contextual Suitability**: Assess whether the chosen filler types (e.g., "um," "oh," "yeah") fit naturally within the conversational context of the sentence, enhancing the flow and coherence.
2. **Human-like Selection**: Determine if the filler type corresponds to what a human speaker would likely use in the given situation, considering the tone, intent, and conversational style of the sentence.

Scoring Guidelines
Provide a **single score** on a scale from **1 to 5**, where:

- **1**: Poor
- Filler types are unnatural or disrupt the conversational flow.
- **2**: Below Average
- Some filler types seem out of place or could be improved.
- **3**: Neutral
- Filler types are acceptable but do not necessarily enhance the sentence.
- **4**: Good
- Fillers are mostly well-chosen, making the sentence sound natural.
- **5**: Excellent
- Filler types are **perfectly suited**, improving the conversational tone.

Important: Focus **only** on the filler type selection, not the placement.
Ignore grammar, word choice, and meaning—evaluate only whether the **type of fillers** used is what a human would naturally say.
After evaluating, output the score **only as a number** (e.g., `4`).
Evaluate the following sentence:\n'**{sentence}**'

Figure 5: Prompt templates for GPT-based filler evaluation, using a 1–5 scoring scale.



(a) "ah" (b) "aha" (c) "eh" (d) "ha" (e) "hm" (f) "huh" (g) "oh" (h) "uh" (i) "um" (j) "well" (k) "ya" (l) "yeah"

Figure 6: F0 distribution for each filler type. Odd-numbered columns correspond to female speakers, while even-numbered columns correspond to male speakers.

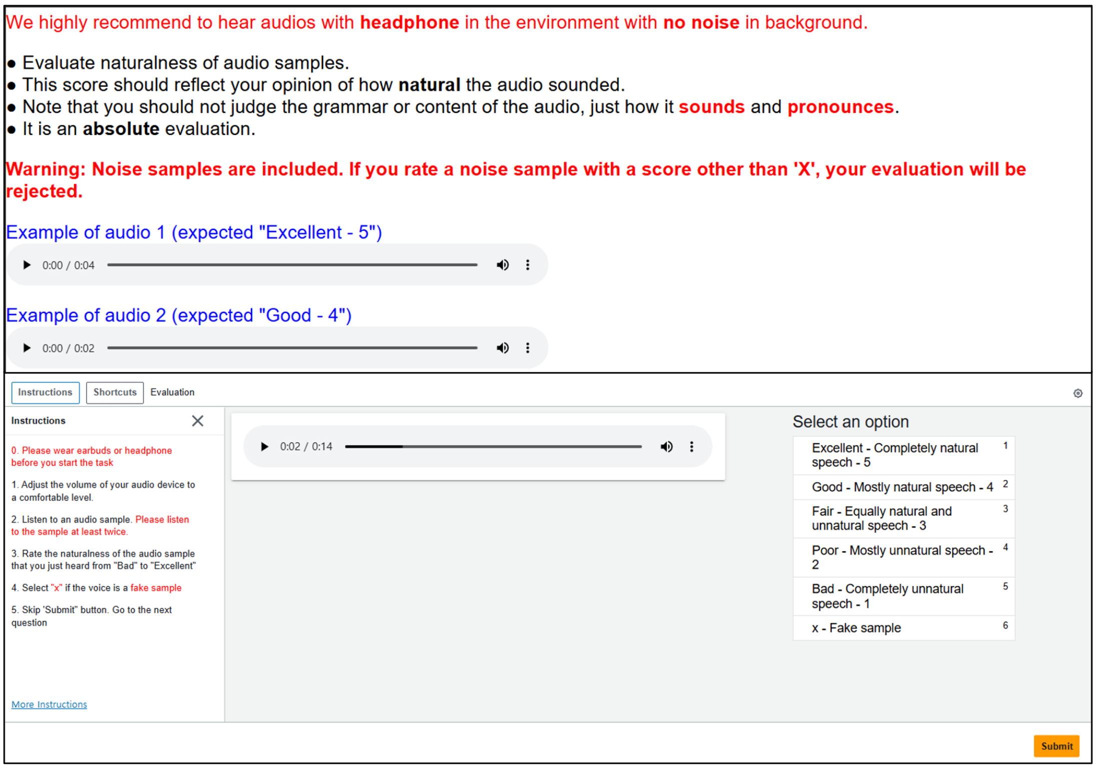


Figure 7: MOS evaluation interface.

