A Machine Speech Chain Approach for Dynamically Adaptive Lombard TTS in Static and Dynamic Noise Environments

Sashi Novitasari, Sakriani Sakti, *Member, IEEE*, and Satoshi Nakamura, *Fellow, IEEE*

***Abstract*—Recent end-to-end text-to-speech synthesis (TTS) systems have successfully synthesized high-quality speech. However, TTS speech intelligibility degrades in noisy environments because most of these systems were not designed to handle noisy environments. Several works attempted to address this problem by using offline fine-tuning to adapt their TTS to noisy conditions. Unlike machines, humans never perform offline fine-tuning. Instead, they speak with the Lombard effect in noisy places, where they dynamically adjust their vocal effort to improve the audibility of their speech. This ability is supported by the speech chain mechanism, which involves auditory feedback passing from speech perception to speech production. This paper proposes an alternative approach to TTS in noisy environments that is closer to the human Lombard effect. Specifically, we implement Lombard TTS in a machine speech chain framework to synthesize speech with dynamic adaptation. Our TTS performs adaptation by generating speech utterances based on the auditory feedback that consists of the automatic speech recognition (ASR) loss as the speech intelligibility measure and the speech-to-noise ratio (SNR) prediction as power measurement. Two versions of TTS are investigated: non-incremental TTS with utterance-level feedback and incremental TTS (ITTS) with short-term feedback to reduce the delay without significant performance loss. Furthermore, we evaluate the TTS systems in both static and dynamic noise conditions. Our experimental results show that auditory feedback enhanced the TTS speech intelligibility in noise.**

***Index Terms*—Text-to-speech, machine speech chain inference, Lombard effect, dynamic adaptation.**

I. Introduction

HUMANS maintain their speech quality in various situations by simultaneously listening to their speech, a mechanism that is also known as the speech chain [1]. The auditory feedback produced from the self-evaluation inside this speaking-while-listening process plays a critical role in speech production [2]–[4]. Moreover, ineffective monitoring of auditory feedback could cause speaking issues [5], [6]. When an error is detected in their speech, humans adapt or tune their speech plan according to the auditory feedback. Auditory feedback is used not only to maintain stability between the sound output and the acoustic goal but also to make situation-dependent adjustments of the prosody attributes [7]. Adjustment is not only manifested in the next utterance but also sometimes presented as a correction of the previous utterance through a re-speaking attempt [8].

In a noisy environment, humans dynamically adjust their vocal effort according to the auditory feedback as a way to maintain their speech intelligibility, a phenomenon known as the Lombard effect [9], [10]. This adjustment affects not only speech intensity but also other aspects such as speech pitch and speaking rate [11]–[13]. As a response to ambient noise, intensity and pitch tend to increase, while speaking rate tends to become slower. Several works have reported the response latency in the human Lombard effect to be about 90-287 ms [14]–[16].

Text-to-speech synthesis (TTS) systems have been developed to mimic human speech production. However, unlike the human communication system in which speech production and perception are closely connected, TTS development focuses only on speech production, independent of speech perception. Under clean conditions, neural TTS successfully synthesize a highly natural speech given only a text [17]–[20]. However, in noisy conditions, TTS speech intelligibility degrades because most systems have not been designed to handle noisy environments. Furthermore, since TTS only learns to speak without listening and understanding the situation, they do not have the ability to adapt to the situation. A widely used solution for achieving TTS with high intelligibility in noisy places is to adapt the system offline using Lombard speech from a particular noisy condition [21], [22].

This paper proposes an alternative approach that is closer to the human Lombard effect. Specifically, we propose Lombard TTS in a machine speech chain framework to synthesize speech with dynamic adaptation. The idea of mimicking the speech chain mechanism by a machine was raised by Tjandra et al. [23], where automatic speech recognition (ASR) system as speech perception and TTS as speech production can support each other given unpaired data (Fig. 1(a)). Unfortunately, this framework was only proposed as a semi-supervised training mechanism.

Manuscript received 25 January 2022; revised 23 May 2022 and 13 July 2022; accepted 22 July 2022. Date of publication 5 August 2022; date of current version 19 August 2022. This work was supported by JSPS KAKENHI under Grants JP21H05054 and JP21H03467. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kai Yu. *(Corresponding author: Sakriani Sakti.)*

Sashi Novitasari is with the Nara Institute of Science and Technology, Ikoma 630-0192, Japan (e-mail: sashi.novitasari.si3@is.naist.jp).

Sakriani Sakti is with the Japan Advanced Institute of Science and Technology, Nomi 923-1211, Japan, and also with the Department of Science and Technology, Nara Institute of Science and Technology, Ikoma 630-0192, Japan.

Satoshi Nakamura is with the Department of Science and Technology, Nara Institute of Science and Technology, Ikoma 630-0192, Japan (e-mail: ssakti@jaist.ac.jp; s-nakamura@is.naist.jp).

Digital Object Identifier 10.1109/TASLP.2022.3196879

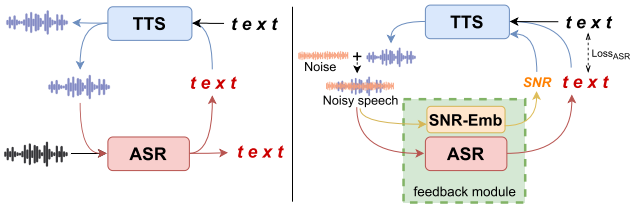

Fig. 1. (a) Previous machine speech chain utilized only for semi-supervised training method; (b) proposed machine speech chain for use in both training and a dynamically adaptive inference method.

During inference, ASR and TTS perform separately as standard systems, leaving the adaptation problem. In contrast, in this work, we propose an advanced version of a machine speech chain that applies the feedback mechanism during training and inference of an end-to-end neural TTS.

The proposed TTS[1] with a machine speech chain mechanism (Fig. 1(b)) speaks with a Lombard effect to enhance speech intelligibility in noisy conditions dynamically based on the auditory feedback. Here, we focus on auditory feedback consisting of the ASR loss as the speech intelligibility measure and the speech-to-noise ratio (SNR) prediction as power measurement. Humans commonly adapt their speech based on several factors, such as the target listeners, tasks, and environments. In this paper, we focus only on adapting the TTS speech prosody based on noise sounds. The prosody attributes adapted by our TTS are speech intensity, pitch, and speaking rate or duration. This study mainly considers a text-to-Mel-spectrogram module and does not discuss the vocoder module in detail.

Two versions of TTS are investigated: non-incremental TTS with utterance-level feedback and incremental TTS (ITTS) with short-term feedback to reduce the delay without significant performance loss. Previously, most existing works focus only on static noises, while here, we also evaluate the TTS given both static and dynamic noise conditions. This may be the first deep learning framework that mimics human the Lombard speech mechanism in a noisy environment and the first TTS study investigating the performance in both static and dynamic noise conditions, to the best of our knowledge. In summary, our contributions include:

- Construction of a Lombard speech dataset for Lombard TTS with well-known Wall Street Journal content [25].
- Construction of non-incremental TTS in a machine speech chain framework, shown in Fig. 2(a), that takes a sentence text as input and then improves the speech using sentence-level auditory feedback through the re-speaking attempt. It synthesizes the speech by assuming that the environmental noise within the re-speaking attempts is the same.
- Construction of incremental TTS (ITTS) in a machine speech chain framework, shown in Fig. 2(b). It incrementally synthesizes the speech by progressively taking a short text segment and feedback from the past speech segment. By using the short-term feedback, ITTS immediately adapts to the environmental changes.
- Evaluation of the performance of both non-incremental TTS and ITTS under static and dynamic noise conditions.

[1]The initial part of this work was presented in [24]. The previous work only focused on non-incremental Lombard TTS in static noises. In this work, we provide a more comprehensive and systematic description of the method, as well as ITTS for dynamic noise conditions.

II. Related Works

A. Existing Lombard TTS

Lombard speech synthesis is designed to produce intelligible speech in the presence of noise. This system has gained attention within the speech community, which was reflected in the Hurricane Challenge [26], [27] for speech synthesis and the evaluation of speech enhancement systems under noisy conditions.

The related works on speech enhancement applied signal processing on speech to improve the intelligibility in noisy conditions. The earlier works performed speech modification through a statistical method with fixed parameters based on known noises. The enhancement operations include modifications of duration [28], pitch, energy contour, formant sharpness, and intensity [29]. Several works also proposed other spectral modification approaches, such as spectral tilt, spectrum contrast enhancement, and harmonic component preservation at in the low-frequency region to emphasize the speech features that are important for speech perception [30]. Spectral shaping and a dynamic range compression method were also studied [31]. Next, AdaptDRC [32] was proposed for speech enhancement controlled by the short-term speech intelligibility index. It enhanced the speech content at high frequencies by also boosting the low-energy speech content through time- and frequency-dependent dynamic range compression and frequency-shaping. Another work also proposed a noise-dependent AdaptDRC with the reverberation-dependent onset enhancement and overlapping masking reduction [33]. Although the above approaches could be applied to both natural and synthesized speech, a noise signal separated from the speech was required. Their experiments were generally carried out by assuming perfect noise was available. Speech and noise separation in real situations might be challenging, especially in dynamic noise conditions. In our proposed approach, we use a TTS to directly synthesize the Lombard speech given the text and feedback based on synthesized speech with the noise.

In the conventional TTS approach, Lombard speech synthesis was commonly done using the parametric model with Hidden Markov Model (HMM). GlottHMM [34], [35] applies a glottal inverse filtering technique in the vocoder of HMM TTS to improve speech intelligibility in the presence of noise. Speech was synthesized by filtering the glottal excitation with a vocal tract filter, where the excitation signal was generated from the real glottal flow extracted from natural speech. A speaking style adaptation approach has also been studied, in which the HMM TTS system is adapted with a small amount of Lombard speech after training with normal speech [21], [36]. The performance of the statistical approach, however, has been limited by poor acoustic modeling [22].

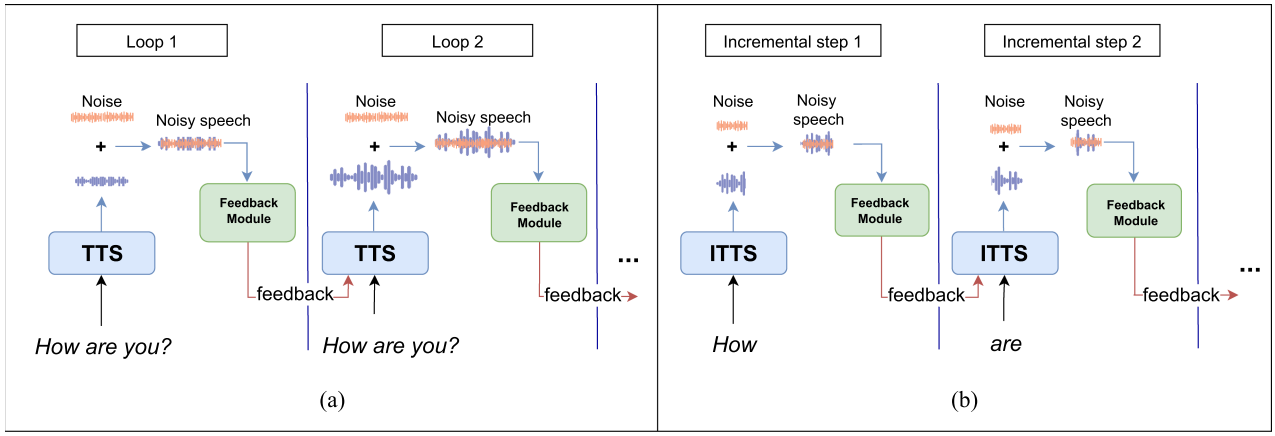

Fig. 2. Unrolled feedback loops of the proposed TTS with the machine speech chain mechanism. (a) Proposed non-incremental TTS performs adaptation by taking auditory feedback of previously synthesized speech with the same text, while (b) proposed ITTS performs adaptation by generating and using the auditory feedback progressively.

Recently, the neural network approach has also gained attention for synthesizing Lombard speech in an end-to-end manner. A recent study proposed Lombard TTS by tuning a Tacotron model on the Lombard speech data [22]. The model was first pre-trained on the normal speech data. In another study, a multi-style Tacotron TTS was proposed with a framework that could synthesize speech in normal, whispered, and Lombard speech styles [37]. In their experiment, TTS training was done by including speech spoken in these three styles by a single speaker in the training material. The TTS generates the styled speech by treating the three speaking styles as three different speakers, so the output speech style is decided based on the speaker embedding vector.

In this work, we focus on an end-to-end transformer network-based Lombard TTS with dynamic adaptation using auditory feedback that is estimated directly from noisy TTS speech. The previous works performed offline tuning to improve speech intelligibility in the presence of noise. For our TTS, instead of relying on the presumption of a single environmental condition, we train it under several noise conditions and allow this system itself to determine the speaking style during inference through a self-evaluation and feedback mechanism.

B. Basic Machine Speech Chain

The basic machine speech chain [23] trains sequence-to-sequence ASR and TTS together by connecting them via closed-loop feedback in a semi-supervised approach, shown in Fig. 3. During inference, ASR and TTS perform separately. The ASR and TTS are the sequence-to-sequence (seq2seq) neural networks consisting of the encoder, decoder, and attention module [38].

Machine speech chain training consists of two stages: supervised and unsupervised training. The supervised training stage is done by training the ASR and TTS independently with a small amount of paired speech-text data. This stage acts as a knowledge initialization phase for both models. In the unsupervised training stage, ASR and TTS are trained together via closed-loop feedback using the unpaired speech and text data. In this stage, ASR and TTS support each other through the mutual use of the feedback. The feedback loop consists of two unrolled processes:

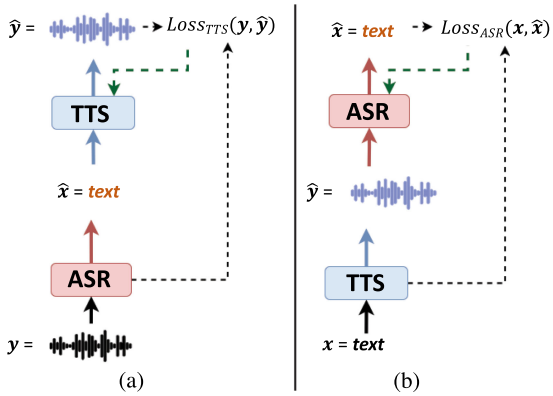

Fig. 3. Basic machine speech chain feedback loop unrolled into two processes: (a) ASR-to-TTS and (b) TTS-to-ASR.

- **ASR-to-TTS**. ASR transcribes a speech utterance $\boldsymbol{y}$, with a length T, into a sentence text $\hat{\boldsymbol{x}}$, and then TTS generates a speech utterance $\hat{\boldsymbol{y}}$ based on ASR output $\hat{\boldsymbol{x}}$. A training loss is calculated based on the original speech $\boldsymbol{y}$ and TTS speech $\hat{\boldsymbol{y}}$ to optimize the TTS system.
- **TTS-to-ASR**. Given a complete sentence text $\boldsymbol{x}$ with a length S, TTS generates speech $\hat{\boldsymbol{y}}$ and ASR transcribes the TTS speech $\hat{\boldsymbol{y}}$ into text $\hat{\boldsymbol{x}}$. A loss is calculated based on the original text $\boldsymbol{x}$ and ASR output text $\hat{\boldsymbol{x}}$ to optimize the ASR system.

As explained earlier, in this work, we propose an advanced version of a machine speech chain that applies a feedback mechanism during training and inference with transformer-based TTS and ASR.

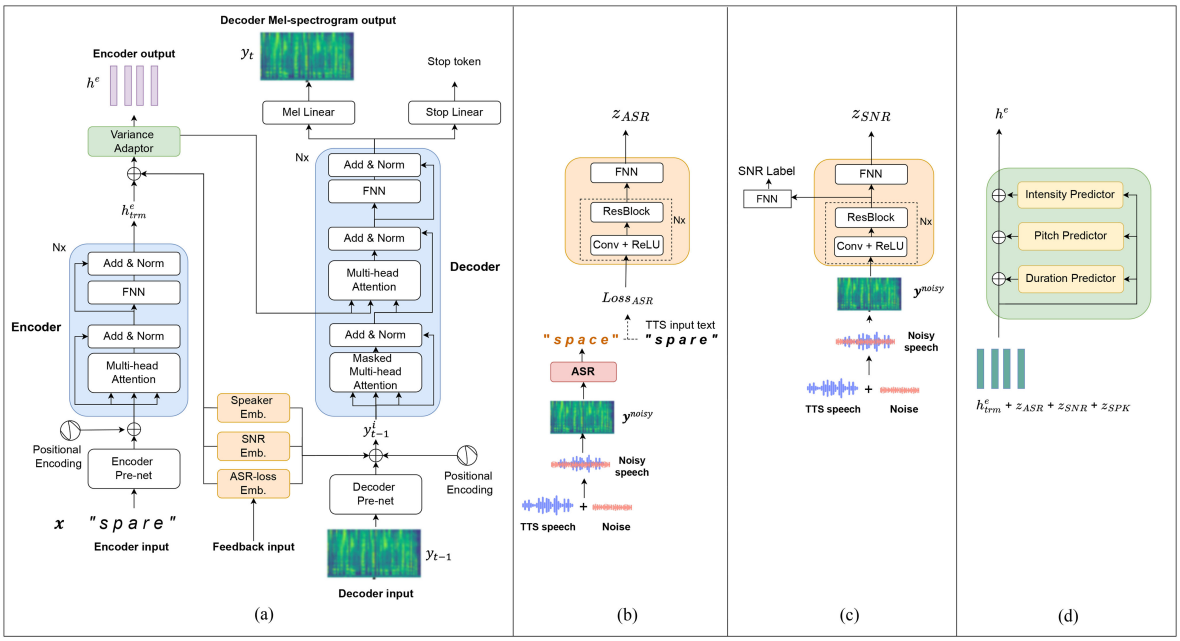

Fig. 4. Architecture: (a) proposed TTS with an autoregressive transformer-based encoder-decoder structure, extended with (b) ASR-loss embedding, (c) SNR embedding, and (d) variance adaptor [39] modules.

III. Proposed Non-Incremental TTS in Machine Speech Chain Framework

A. Overview

When the environment becomes noisy, our TTS tries to synthesize the speech with higher intensity, higher pitch, and slower speed than the speech before the adaptation. The basic or standard TTS is based on the MultiSpeech framework [19], which is a multi-speaker transformer TTS with an encoder and an autoregressive decoder [19]. To achieve dynamic adaptation, we extended the basic structure with auditory feedback modules (ASR-loss embedding and SNR embedding) and a variance adaptor; an overview of this architecture is given in Fig. 4(a). The proposed TTS generates the speech Mel-spectrogram $\boldsymbol{y} = [y_1, y_2, \ldots, y_T]$ with a length of T given the character sequence $\boldsymbol{x} = [x_1, x_2, \ldots, x_S]$ with length S. The proposed TTS adapts the speech prosody attributes by also taking the auditory feedback in SNR embedding (z_{SNR}) and ASR-loss embedding (z_{ASR}) as input. In inference, adaptation is done in several feedback iterations until ASR loss converges. The conversion of the Mel-spectrogram into a waveform is done using a CBHG (1-D Convolution Bank + Highway + bidirectional GRU) module and the Griffin-Lim algorithm, similar to the Tacotron framework [17].

We use a speaker recognition module implementing the DeepSpeaker framework [40] to generate the speaker embedding vector for our multi-speaker TTS, following the implementation of TTS in the basic machine speech chain framework. The module generates the embedding vector $z_{SPK} = Speaker\ Embedding(\boldsymbol{y})$ from a Mel-spectrogram to represent the speaker identity using a convolution network-based structure. We pre-train the DeepSpeaker model, and the weight is maintained during the TTS training for stable embedding. Inside the TTS, speaker embedding is merged with the encoder output and the decoder input.

In this study, we construct three TTS models with different feedback configurations to investigate the effect of auditory feedback in the machine speech chain framework. Each system is trained using normal speech and Lombard speech.

1) TTS With SNR Feedback: The proposed TTS synthesizes speech based on text input and SNR feedback as embedding. The SNR feedback represents the SNR or speech and noise intensity ratio, which is a measure of how well the TTS speech can be heard in a noisy environment. Commonly, SNR can be calculated by measuring the intensities of the speech and noise separately. However, separating speech and noise in a real-world situation could be challenging because, for example, noises might dynamically change. In our approach, we use machine learning as a neural network to obtain the SNR directly from noisy speech, where the speech and noise are mixed. Given an SNR embedding, the proposed TTS attempts to re-synthesize speech with a higher SNR ($\geq$ 20 dB), indicating that the speech is louder than the noise.

We implement the SNR embedding module using convolution network layers with an average pooling operation (Fig. 4(c)). This generates an utterance-level embedding z_{SNR} from noisy TTS speech features $\boldsymbol{y}^{noisy}$:

$$z_{SNR} = SNR\ Embedding(\boldsymbol{y}^{noisy}). \tag{1}$$

Before training the TTS, we pre-train the SNR embedding module (Conv + ReLU and ResBlock) as an SNR recognition module so that the TTS can converge faster. We can initialize the SNR recognition model as a classification or a regression model. The SNR recognition model recognizes the average SNR in an utterance. In SNR classification, we first define several SNR classes. Here, the SNR recognition model generates SNR

embedding vectors by learning to classify the SNR given noisy speech utterances. Model optimization is done by minimizing the cross-entropy loss:

$$Loss_{SNR-CLS}(l, p_l) = -\sum_{c_l=1}^{C_l} \mathbb{1}(l = c_l) * \log p_l[c_l], \quad (2)$$

where l is the reference SNR label, p_l is the predicted SNR probability, and C_l is the number of SNR classes. On the other hand, the SNR regression model is trained to estimate the SNR as a real value. It is optimized using L2 loss:

$$Loss_{SNR-REG}(l, \hat{l}) = (l - \hat{l})^2, \quad (3)$$

where $\hat{l}$ is the predicted SNR at the utterance level.

Inside the TTS encoder, SNR embedding vector z_{SNR} is integrated with the TTS encoder transformer output h^e_{trm} and speaker embedding z_{SPK} to obtain the final TTS encoder output h^e, written as

$$h^e = h^e_{trm} + z_{SPK} + z_{SNR}. \quad (4)$$

On the decoder side, embedding vectors z_{SPK} and z_{SNR} are also combined with the decoder pre-net output and the positional encoding PE to obtain the decoder intermediate input y^i_{t-1}:

$$y^i_{t-1} = prenet(y_{t-1}) + z_{SPK} + z_{SNR} + PE. \quad (5)$$

Following this, the decoder multi-head attention query, key, and value are the encoder output and decoder input that have been embedded with the auditory feedback.