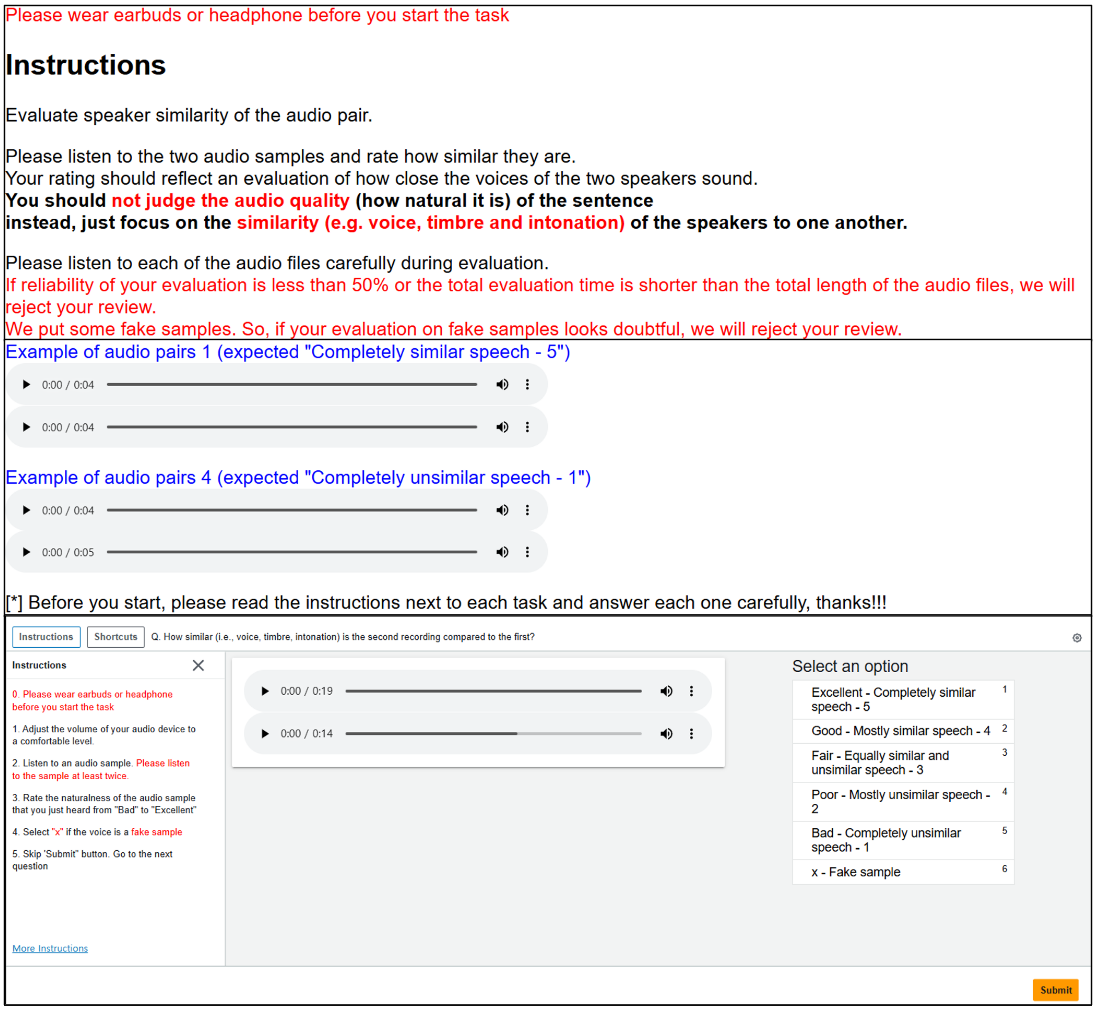


Figure 8: sMOS evaluation interface.