TTS model optimization is done based on the standard transformer TTS loss function:

$$Loss_{TTS}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) = \frac{1}{T}\sum_{t=1}^{T}((y_t - \hat{y}_t)^2 - (b_t \log(\hat{b}_t) + (1 - b_t)\log(1 - \hat{b}_t))), \quad (6)$$

where $\boldsymbol{Y} = [\boldsymbol{y}, \boldsymbol{b}]$ and $\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}, \hat{\boldsymbol{b}}]$. $\boldsymbol{b}$ and $\hat{\boldsymbol{b}}$ are the reference and the predicted probability of stop token that marks the end of speech.

2) TTS With SNR-ASR Feedback: The proposed TTS generates speech based on text input and auditory feedback in SNR and ASR-loss embedding. The ASR-loss embedding, shown in Fig. 4(b), represents the speech intelligibility measurement of how well the noisy TTS speech can be recognized by an ASR. ASR-loss embedding vector z_{ASR} is generated by transcribing a noisy TTS speech using an ASR, which is written as

$$\boldsymbol{p}_x = p(\boldsymbol{x}|\boldsymbol{y}^{noisy}), \quad (7)$$

where $\boldsymbol{p}_x$ is the ASR posterior, and then calculating the loss between the ASR hypothesis and the TTS input text. The ASR-loss embedding module, which is a stack of convolutional layers with average pooling, produces z_{ASR} as an utterance-level embedding by taking $Loss_{ASR}(\boldsymbol{x}, \boldsymbol{p}_x)$, which is a sequence of character-level loss in a sentence:

$$z_{ASR} = ASR\ Loss\ Embedding(Loss_{ASR}(\boldsymbol{x}, \boldsymbol{p}_x)), \quad (8)$$

Here, suppose a sentence text $\boldsymbol{x}$ consists of S characters, the s-th character (x_s) loss is calculated by

$$Loss_{ASR}(x_s, p_{x_s}) = -\sum_{c=1}^{C} \mathbb{1}(x_s = c) * \log p_{x_s}[c], \quad (9)$$

where $Loss_{ASR}(x_s, p_{x_s})$ is the character-level loss and C is the size of ASR output vocabulary. ASR text decoding is done by teacher-forcing mechanism based on the TTS text input.

Inside the main part of TTS, ASR-loss embedding is combined with the TTS encoder output and the decoder input along with the speaker and the SNR embedding vectors:

$$h^e = h^e_{trm} + z_{SPK} + z_{SNR} + z_{ASR}, \quad (10)$$

$$y^i_{t-1} = prenet(y_{t-1}) + z_{SPK} + z_{SNR} + z_{ASR} + PE. \quad (11)$$

In the proposed TTS training and inference, we use a pre-trained ASR to transcribe TTS speech. The ASR-loss embedding module is trained directly during TTS training without a pre-training step. TTS optimization is done by minimizing the TTS loss in (6).

3) TTS With SNR-ASR Feedback and Variance Adaptor: In addition to the SNR and ASR-loss embedding feedback, we implement a variance adaptor module in the proposed TTS with a similar approach to FastSpeech2 [39]. The variance adaptor is intended to guide the prosody adaptation by predicting the prosody attributes from the encoded text input and the auditory feedback. The variance adaptor, shown in Fig. 4(d), consists of three components: a pitch predictor, an intensity predictor, and a duration predictor. This module is applied in the TTS encoder and provides the following output:

$$h^e = Var\ Adaptor(h^e_{trm} + z_{SPK} + z_{SNR} + z_{ASR}). \quad (12)$$

The decoder input follows (11). In our duration predictor, instead of predicting the token duration as an integer to regulate the encoder output length like in the original FastSpeech2 framework, it estimates the duration as a real value similar to the other predictors. The encoder output length in our model follows the standard autoregressive transformer TTS.

The proposed TTS with variance adaptor is trained with the standard TTS loss function combined with the variance predictor losses. The variance predictor loss is calculated by the MSE loss function:

$$Loss_{pred}(\boldsymbol{v}, \hat{\boldsymbol{v}}) = \frac{1}{S}\sum_{s=1}^{S}(v_s - \hat{v}_s)^2, \quad (13)$$

where v_s is the normalized reference value for the predictors inside the variance adaptor and $\hat{v}_s$ is the predictor output at timestep s. The reference intensity, pitch, and duration are estimated from the TTS reference output speech. The TTS training loss function becomes

$$\begin{aligned} Loss_{TTS}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) &= \frac{1}{T}\sum_{t=1}^{T}((y_t - \hat{y}_t)^2 - (b_t \log(\hat{b}_t) + (1 - b_t)\log(1 - \hat{b}_t))) \\ &\quad + Loss_{pred}(\boldsymbol{v}^P, \hat{\boldsymbol{v}}^P) + Loss_{pred}(\boldsymbol{v}^G, \hat{\boldsymbol{v}}^G) \\ &\quad + Loss_{pred}(\boldsymbol{v}^D, \hat{\boldsymbol{v}}^D), \end{aligned} \quad (14)$$

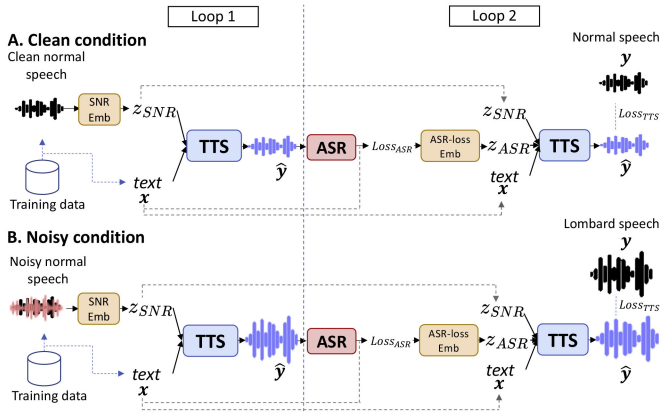

Fig. 5. Proposed TTS training in two feedback loops based on clean and noisy conditions.

where $\boldsymbol{Y} = [\boldsymbol{y}, \boldsymbol{b}, \boldsymbol{v}^P, \boldsymbol{v}^G, \boldsymbol{v}^D]$ and $\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{v}}^P, \hat{\boldsymbol{v}}^G, \hat{\boldsymbol{v}}^D]$. Here, $\boldsymbol{v}^P$, $\boldsymbol{v}^G$, and $\boldsymbol{v}^D$ are the reference pitch, the intensity, and the duration, and $\hat{\boldsymbol{v}}^P$, $\hat{\boldsymbol{v}}^G$, and $\hat{\boldsymbol{v}}^D$ are the pitch, the intensity, and the duration predicted by the predictor.

B. Training Method

The proposed TTS training method is illustrated in Fig. 5. To enable dynamic adaptation, we train the proposed TTS using inputs, consisting of text and auditory feedback embedding vectors, and an output target, which is reference speech representing the speech after adaptation. Auditory feedback represents the speech condition before it is adapted into the target speech. In training, the SNR embedding is pre-computed from clean or noisy normal speech in the training data, while ASR-loss embedding is computed from TTS speech generated during training. Therefore, the speech data required for training are the clean normal speech, the normal speech with additive noise (noisy normal speech), and the clean Lombard speech.

For speech synthesis and adaptation with the re-speaking mechanism, we trained the proposed TTS in one or two feedback loops based on the type of architecture:

1) **TTS training with SNR feedback**: For the proposed TTS without the ASR-loss embedding module, we apply one-loop training using pre-computed SNR embedding and text based on the training data.
2) **TTS training with SNR-ASR feedback**: For the proposed TTS with the ASR-loss embedding module, we generate speech in two feedback loops. The SNR embedding vector is calculated in the first loop based on the training data, and we use the same vector in the second loop. The ASR-loss embedding is calculated in the second loop based on the TTS speech generated in the first loop, thus, the TTS can learn the ASR-loss pattern based on the synthesized speech for a more realistic ASR feedback processing.

We consider two target conditions: clean and noisy. In the clean condition, the proposed TTS produces normal speech without the Lombard effect. In the noisy condition, the proposed TTS produces Lombard speech. We apply batch training to train the proposed TTS in which a batch consists of a mix of speech samples for clean and noisy conditions. The details of the training mechanism are described below with two feedback loops based on the type of target condition:

1) **TTS training in clean condition**: Speech generation is learned using text and clean normal speech data. Normal speech is speech which is uttered in a clean condition without the Lombard effect. It has a lower intensity, a lower pitch, and a faster speaking rate than Lombard speech. In the training, the SNR embedding and the output speech reference are based on clean normal speech. Therefore, before starting the training, we first compute the SNR embedding from clean normal speech. In the first feedback loop, TTS generates normal speech by taking the text, pre-computed SNR embedding, and ASR-loss embedding in a zero vector as the input. In the second feedback loop, we repeat the same process but use the ASR-loss embedding computed from the TTS speech features predicted in the first loop.
2) **TTS training in noisy condition**: Speech generation is learned using text, noisy normal speech, and clean Lombard speech with high SNR and low ASR loss in the corresponding noisy condition. Clean Lombard speech is a speech under the Lombard effect but without noise in the audio. In our experiment, the clean Lombard speech is a synthetic Lombard speech generated by modifying the prosody of normal speech (intensity, pitch, duration) into Lombard speech using SoX audio manipulation toolkit [41], [42]. Noises were not included in the resulting audio. The detail is discussed in Section V. In the training loop, we use SNR embedding generated from noisy normal speech in the training data and the text as the input and the Lombard speech as the target. The ASR-loss embedding generation and utilization method are the same as those in the clean condition case. In the first loop, ASR-loss embedding is a zero vector. In the second loop, the ASR-loss embedding is generated from TTS speech features generated in the previous loop.

IV. Proposed Incremental TTS in Machine Speech Chain Framework

A. Overview

The proposed ITTS in the machine speech chain framework synthesizes the speech incrementally with the auditory feedback (Fig. 2(b)). Our ITTS incremental unit is fixed to W words. It is a complex task to incrementally synthesize a well-performed speech based on short text. ITTS has to decide on the speech output based on a short information sequence, while speech has a continuous representation and heavily depends on context. A general approach to improving performance is to introduce contextual look-back words and look-ahead words in the input text window [43]–[45].

Incremental speech synthesis and adaptation are done by generating and utilizing auditory feedback incrementally or progressively. In the first incremental step, ITTS synthesizes the first W words of speech. Then in the second incremental step,

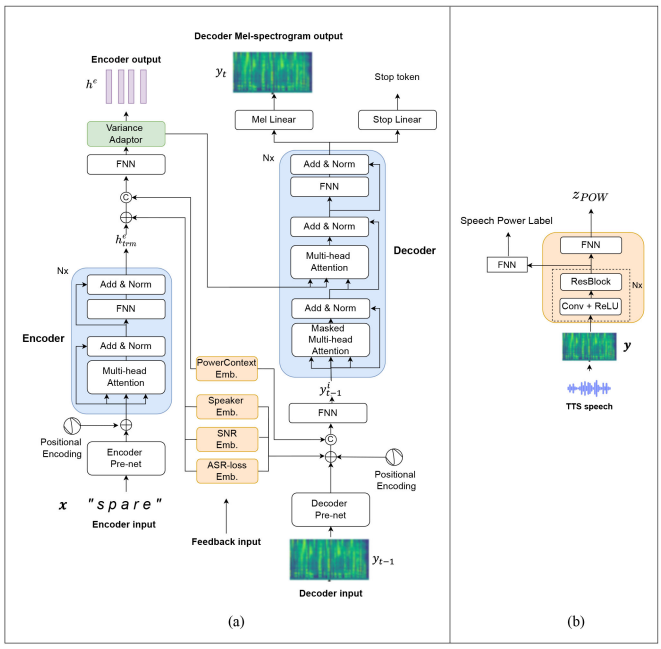

Fig. 6. (a) Proposed PCD-ITTS structure with (b) power context embedding module.

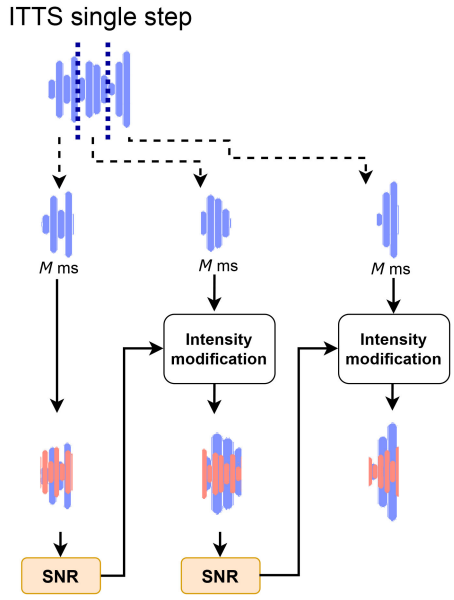

Fig. 7. ITTS speech intensity post adaptation in a noisy condition. The process is done incrementally with an M-ms unit.

we compute the feedback embedding from the first incremental step's speech and use it to synthesize the next W words of speech. For the third incremental step and beyond, we repeat the same process by taking the previous step's output as the feedback.

ITTS not only has to speak more loudly when noises begin but also continue to speak loudly while the noise remains. SNR information, which contains the environmental information, only reveals the ratio between the speech intensity and noise intensity, and it might be insufficient to continuously induce ITTS to speak with the Lombard effect. Therefore, we construct two ITTS systems based on how the system treats the intensity-based context.

1) Power-Context Dependent ITTS (PCD-ITTS): The proposed ITTS architecture, shown in Fig. 6(a), is based on the non-incremental TTS structure with SNR embedding, ASR-loss embedding, and variance adaptor (see Sec. III-A3). In addition to SNR and ASR-loss embeddings, ITTS also takes a power-context embedding (Fig. 6(b)) that contains the intensity information of the previous speech output. By using the intensity cues along with the auditory feedback, we aim to help ITTS control the speech better. This not only helps to control the intensity but also the Lombard speech in overall.

Power-context embedding takes ITTS speech without noise from the previous incremental step and then outputs an embedding vector representing the intensity information. This module consists of the convolution network layers. Before ITTS training, we pre-train the power-context embedding as a speech intensity or power recognition model.

Inside the ITTS, feedback embeddings are used to compute the encoder output h^e, written as

$$z = z_{SPK} + z_{SNR} + z_{ASR}, \tag{15}$$

$$h^e = Var\ Adaptor(FNN([h^e_{trm} + z, z_{POW}])), \tag{16}$$

and also the decoder first transformer layer input

$$y^i_{t-1} = FNN([prenet(y_{t-1}) + PE + z, z_{POW}]), \tag{17}$$

where z_{POW} is the power context embedding vector. To produce accurate embedding, we pre-train all feedback components for incremental tasks.

2) Power-Context Independent ITTS (PCI-ITTS): PCI-ITTS architecture is similar to PCD-ITTS but without the power context embedding. The SNR and ASR-loss embedding vectors are also generated from the noisy speech segment in the previous step and are utilized as feedback to generate the speech segment in the current step.

B. Intensity Post Adaptation

The proposed ITTS has an adaptation delay when adapting to the environment. ITTS synthesizes a speech segment in an incremental step by looking at the speech generated in the previous step. This implies the adaptation is delayed by one incremental step. In an environment with increasing noise, the unadapted speech segment could have low audibility.

To remedy this problem, we apply a straightforward additional intensity modification after the ITTS synthesizes the speech segment, done incrementally on the M-ms unit for each ITTS incremental step (Fig. 7). First the system plays the M-ms speech segment and then it estimates the SNR of that segment, which has been fused with noises. The SNR of the latest speech segment is then used to improve the next M-ms speech part and so on. Intensity modification is done when the SNR is below a pre-defined threshold.

C. Training Method

The proposed ITTS is trained with a similar method as the proposed non-incremental TTS, but with two differences. First, the feedback loop is done once. The SNR embedding vector

is pre-computed from the speech data in the training materials, while the ASR-loss embedding vector is computed based on the ITTS speech segment generated in the earlier incremental step. Second, the speech synthesis in noise to produce the Lombard speech is learned through two cases. The first case is when the pre-computed auditory feedback is generated from noisy normal speech. In this case, ITTS tries to learn how to change the speech prosody attribute from normal to Lombard speech, and the training method is similar to the non-incremental version. The second case is when the pre-computed auditory feedback is generated from noisy Lombard speech. Here, ITTS learns how to maintain or improve the prosody attributes while the environment is still noisy.

V. Dataset

Our systems are constructed using normal speech, normal speech with additive noise (noisy normal speech), and Lombard speech datasets. Since the availability of Lombard speech data is limited, we constructed a synthetic Lombard speech dataset by observing the natural Lombard speech and modifying the normal speech into Lombard speech.

Our experiment was based on the Wall Street Journal (WSJ) corpus [25]. TTS training was done based on three static noise conditions containing noises from (1) clean, (2) SNR 0 dB, and (3) SNR -10 dB conditions, where SNR is relative to normal speech of 44.44 dB in WSJ. The proposed TTS with a variance adaptor was also trained using the character-level prosody attribute labels. These labels were generated by first extracting a character-level speech timing using Montreal forced-alignment toolkit [46]. The details of the data we constructed from WSJ and then utilized for model training are given below.

A. Normal Speech

The WSJ dataset consists of multi-speaker English speeches recorded by reading news text in a clean condition, sampled at 16 kHz. We utilized the *SI-284*, *dev93*, and *eval92* sets as the training, development, and evaluation sets. The *SI-284* set consists of 81 hours of speech. The average speech intensity in WSJ utterances was 44.44 dB. A speech utterance length is 7.88 sec and 17 words on average.

B. Normal Speech With Additive Noise

We combined the WSJ normal speech with noisy sounds to train our system. The noises were white[2] and restaurant babble[3] noises with SNR levels of 0 dB and -10 dB relative to the WSJ speech. This dataset was mainly utilized to train the SNR recognition model and ASR.

C. Natural Lombard Speech

To learn how human vocalization changes in noisy conditions, we recorded natural Lombard speech with a single male speaker who read the WSJ *dev93* and *eval92* set transcriptions in noisy conditions. The noises in the recording were the same noises generated for our normal speech dataset with additive noise. The noise level is considered constant within an utterance. Given only the noise signals, the speaker read aloud the WSJ text as if it were aimed at someone in a noisy condition. Then, we estimated prosody attributes from the normal and the Lombard speech in phoneme-level detail, and the averages of these values can be seen in Table I. For comparison, we also recorded the normal speech in the clean environment as well as in the dynamic noise environments with the same speaker.

TABLE I
Statistic of the Natural Lombard Speech Spoken by a Single Male Speaker

Noise source condition	Noise Intensity (dB)	Speech		
		Intensity (dB)	Pitch (Hz)	Speaking rate (words/sec)
Clean	-	56.92	124.63	2.05
SNR0	44.44	59.73	132.56	1.99
SNR-10	54.44	63.68	143.23	1.93

D. Synthetic Lombard Speech

Next, based on the prosody attribute changes observed in the recorded Lombard speech, we constructed synthetic Lombard speech of a full set of WSJ data. Synthetic Lombard speech was made by modifying the original WSJ speech pitch, duration, and intensity.[4] First, since the WSJ speech consists of multi-speaker data, to maintain the speaker characteristics, we modified the speech pitch and duration based on the attribute shift between the clean and noisy conditions based on the statistic as shown in Table I. The modification was done by, first, aligning the Lombard speech and the normal speech in our recording data, and then estimating the attribute shifts at phoneme-levels. Next, we estimate the pitch and duration of normal WSJ speech at phoneme-level. Then, based on the database of the prosody shifts in the recording data, the target noise, and the value of the WSJ normal speech prosody attributes, we estimate the target pitch and the duration and modify the corresponding normal speech using SoX commands. After that, we modified the speech intensity into a target SNR of 20 dB relative to the noise level. The maximum intensity of the resulting Lombard speech was 75 dB to avoid clipping. This dataset was utilized as the target Lombard speech in the TTS training.

VI. Experiment

A. Experiment Setting

1) TTS Model: Our TTS model consists of a transformer-based encoder and autoregressive decoder. The TTS input was the character sequence, and the output was the 80 dimensions of the Mel-spectrogram. The encoder character embedding layer consists of 256 units, followed by an encoder pre-net that consists of three convolution layers. In the decoder part, the decoder

[2]Generated using white-noise-generator toolkit ([Online]. Available: https://github.com/jannispinter/white-noise-generator)

[3]From the noise sounds in AURORA-2 corpus [47]

[4]The speech pitch, duration, and intensity were modified using the SoundExchange (SoX) toolkit ([Online]. Available: http://sox.sourceforge.net/).

pre-net consists of three linear layers. For both the encoder and decoder, the transformer module consists of six transformer blocks with a dimension of 512, eight attention heads, and a feed-forward inner dimension size of 2048.

The ITTS model configuration, as well as the feedback components architecture, was the same as for the non-incremental TTS. The ITTS incremental unit was three words with the previous ten words as the look-back input and the next two words as the look-ahead input. In the training material, the speech segment length in an incremental unit was 1.40 sec on average. Word sequence was converted into a character sequence before given to the ITTS.

2) ASR Model: Our ASR takes a sequence of speech Mel-spectrogram input to predict its character level transcription. The model configuration follows a similar configuration to the big model proposed in Speech-Transformer [48]. It consisted of twelve encoder layers and six decoder layers. The transformer dimension was 512 with the feed-forward inner dimension of 2048. The attention module consisted of multi-head self-attention with four heads. We prepared two ASR systems to evaluate the TTS in the ASR objective measure. One was trained in a clean condition only and the other one was trained under multiple conditions (mixed clean and noisy speech data). The multi-condition training ASR was also used to compute the TTS ASR feedback. ASR feedback for non-incremental TTS was generated using a standard utterance-level ASR.

In the ITTS, we utilized the ASR trained on short speech segments in which we treat a speech segment as an utterance. The segment length in ASR training was randomized in a range from one to five words.

3) SNR Recognition Model: The SNR recognition model consisted of four stacks of convolution and residual blocks and a linear layer. As mentioned earlier, we experimented on two SNR recognition tasks: classification and regression. The difference between those models lies in the output layer size. The SNR classification model output layer dimension was three based on the number of SNR classes: SNR 0 dB, SNR -10 dB, and clean (no noise). The SNR regression model output layer dimension was one, and the SNR level was output as a real number scaled in the range of -1 to 1.

The SNR recognition model for the non-incremental TTS was trained to recognize the average SNR of a noisy speech utterance. For the ITTS, we also trained the model on short speech segments. Here, the speech segment length was randomized among lengths of one to five words.

4) Power Recognition Model: The speech power recognition model was first trained for the intensity regression task. It consisted of four stacks of convolution and residual blocks and a linear layer. Before ITTS training, this module was trained to do short speech intensity recognition, where the speech length was randomized among lengths of one to five words. The training label was the speech intensity scaled in the range of -1 to 1.