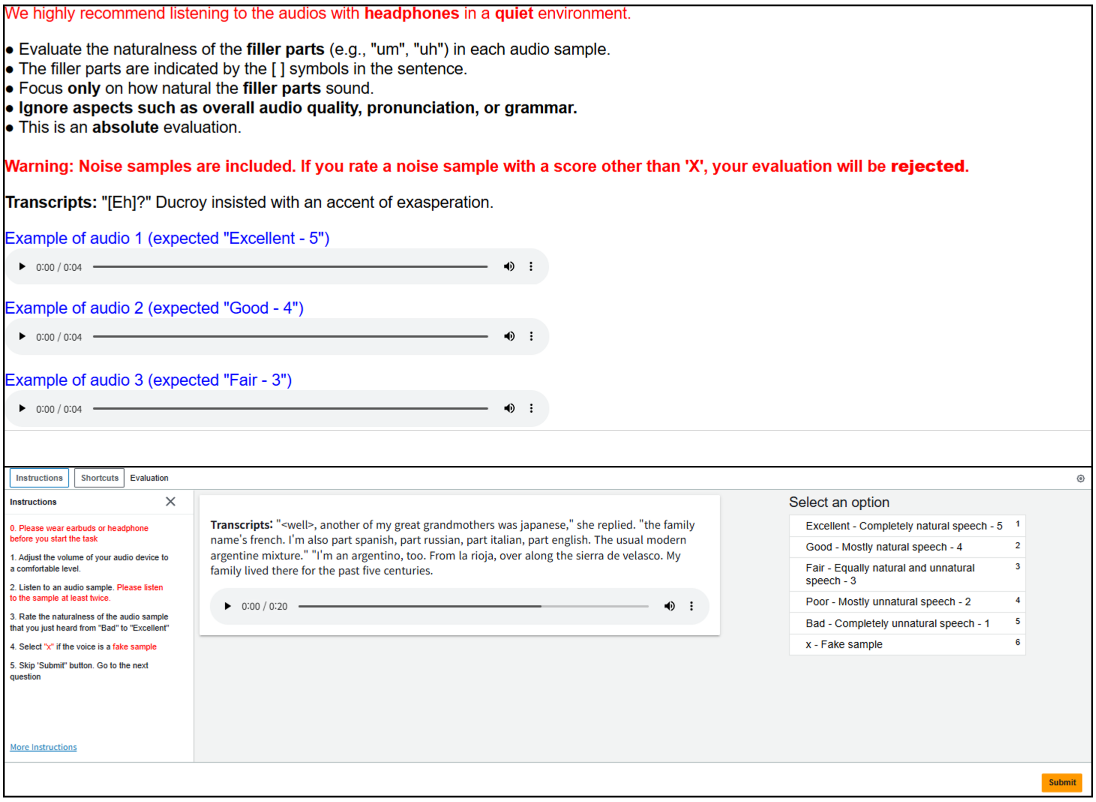


Figure 9: fMOS evaluation interface.

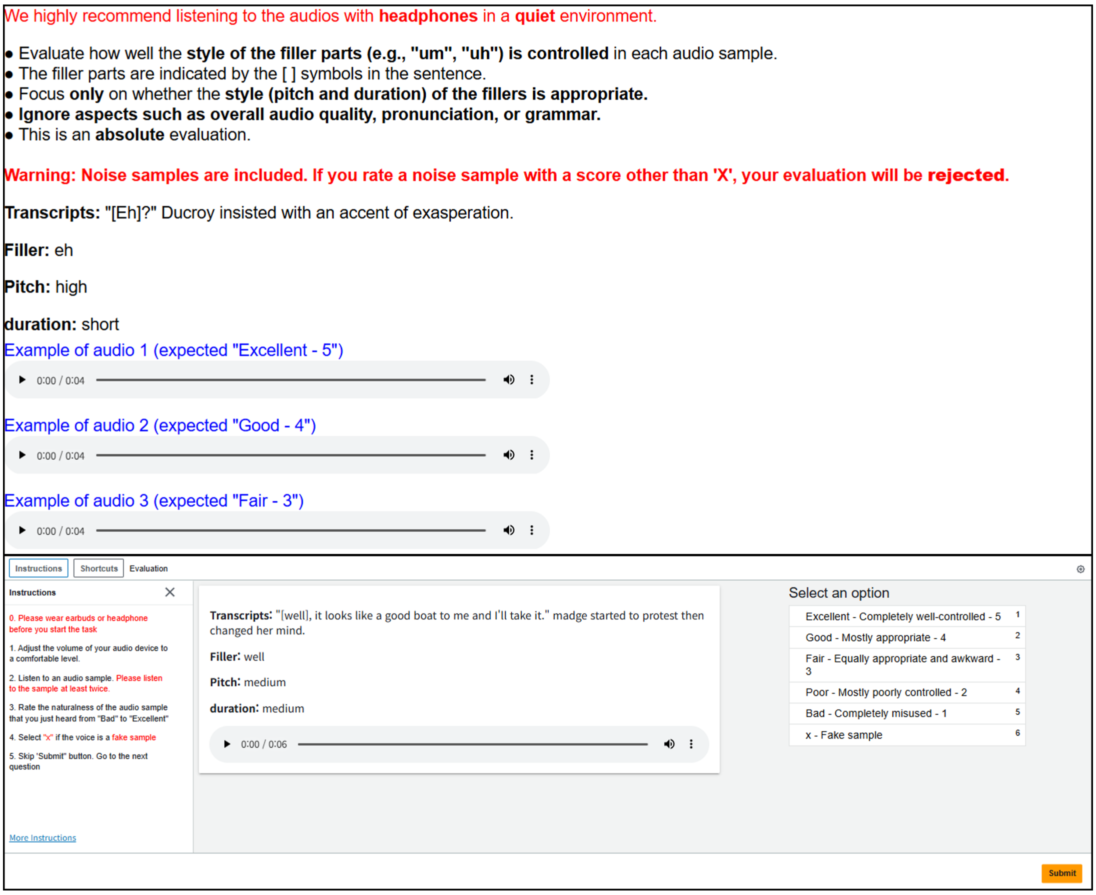


Figure 10: cMOS evaluation interface.