5) Intensity Post Adaptation: The intensity post adaptation incremental unit was 200 ms. We modified the speech intensity to reach SNR 20 dB in noisy conditions using the SoX toolkit. SNR estimation was done using the SNR regression model trained on short speech segments. This model is the same model utilized as the SNR embedding module in the proposed ITTS. In this paper, we do not take the computation time into account in the speech performance evaluation.

B. Non-Incremental TTS in Static Noise Conditions

1) Overall Comparison: In this study, we focused on evaluating TTS speech intelligibility in clean and noisy conditions, and the results in ASR character error rate (CER) are shown in Table II. All TTS systems generated speech using speaker embedding extracted from the normal natural speech in our recording data. The clean condition testing was done using TTS speech without noise, and the noisy condition testing was done using noise signals of the corresponding SNR condition. SNR levels here are the initial SNR conditions before adaptation. The proposed TTS with 'SNR (cls)' generated SNR feedback based on SNR classification, while the system with 'SNR (reg)' was based on SNR regression. We allow the proposed TTS to refine the speech in five feedback iterations at most, and we present the speech that achieved the lowest ASR loss. We compared our proposed systems with several baselines: (1) the standard TTS trained using normal speech from a clean condition, in which the speech in noisy testing was merged with the noise without any modification; (2) the rule-based modification into the Lombard speech, in which the original output of the standard TTS was modified by the same method as that used in the synthetic Lombard WSJ speech construction; (3) the standard TTS that was fine-tuned to Lombard speech [22]; and (4) the standard TTS trained on normal and Lombard speech. Note that these systems did not have feedback components and were also trained based on static noise conditions. The topline speech is the natural clean and Lombard speech produced by a human. We also included synthetic modifications from the natural human speech.

From the baseline results, we found that the speech CER of the standard TTS from clean condition training could be reduced by post-processing the speech prosody into Lombard speech-like. We also obtained further improvement by fine-tuning the standard model using Lombard speech data. However, by incorporating SNR and ASR feedback together, the proposed models were able to outperform the fine-tuned baseline models and more closely approached the CER of topline human speech. In this experiment, the best TTS performance was achieved by the proposed TTS with SNR classification-based feedback, ASR feedback, and a variance adaptor.

2) Effect of SNR and ASR-Loss Embedding on TTS Performance: Using the best proposed system, we analyzed how the SNR and ASR-loss embedding affected the TTS performance. To clarify this, we experimented on the various embedding coefficients to scale the SNR and ASR-loss embedding when they were combined into TTS encoder output h^e and the decoder's first transformer layer input y^i_{t-1}. From the results shown in Fig. 8, SNR feedback alone is shown to be sufficient to improve the speech intelligibility in noise, but it did not result in the best performance in our setting. Using only ASR feedback could result in the best TTS speech when the condition is clean. This shows that ASR feedback also contributed to speech

TABLE II
AVERAGE TTS SPEECH INTELLIGIBILITY (CER %) AT DIFFERENT SNR LEVELS IN BABBLE- AND WHITE-NOISE CONDITIONS USING CLEAN- AND MULTI-CONDITION TRAINING ASR. SNR LEVELS DENOTE THE SNR CONDITION BEFORE ADAPTATION WAS PERFORMED

System	Clean condition training ASR			Multi-condition training ASR		
	Clean	SNR 0	SNR -10	Clean	SNR 0	SNR -10
Baseline TTS						
Standard TTS (Clean)	18.92	118.72	106.25	18.32	70.54	77.07
+ Rule-based modification	18.92	102.96	104.69	18.32	43.25	55.79
+ Fine-tuning (SNR 0 + SNR -10)	13.58	68.53	94.75	14.82	21.99	37.41
Standard TTS (Clean + SNR 0 + SNR -10)	11.04	114.36	102.83	12.89	56.57	70.41
Proposed TTS						
TTS in speech chain framework	11.04	114.36	102.83	12.89	56.57	70.41
+ SNR (cls)	10.21	83.15	101.41	11.58	22.82	42.00
+ SNR (cls) + ASR	10.76	52.51	87.72	12.55	16.11	25.62
+ SNR (cls) + ASR + variance adaptor	10.47	55.70	92.75	11.99	14.70	24.96
+ SNR (reg) + ASR + variance adaptor	12.63	66.84	86.41	13.52	18.57	31.19
Topline (human natural speech)						
Normal speech	5.77	92.56	98.98	7.43	22.17	58.81
+ Rule-based modification	5.77	58.40	67.78	7.43	13.24	15.15
Lombard speech	5.77	25.38	59.25	7.43	11.46	20.46

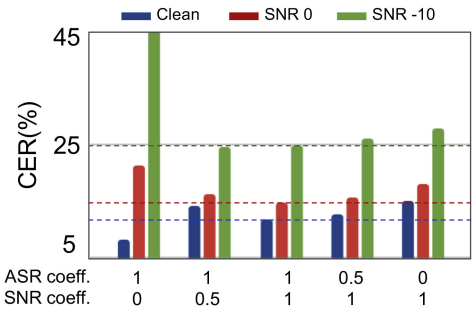

Fig. 8. Effect of auditory feedback on the TTS speech intelligibility based on the embedding coefficients.

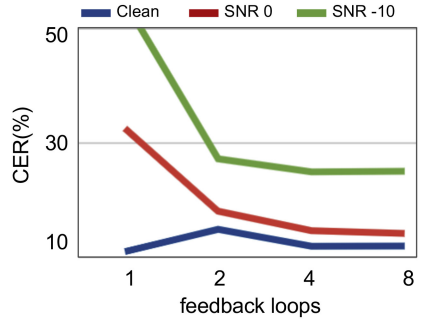

Fig. 9. Proposed TTS speech intelligibility in different numbers of feedback loop iterations.

enhancement. But when the environment becomes noisy, SNR feedback becomes critical to the system. Here the optimum performance is when the SNR embedding and ASR-loss embedding coefficients are equal to one, indicating that both feedbacks are crucial to the Lombard effect by TTS.

The number of feedback loop iterations also affected our system. Interestingly, the training loop consisted of only two iterations, but in inference, a higher number of loops resulted in better speech intelligibility as shown in Fig. 9. Here, for each inference iteration, TTS continuously received ASR-loss embedding and SNR embedding based on the speech that needs to be improved, from which the TTS obtained the current intelligibility information, leading to better speech performance along with the increased number of loops. Humans during conversation, by comparison, might speak with the Lombard effect in several trials so that their speech could be heard over the noise. Our results reveal that a machine can also dynamically adapt in several loop iterations, similar to a re-speaking attempt.

3) Speech Prosody Attributes: We analyzed the improvement in speech prosody attributes in our Lombard TTS as well as in human Lombard speech. Here, we focus on improving speech intensity, pitch, and duration. The value of TTS speech prosody attributes are shown in Table III with a visualized example in Fig. 10. The results show that our proposed TTS produced speech with a higher intensity and longer duration in a noisy environment than the normal speech. The adaptation also included an increase in the pitch compared to the normal speech, which is also shown in Fig. 11. We also evaluated the F0 mean squared-error (MSE) to compare the TTS and human speech in Table IV. The best MSE was achieved by the proposed TTS with 'SNR (reg)' feedback, ASR feedback, and a variance adaptor, showing that it created synthesized Lombard speech pitch that is closer to human Lombard speech.

The systems with SNR and ASR feedback show a higher improvement than the system with SNR feedback only. They produce speech with higher intensity, pitch, and duration or a slower speaking rate. Here TTS with 'SNR (reg)' resulted in more dynamic intensity and pitch than the TTS with 'SNR (cls).' This could be related to the model's output precision in representing the SNR, which is discussed in section VI.C in more detail. In Table II, TTS with 'SNR (cls),' ASR feedback, and variance adaptor show higher intelligibility because its Lombard speech was louder than that from TTS with similar auditory feedback using 'SNR (reg).'

Interestingly, our proposed TTS Lombard speech was louder than the human Lombard speech, but human speech had better

TABLE III
AVERAGE VALUE OF TTS PROSODY ATTRIBUTES

System	Intensity (dB)			Pitch (Hz)			Speaking rate (words/sec)		
	Clean	SNR 0	SNR -10	Clean	SNR 0	SNR -10	Clean	SNR 0	SNR -10
Baseline TTS									
Standard TTS (Clean)	43.58			120.98			3.37		
+ Fine-tuning (SNR 0 + SNR -10)	66.30			122.62			3.05		
Proposed TTS									
TTS + SNR (cls)	45.59	61.01	61.10	123.79	123.31	123.09	3.14	3.07	3.07
TTS + SNR (cls) + ASR + variance adaptor	43.77	67.28	67.37	116.80	123.69	123.86	3.35	2.99	3.00
TTS + SNR (reg) + ASR + variance adaptor	54.32	62.65	64.20	122.81	124.88	125.15	2.93	2.80	2.79
Topline (human natural speech)									
Normal/Lombard speech	56.92	59.73	63.68	124.63	132.56	143.23	2.05	1.99	1.93

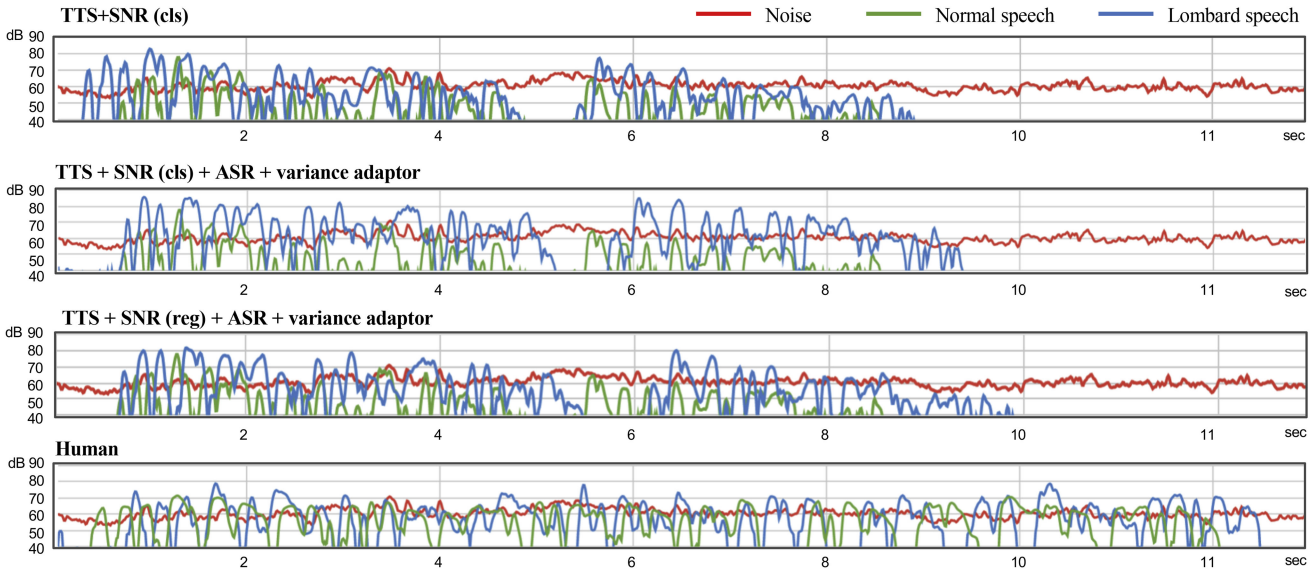

Fig. 10. Intensities of the normal speech and Lombard speech produced by human and TTS in the babble-noise condition. The speech transcription is the same.

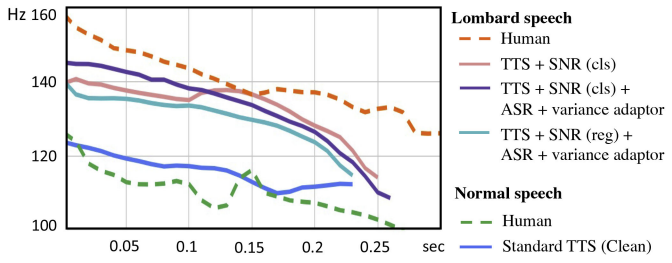

Fig. 11. Normal and Lombard speech pitch of the word "*ruling*" produced by human and TTS. The Lombard speech was produced in a babble noise with an intensity of 60.35 dB. Natural speech was spoken by a male speaker. TTS speech was generated using a speaker embedding of the same speaker.

TABLE IV
F0 MSE BETWEEN TTS SPEECH AND NATURAL SPEECH

System	Clean	SNR 0	SNR -10
Baseline TTS			
Standard TTS (Clean)	0.231	0.283	0.380
+ Fine-tuning (SNR 0 + SNR -10)	0.216	0.278	0.368
Proposed TTS			
TTS + SNR(cls)	0.256	0.318	0.406
TTS + SNR(cls) + ASR + variance adaptor	0.239	0.296	0.369
TTS + SNR(reg) + ASR + variance adaptor	0.214	0.266	0.367

intelligibility. Here human spoke more slowly with a higher pitch than TTS. This shows that the simultaneous enhancement of these three attributes is necessary. Our proposed TTS speech was shorter than human speech, perhaps due to the speaking rate difference between the speech in WSJ training materials and our natural Lombard speech data. In natural speech, the Lombard effect is not simply a temporal envelope expansion from normal speech and it produce more pronounced amplitude modulations in noise [49], which is not limited to intensity, pitch, and duration enhancement. This might also be the reason for the performance difference between human and TTS speech, since our TTS was trained using Lombard speech with a focus on prosody improvement. Speech naturalness might have also contributed to intelligibility.

The proposed TTS in the machine speech chain framework successfully improved the speech intelligibility in noisy conditions. However, a high adaptation delay still occurs because the feedback is processed at the utterance level. Thus, if the environment becomes noisier in the middle of an utterance, TTS has to wait for the utterance to finish to begin the adaptation. In the next experiment, we focus on ITTS with machine speech chain

TABLE V
AVERAGE TTS SPEECH INTELLIGIBILITY (CER%) IN BABBLE- AND WHITE-NOISE CONDITIONS BASED ON MULTI-CONDITION TRAINING ASR. SNR EMBEDDING IN THE PROPOSED SYSTEMS WAS GENERATED USING SNR CLASSIFICATION (CLS) OR REGRESSION (REG)

System	Static Noise			Dynamic Switch Noise		Dynamic Smooth Noise	
	Clean	SNR 0	SNR -10	Clean, SNR 0, SNR -10	SNR 0, Clean, SNR -10	Clean, SNR 0, SNR -10	SNR 0, Clean, SNR -10
Baseline Non-incremental TTS							
Standard TTS (Clean)	18.32	70.54	77.07	53.64	49.77	47.39	45.11
+ Fine-tuning (SNR 0 + SNR -10)	14.82	21.99	37.41	20.30	20.33	19.41	18.74
Proposed Non-incremental TTS							
TTS + SNR (cls) + ASR + var.adaptor (speak 5x)	11.99	14.70	24.96	61.94	70.98	28.09	17.88
TTS + SNR (reg) + ASR + var.adaptor (speak 1x)	14.76	32.91	56.42	27.88	26.60	28.48	28.22
+ intensity post adaptation	14.76	21.43	27.23	20.32	20.57	20.22	19.70
TTS + SNR (reg) + ASR + var.adaptor (speak 5x)	13.52	18.57	31.19	16.95	18.70	18.00	17.57
+ intensity post adaptation	13.52	16.16	22.37	14.54	14.62	13.94	12.97
Proposed Incremental TTS (ITTS)							
PCI-ITTS + SNR (reg) + ASR + variance adaptor (speak 1x)	18.96	38.26	60.64	34.90	32.00	34.02	33.69
PCD-ITTS + SNR (reg) + ASR + variance adaptor (speak 1x)	14.42	23.32	41.89	26.96	28.13	23.48	22.53
+ intensity post adaptation	14.42	20.59	31.05	20.64	17.30	20.10	20.99
Topline (human natural speech)							
Normal speech	7.43	22.17	58.81	32.10	32.93	15.04	14.97
+ Rule-based modification	7.43	13.24	15.15	22.41	23.25	12.37	12.60
Lombard speech	7.43	11.46	20.46	22.92	17.77	-	-

to start the adaptation with a short delay, i.e., approximately one sec or three words in our setting.

C. ITTS in Dynamic Noises

We evaluated our systems under two types of dynamic noise conditions: (1) switch noise and (2) smooth noise, as well as in the static noise condition. In switch noise, noise intensity changed without transition, while in smooth noise conditions the noise intensity changed gradually. The baseline, the proposed non-incremental TTS, and the proposed ITTS intelligibility measured in ASR CER are shown in Table V.

First, we evaluated the proposed non-incremental TTS intelligibility in dynamic noises. We ran our TTS to speak five times at most, assuming that the noise conditions in all re-speaking attempts were the same. From the results, the non-incremental TTS with SNR classification feedback could not perform well in dynamic switch noise. Meanwhile, the system with SNR regression feedback was more robust; it synthesized a highly intelligible speech in dynamic noises. This is because the SNR regression model could recognize the SNR in dynamic noise conditions more accurately as a real value than the SNR classification model. SNR classification categorized the dynamic noises into pre-defined classes, which might not represent the actual SNR level. For this factor, in further experiments, we focus only on systems with SNR regression feedback.

In the proposed incremental speech synthesis without the intensity post adaptation, PCD-ITTS produced a more intelligible speech than the PCI-ITTS. This indicates that power context embedding in PCD-ITTS is critical to maintaining speech intelligibility. When we manually inspect the utterances made under the static noise conditions, we hear an intensity fluctuation in the PCI-ITTS speech. For example, PCI-ITTS speaks with a Lombard effect at the initial incremental step more loudly than the noise, resulting in an SNR of 20 dB. In the next step, it produces a speech segment with a reduced intensity, since the SNR might suggest a less noisy condition. Meanwhile, the PCD-ITTS tracks the previous speech intensity so that the system has better control of the speech.

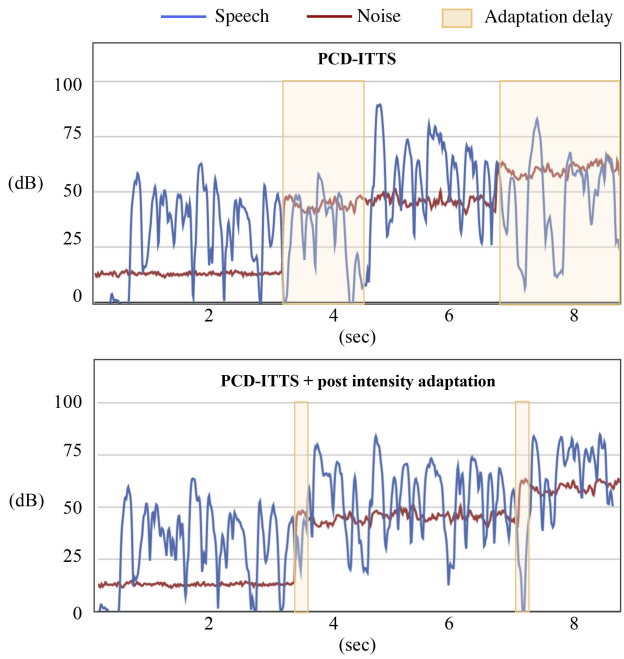

Fig. 12. PCD-ITTS speech intensity with and without the intensity post adaptation with a 200-ms incremental unit in the dynamic switch noise condition.

When intensity post adaptation was not applied in all systems, PCD-ITTS in static noises performed closely to or better than the proposed non-incremental TTS that spoke only once. However, in dynamic noise, PCD-ITTS suffered from adaptation delay for one incremental step (Fig. 12), resulting in the unadapted speech part having a low speech intelligibility. The average speech segment length produced by our system in an incremental step was 1.11 sec on average. By applying an intensity post adaptation, we were able to reduce the adaptation delay and thus improved the PCD-ITTS intelligibility.

TABLE VI
SHORT-TERM OBJECTIVE INTELLIGIBILITY (STOI)

System	Static Noise		Dynamic Switch Noise	Dynamic Smooth Noise
	SNR0	SNR-10		
Baseline Non-incremental TTS				
Standard TTS(Clean)	49.69	38.07	65.13	71.93
+ Fine-tuning (SNR0 + SNR -10)	83.56	71.07	89.97	92.10
Proposed TTS + SNR (reg) + ASR + variance adaptor				
Non-incremental TTS	87.57	76.08	91.76	92.17
Incremental TTS (PCI-ITTS)	82.62	65.87	79.00	76.44
Incremental TTS (PCD-ITTS)	89.95	77.04	84.31	82.71
+ intensity post adaptation	93.00	82.85	94.46	94.58
Topline (human natural speech)				
Normal speech	60.31	47.57	76.06	88.88
+ Rule-based modification	90.48	86.99	84.34	93.84

TABLE VII
SUS INTELLIGIBILITY EVALUATION RESULTS IN CER (%) ($\star$: STATISTICALLY DIFFERENT FROM THE BASELINE IN THE SAME ENVIRONMENT)

System	Noise	Objective (ASR)	Subjective (Human)
Baseline Non-incremental TTS			
Fine-tuning (SNR 0 + SNR -10)	Clean	21.76	6.74
	Static	35.31	10.80
	Dynamic	27.69	7.44
Proposed TTS + SNR (reg) + ASR + variance adaptor			
Non-incremental TTS	Clean	14.72	4.94 $\star$
	Static	16.94	5.78 $\star$
	Dynamic	14.29	4.94 $\star$
Incremental TTS (PCD-ITTS + intensity post adaptation)	Clean	17.73	6.05
	Static	26.26	8.58 $\star$
	Dynamic	24.14	7.92

D. Short-Term Objective Intelligibility Measure

Next, we evaluated our TTS speech intelligibility based on the short-term objective intelligibility (STOI) measure [50]. STOI estimates the temporal envelope correlation between the speech signal disturbed by noise and the reference speech signal before being disturbed. A higher correlation indicates a higher speech signal intelligibility. We estimated the noisy speech STOI by using the noisy speech as the disturbed speech and the speech before integration with the noise signal as the reference speech. The noisy condition evaluated here are the static noises and the dynamic noises consisting of transitions from clean, SNR 0 dB, and then SNR -10 dB. The SNR conditions here reflect the SNR before the adaptation was performed, which has a magnitude relative to that of normal speech with an intensity of 44.44 dB.

Table VI shows the STOI measurement of our system's Lombard speech under the noisy condition. As expected, the standard TTS trained using normal speech from the clean condition has the lowest STOI. The proposed non-incremental TTS and PCD-ITTS with intensity post adaptation show an improvement in STOI. Interestingly, PCD-ITTS had a better STOI than the non-incremental TTS, but for intelligibility in ASR CER this relationship was reversed. Our analysis suggests that this is related to the speech intelligibility as signal and sentence. PCD-ITTS speech signals before and after disturbance with noises show a high correlation, implying the speech signal is audible in noises, for example, because the speech is very loud. But its comprehensibility as a sentence is not as high as the non-incremental TTS. This is because the proposed non-incremental TTS was allowed to synthesize the speech by using a complete sentence's text and sentence-level feedback with the re-speaking. PCD-ITTS with intensity post adaptation achieved a close ASR CER and higher STOI to the non-incremental TTS, even though PCD-ITTS did not perform re-speaking. This illustrates how speech adaptation within a short time improved the speech intelligibility in incremental speech synthesis.

E. Subjective Evaluation

In the next experiment, we evaluated our system through a subjective evaluation on speech intelligibility and naturalness. First, a speech intelligibility test was carried out by asking the human listener to write a transcription of noisy speech. In this test, we used semantically unpredictable sentence (SUS) [51] as the TTS input text. SUS is a syntactically correct but semantically unpredictable sentence. This ensures that the listener does not guess the unintelligible speech based on the sentence context. Then the second test, a speech-naturalness evaluation, was done through a mean opinion score (MOS) test by asking the listener to score the speech naturalness on a scale of 1-5 points. The speech signals were also mixed with noises. The sentences used in the MOS test were the normal sentences obtained from the WSJ evaluation set. The intelligibility and MOS tests were done through crowd-sourcing with 61 participants for the intelligibility test and 138 participants for the MOS test. All participants were located in the United States.

In this work, we mainly focused on improving TTS speech intelligibility. In related work, it has been suggested that speech intelligibility and naturalness do not always imply each other [52], and thus improvement in intelligibility might not necessarily improve naturalness. In overall, our subjective evaluation results revealed that the proposed systems achieved a significant improvement in speech intelligibility while also preserving speech naturalness. The details are below.

1) Speech Intelligibility: The SUS intelligibility test results in CER are shown in Table VII, and they are also visualized in Fig. 13. We conducted a statistical t-test to show the significance of the improvement in the proposed system by comparing this system to the fine-tuned baseline system in the same environment. The significance level was 0.05. In Table VII, the systems with a statistically different result against the baseline are marked with a star "$\star$". TTS speech was generated based on three noisy conditions: (1) clean, (2) static noise from the SNR -10 dB condition, and (3) dynamic smooth noise consisting of noise transitions from clean, SNR 0 dB, and then SNR -10 dB noises. We also present the ASR CER as the objective measure. Here, the PCD-ITTS was applied with intensity post adaptation to overcome the adaptation delay. We did not use the intensity post adaptation in the non-incremental system to see how our basic framework would perform, with the speech improvement solely done within the TTS. Based on the evaluation results, in the clean condition, the proposed non-incremental TTS was more intelligible than the other systems, while the PCD-ITTS and the baseline TTS performed similarly. In the static noise condition, all proposed systems were also more intelligible than the baseline. In the dynamic noise condition, the proposed non-incremental TTS showed the best intelligibility performance.

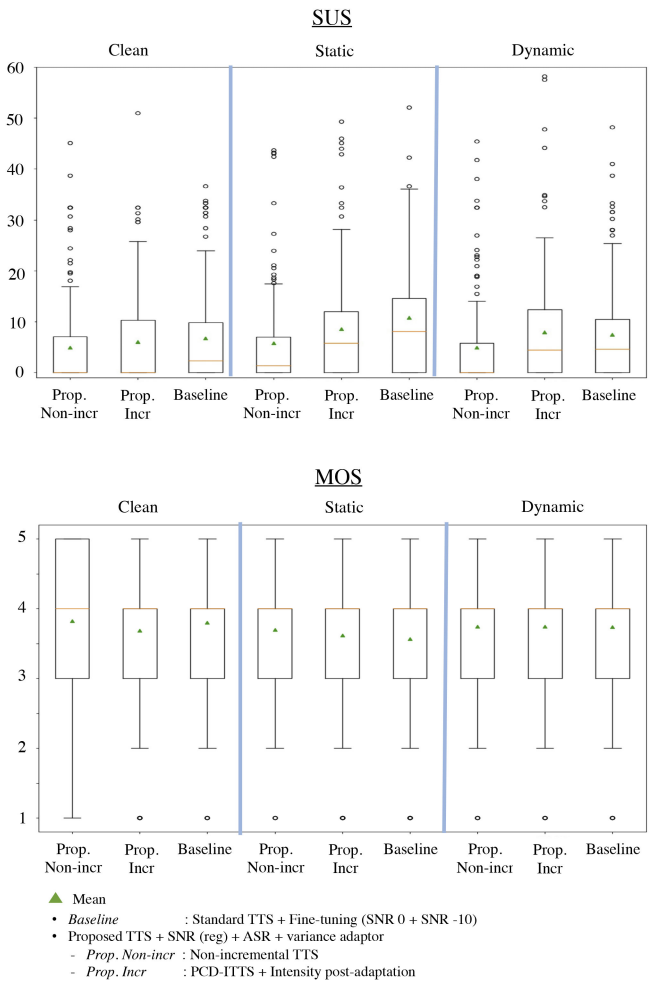

Fig. 13. SUS intelligibility and MOS naturalness scores.

TABLE VIII
MOS EVALUATION RESULTS (⋆ : STATISTICALLY DIFFERENT FROM THE BASELINE IN THE SAME ENVIRONMENT)

System	Noise	MOS
Baseline Non-incremental TTS		
Fine-tuning (SNR 0 + SNR -10)	Clean	3.80
	Static	3.56
	Dynamic	3.74
Proposed TTS + SNR (reg) + ASR + variance adaptor		
Non-incremental TTS	Clean	3.82
	Static	3.70
	Dynamic	3.74
Incremental TTS (PCD-ITTS + intensity post adaptation)	Clean	3.69
	Static	3.61
	Dynamic	3.75

2) Speech Naturalness: The MOS scores are shown in Table VIII and in Fig. 13. We performed a Mann-Whitney U statistical test whose results show that the presented systems have statistically similar MOS scores, indicating that they preserved the naturalness. Here the proposed non-incremental TTS achieved the highest average score in general. PCD-ITTS naturalness was lower than that of the proposed non-incremental TTS. When we inspected the audio, the naturalness degradation was mostly caused by speech discontinuities in the ITTS speech, which often occur in incremental speech synthesis. But by incorporating feedback into the system, our ITTS achieved higher average scores under noisy conditions than did the non-incremental baseline system that also speaks once and loudly. This also demonstrates that auditory feedback has a positive impact on online speech synthesis, similar to human real-time speech production

VII. CONCLUSION

We constructed a dynamically adaptive machine speech chain inference framework to support TTS under noisy conditions. We proposed two TTS systems with auditory feedback: non-incremental TTS and incremental TTS (ITTS). Our proposed systems with auditory feedback and a variance adaptor successfully produced highly intelligible speech that surpassed a standard TTS with a fine-tuning method and achieved results closer to human performance. The non-incremental TTS in the machine speech chain framework achieved the best performance by refining the speech utterance more than once. On the other hand, our ITTS in the machine speech chain framework was able to produce highly intelligible speech by performing dynamic adaptation within an utterance according to the environmental changes with a short delay, thus enabling the TTS to more closely resemble the human speech chain and thus improve the speech quality. Our experimental results reveal that dynamic adaptation with auditory feedback could be an essential tool for optimal speech generation by machines, and not only for human speech production. Our current system focuses on the Lombard effect with prosodic adaptation in a known noisy environment.

The proposed systems still have performance limitations compared to human natural speech. To improve this, in future work, we would like to collect multi-speaker natural Lombard speech data for a better Lombard speech analysis in the synthetic Lombard speech construction, since Lombard effect is speaker- and gender-dependent [53]. We also intend to carry out TTS training using the natural Lombard speech data. Furthermore, we intend to consider a better speech modification approach than the current prosody modification with SoX. We currently generate the Mel-spectrogram using the proposed TTS with the Griffin-Lim vocoder, therefore, we could also expect more improvement using an advanced neural vocoder, such as WaveRNN [54] or HiFi-GAN [55], which we plan to investigate in future work. In the next task, we are also interested in TTS with dynamic spectral adaptation and amplitude modulations in Lombard speech as well as TTS adaptation in an unseen noise condition.

REFERENCES

[1] P. B. Denes and E. N. Pinson, *The Speech Chain: The Physics and Biology of Spoken Language* (Science/Communication Series). New York, NY, USA: W. H. Freeman, 1993.
[2] F. Guenther, "Cortical interactions underlying production of speech sounds," *J. Commun. Disord.*, vol. 39, pp. 350–65, Sep. 2006.
[3] A. Lind, L. Hall, B. Breidegard, C. Balkenius, and P. Johansson, "Auditory feedback of one's own voice is used for high-level semantic monitoring: The self-comprehension hypothesis," *Front. Hum. Neurosci.*, vol. 8, 2014, Art. no. 166.
[4] N. E. Scheerer and J. A. Jones, "The role of auditory feedback at vocalization onset and mid-utterance," *Front. Psychol.*, vol. 9, 2018, Art. no. 2019.

[5] M. Badian, E. Appel, D. Palm, W. Rupp, W. Sittig, and K. Taeuber, "Standardized mental stress in healthy volunteers induced by delayed auditory feedback (DAF)," *Eur. J. Clin. Pharmacol.*, vol. 16, pp. 171–176, 1979.

[6] K. Kurihara and K. Tsukada, "SpeechJammer: A system utilizing artificial speech disturbance with delayed auditory feedback," 2012, *arXiv:1202.6106*. [Online]. Available: http://arxiv.org/abs/1202.6106

[7] J. S. Perkell et al., "Speech motor control: Acoustic goals, saturation effects, auditory feedback and internal models," *Speech Commun.*, vol. 22, pp. 227–250, 1997.

[8] A. Postma, "Detection of errors during speech production: A review of speech monitoring models," *Cognition*, vol. 77, no. 2, pp. 97–132, 2000.

[9] H. Lane and B. Tranel, "The Lombard sign and the role of hearing in speech," *J. Speech Hear. Res.*, vol. 14, no. 4, pp. 677–709, 1971.

[10] M. Garnier, N. Henrich, and D. Dubois, "Influence of sound immersion and communicative interaction on the Lombard effect," *J. Speech Lang. Hear. Res.*, vol. 53, no. 3, pp. 588–608, 2010.

[11] T. Letowski, T. Frank, and J. Caravella, "Acoustical properties of speech produced in noise presented through supra-aural earphones," *Ear Hear.*, vol. 14, pp. 332–338, 1993.

[12] G. K. Anumanchipalli, P. K. Muthukumar, U. Nallasamy, A. Parlikar, A. W. Black, and B. Langner, "Improving speech synthesis for noisy environments," in *Proc. ISCA Workshop Speech Synth.*, 2010, pp. 154–159.

[13] L. Stowe and E. Golob, "Evidence that the Lombard effect is frequency-specific in humans," *J. Acoust. Soc. Amer.*, vol. 135, pp. 640–647, Jan. 2014.

[14] T. Heinks-Maldonado and J. Houde, "Compensatory responses to brief perturbations of speech amplitude," *Acoust. Res. Lett. Online*, vol. 6, no. 3, pp. 131–137, Jul. 2005.

[15] J. Bauer, J. Mittal, C. Larson, and T. Hain, "Vocal responses to unanticipated perturbations in voice loudness feedback: An automatic mechanism for stabilizing voice amplitude," *J. Acoust. Soc. Amer.*, vol. 119, pp. 2363–2371, May 2006.

[16] K. R. A. Foery, "Triggering the Lombard effect: Examining automatic thresholds," Ph.D. dissertation, Department of Speech, Language and Hearing Sciences, Univ. Colorado at Boulder, Boulder, CO, USA, 2008.

[17] Y. Wang et al., "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.

[18] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 6706–6713.

[19] M. Chen et al., "MultiSpeech: Multi-speaker text to speech with transformer," in *Proc. Interspeech*, 2020, pp. 4024–4028.

[20] J. Donahue, S. Dieleman, M. Bińkowski, E. Elsen, and K. Simonyan, "End-to-end adversarial text-to-speech," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://iclr.cc/virtual/2021/oral/3498

[21] T. Raitio, A. Suni, M. Vainio, and P. Alku, "Analysis of HMM-based Lombard speech synthesis," in *Proc. Interspeech*, 2011, pp. 2781–2784.

[22] D. Paul, M. P. Shifas, Y. Pantazis, and Y. Stylianou, "Enhancing speech intelligibility in text-to-speech synthesis using speaking style conversion," in *Proc. Interspeech*, 2020, pp. 1361–1365.

[23] A. Tjandra, S. Sakti, and S. Nakamura, "Machine speech chain," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 976–989, 2020.

[24] S. Novitasari, S. Sakti, and S. Nakamura, "Dynamically adaptive machine speech chain inference for TTS in noisy environment: Listen and speak louder," in *Proc. Interspeech*, 2021, pp. 4124–4128.

[25] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proc. Workshop Speech Natural Lang.*, 1992, pp. 357–362.

[26] M. Cooke, C. Mayo, and C. Valentini-Botinhao, "Intelligibility-enhancing speech modifications: The Hurricane Challenge," in *Proc. Interspeech*, 2013, pp. 3552–3556.

[27] J. Rennies, H. Schepker, C. Valentini-Botinhao, and M. Cooke, "Intelligibility-enhancing speech modifications– The Hurricane Challenge 2.0," in *Proc. Interspeech*, 2020, pp. 1341–1345.

[28] V. Aubanel and M. Cooke, "Information-preserving temporal reallocation of speech in the presence of fluctuating maskers," in *Proc. Interspeech*, 2013, pp. 3592–3596.

[29] D. Erro, T.-C. Zorila, Y. Stylianou, E. Navas, and I. Hernáez, "Statistical synthesizer with embedded prosodic and spectral modifications to generate highly intelligible speech in noise," in *Proc. Interspeech*, 2013, pp. 3557–3561.

[30] R. Takou, N. Seiyama, and A. Imai, "Improvement of speech intelligibility by reallocation of spectral energy," in *Proc. Interspeech*, 2013, pp. 3605–3607.

[31] E. Godoy and Y. Stylianou, "Increasing speech intelligibility via spectral shaping with frequency warping and dynamic range compression plus transient enhancement," in *Proc. Interspeech*, 2013, pp. 3572–3576.

[32] H. Schepker, J. Rennies, and S. Doclo, "Speech-in-noise enhancement using amplification and dynamic range compression controlled by the speech intelligibility index," *J. Acoust. Soc. America*, vol. 138, no. 5, pp. 2692–706, 2015.

[33] F. Bederna et al., "Adaptive compressive onset-enhancement for improved speech intelligibility in noise and reverberation," in *Proc. Interspeech*, 2020, pp. 1351–1355.

[34] A. Suni, T. Raitio, M. Vainio, and P. Alku, "The GlottHMM speech synthesis entry for Blizzard Challenge 2010," in *Proc. Blizzard Challenge Workshop*, 2010, pp. 1–6.

[35] A. Suni, R. Karhila, T. Raitio, M. Kurimo, M. Vainio, and P. Alku, "Lombard modified text-to-speech synthesis for improved intelligibility: Submission for the Hurricane Challenge 2013," in *Proc. Interspeech*, 2013, pp. 3562–3566.

[36] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 1, pp. 66–83, Jan. 2009.

[37] Q. Hu, T. Bleisch, P. Petkov, T. Raitio, E. Marchi, and V. Lakshminarasimhan, "Whispered and Lombard neural speech synthesis," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021.

[38] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[39] Y. Ren et al., "FastSpeech 2: Fast and high-quality end-to-end text to speech," 2020, *arXiv:2006.04558*.

[40] C. Li et al., "Deep Speaker: An end-to-end neural speaker embedding system," 2017, *arXiv:1705.02304*.

[41] Z. Byambadorj, R. Nishimura, A. Ayush, K. Ohta, and N. Kitaoka, "Text-to-speech system for low-resource language using cross-lingual transfer learning and data augmentation," *EURASIP J. Audio Speech Music Process.*, vol. 2021, no. 1, pp. 1–20, Dec. 2021. [Online]. Available: https://doi.org/10.1186/s13636-021-00225-4

[42] G. Chen et al., "Data augmentation for children's speech recognition - the "Ethiopian" system for the SLT 2021 Children Speech Recognition Challenge,", 2020, *arXiv:2011.04547*.

[43] M. Ma et al., "Incremental text-to-speech synthesis with prefix-to-prefix framework," 2019, *arXiv:1911.02750*.

[44] D. Liu, C. Wang, H. Gong, X. Ma, Y. Tang, and J. Pino, "Incremental speech synthesis for speech-to-speech translation," 2021, *arXiv:2110.08214*.

[45] T. Saeki, S. Takamichi, and H. Saruwatari, "Incremental text-to-speech synthesis using pseudo lookahead with large pretrained language model," *IEEE Signal Process. Lett.*, vol. 28, pp. 857–861, 2021.

[46] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: Trainable text-speech alignment using Kaldi," in *Proc. Interspeech*, 2017, pp. 498–502.

[47] D. Pearce and H. G. Hirsch, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy condition," in *Proc. 6th Int. Conf. Spoken Lang. Process.*, 2000, pp. 29–32.

[48] L. Dong, S. Xu, and B. Xu, "Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5884–5888.

[49] H. Bosker and M. Cooke, "Enhanced amplitude modulations contribute to the Lombard intelligibility benefit: Evidence from the Nijmegen corpus of Lombard speech," *J. Acoust. Soc. America*, vol. 147, no. 2, 2020, Art. no. 721.

[50] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, Sep. 2011.

[51] C. Benoît, M. Grice, and V. Hazan, "The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using semantically unpredictable sentences," *Speech Commun.*, vol. 18, no. 4, pp. 381–392, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/016763939600026X

[52] T. Ngo, R. Kubo, and M. Akagi, "Increasing speech intelligibility and naturalness in noise based on concepts of modulation spectrum and modulation transfer function," *Speech Commun.*, vol. 135, pp. 11–24, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016763932100100X

[53] J.-C. Junqua, S. Fincke, and K. Field, "The Lombard effect: A reflex to better communicate with others in noise," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 4, 1999, pp. 2083–2086.
[54] N. Kalchbrenner et al., "Efficient neural audio synthesis," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2410–2419.
[55] J. Kong, J. Kim, and J. Bae, "Hifi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 17022–17033, 2020.

Sashi Novitasari received the B.S. degree in informatics (*cum laude*) from the Bandung Institute of Technology, Indonesia, Bandung, in 2018, and M.E. degree in 2020 from the Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma Japan, where she is currently working toward the Doctoral degree. She was the recipient of the Japanese Ministry of Education, Culture, Sport, Science, and Technology (MEXT) Scholarship. Her research interests include speech recognition, speech synthesis, and spoken language translation systems.

Sakriani Sakti (Member, IEEE) received the B.E. degree in informatics (*cum laude*) from the Bandung Institute of Technology, Bandung, Indonesia, in 1999, and the M.Sc. and the Ph.D. degrees from the University of Ulm, Ulm, Germany, in 2002 and 2008, respectively. During her thesis work, she worked with Speech Understanding Department, DaimlerChrysler Research Center, Ulm, Germany. Between 2003 and 2009, she was a Researcher with ATR SLC Labs, Japan, and during 2006-2011, she was an Expert Researcher with NICT SLC Groups, Japan. While working with ATR-NICT, Japan, she continued her studies from 2005 to 2008) with Dialog Systems Group, University of Ulm. She was actively involved in collaboration activities, such as Asian Pacific Telecommunity Project from 2003 to 2007, A-STAR, and USTAR from 2006 to 2011. From 2009 to 2011, she was a Visiting Professor with the Computer Science Department, University of Indonesia (UI), Indonesia. From 2011 to 2017, she was an Assistant Professor with the Augmented Human Communication Laboratory, NAIST, Japan. She was a Visiting Scientific Researcher of INRIA Paris-Rocquencourt, France, from 2015 to 2016, under JSPS Strategic Young Researcher Overseas Visits Program for Accelerating Brain Circulation. From 2018 to 2021, she was a Research Associate Professor with NAIST and a Research Scientist with RIKEN, Center for Advanced Intelligent Project (AIP), Japan. She is currently an Associate Professor with JAIST, adjunct Associate Professor with NAIST, Visiting Research Scientist with RIKEN AIP, and Adjunct Professor with the University of Indonesia. Her research interests include statistical pattern recognition, graphical modeling framework, deep learning, multilingual speech recognition and synthesis, spoken language translation, affective dialog system, and cognitive-communication. In 2000, she was the recipient of the DAADSiemens Program Asia 21st Century Award to study in Communication Technology, University of Ulm. She is a Member of JNS, SFN, ASJ, ISCA, and IEICE. She is also a Committee Member of IEEE SLTC From 2021 to 2023 and an Associate Editor for the IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING from 2020 to 2023. Furthermore, she is the Chair of ELRA/ISCA Special Interest Group on Under-resourced Languages (SIGUL) and a Board Member of Spoken Language Technologies for Under- Resourced Languages (SLTU).

Satoshi Nakamura (Fellow, IEEE) received the B.S. degree from the Kyoto Institute of Technology, Kyoto, Japan, in 1981, and the Ph.D. degree from Kyoto University, Kyoto, Japan, in 1992. He is currently a Professor of the Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma, Japan, and the Honorary Professor of the Karlsruhe Institute of Technology, Karlsruhe, Germany. He was an Associate Professor with the Graduate School of Information Science, Nara Institute of Science and Technology, from 1994 to 2000 and the department Head and Director of the ATR Spoken Language Communication Research Laboratories from 2000 to 2008 and the Vice President of ATR from 2007 to 2008. He was the Director General of the Keihanna Research Laboratories and the Executive Director of the Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology, Japan from 2009 to 2010 and a Project Leader of the Tourism Information Analytics Team, Center for Advanced Intelligence Project in AIP of RIKEN Institute from 2017 to 2021. He is currently the Director of the Augmented Human Communication Laboratory and a Full Professor of the Data Science Center and Graduate School of Science and Technology, Nara Institute of Science and Technology. His research interests include modeling and systems of speech-to-speech translation, speech recognition, and spoken language processing. He is one of the Leaders of speech-tospeech translation research and has been serving for various worldwide speech-to-speech translation research projects, including C-STAR, IWSLT, A-STAR, and ISCA SIG on Spoken Language Translation. He was the recipient of the LREC Antonio Zampolli Award in 2012, Yamashita Research Award, Kiyasu Award from the Information Processing Society of Japan, Telecom System Award, AAMT Nagao Award, Docomo Mobile Science Award in 2007, ASJ Award for Distinguished Achievements in Acoustics, Commendation for Science and Technology by the Minister of Education, Science and Technology, and Commendation for Science and Technology by the Minister of Internal Affair and Communications. He was an Elected Board Member of the International Speech Communication Association, ISCA in 2011- 2018, an *IEEE Signal Processing Magazine* Editorial Board Member in 2012-2014, an IEEE SPS Speech and Language Technical Committee Member in 2013-2015. He is ISCA Fellow, IPSJ Fellow, and ATR Fellow.