*INTERSPEECH 2021*
30 August – 3 September, 2021, Brno, Czechia

# Dynamically Adaptive Machine Speech Chain Inference for TTS in Noisy Environment: Listen and Speak Louder

*Sashi Novitasari*[1,2], *Sakriani Sakti*[1,2], *and Satoshi Nakamura*[1,2]

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project (AIP), Japan
{sashi.novitasari.si3, ssakti, s-nakamura}@is.naist.jp

## Abstract

Although machine speech chains were originally proposed to mimic a closed-loop human speech chain mechanism with auditory feedback, the existing machine speech chains are only utilized as a semi-supervised learning method that allows automatic speech recognition (ASR) and text-to-speech synthesis systems (TTS) to support each other given unpaired data. During inference, however, ASR and TTS are still performed separately. This paper focuses on machine speech chain inferences in a noisy environment. In human communication, speakers tend to talk more loudly in noisy environments, a phenomenon known as the Lombard effect. Simulating the Lombard effect, we implement a machine speech chain that enables TTS to speak louder in a noisy condition given auditory feedback. The auditory feedback includes speech-to-noise ratio prediction and ASR loss as a speech intelligibility measurement. To the best of our knowledge, this is the first deep learning framework that mimics human speech perception and production behaviors in a noisy environment.

**Index Terms**: text-to-speech, machine speech chain inference, Lombard effect, dynamic adaptation

## 1. Introduction

The development of text-to-speech synthesis (TTS) has enabled computers to mimic human speech production and learn how to speak. Various approaches have been conducted, and the recent technologies of end-to-end neural TTS frameworks have successfully produced natural-sounding, human-like speech [1, 2, 3, 4]. Despite remarkable performance, standard systems are commonly developed by assuming they are operating in ideal clean environments. However, in reality, many modern applications, such as digital assistants, require TTS to communicate with users in noisy places. In such cases, the TTS performance may degrade when speech intelligibility drops quite rapidly in adverse conditions. Unfortunately, although TTS can speak, it cannot listen to its own voice, and therefore, it cannot grasp the situation and overcome the problem.

Humans, on the other hand, have a closed-loop speech chain mechanism with auditory feedback from the mouth to the ear. This connection between systems of speech production and speech perception enables speakers to monitor their speech and improve it when necessary. Such a mechanism is critical not only during language acquisition but also during communication. In a noisy environment, particularly, speakers tend to speak louder to increase their speech audibility while simultaneously listening to the noise to ensure that the listeners understand what they are saying [5]. This change, which is known as the Lombard effect [6], includes not only a change of speech intensity but also changes in speech pitch and speed [7].

Inspired by the human speech chain mechanism, a machine speech chain [8, 9] (Figure 1(a)) was previously proposed to establish a closed feedback loop between the listening component (ASR) and speaking component (TTS) so both components can assist each other in semi-supervised learning given unpaired data (speech or text data only). This loop enables the machine to learn, not only by listening or speaking but also by listening while speaking. However, the existing machine speech chain was only utilized as a semi-supervised training method for ASR and TTS. During inference, since both ASR and TTS are still performed separately as in the traditional manner, they are unable to dynamically adapt based on various conditions, unlike the human speech chain.

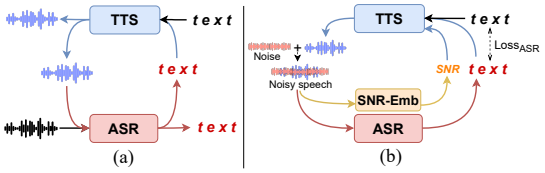


Figure 1: *(a) Previous machine speech chain that was utilized only for semi-supervised training method; (b) proposed machine speech chain utilized for both training and dynamically adaptive inference method.*

In this work, we propose an advanced version of a machine speech chain that utilizes a feedback mechanism, not only during training but also during inference. Simulating the Lombard effect, we implement a machine speech chain for an end-to-end neural TTS in noisy environments (Figure 1(b)) that enables TTS to speak louder in noisy conditions given the auditory feedback. The auditory feedback is given on utterance-level that includes speech-to-noise ratio (SNR) prediction as power measurement and ASR loss as the speech intelligibility measurement. Based on the feedback, the TTS will generate acoustic speech while adapting the speech prosody, focusing on pitch, intensity, and speed to improve the overall speech quality. This adaptation is not performed only once; it is done dynamically during communication. To the best of our knowledge, this is the first deep learning framework that mimics human speech perception and production behaviors in a noisy environment.

## 2. Related Works

The study of Lombard speech synthesis has gained attention starting from the parametric speech synthesis, which is based on the Hidden Markov Model (HMM) framework [11, 12]. The aim is to produce highly intelligible speech in the presence of noise. The Hurricane Challenge [13, 14] evaluated speech synthesis and speech enhancement systems for noisy conditions. From existing approaches, the commonly used method applies post-processing to TTS speech to modify the speech prosody [15, 16, 17].

A recent study with end-to-end neural TTS systems produced Lombard-style speech by applying transfer learning from



http://dx.doi.org/10.21437/Interspeech.2021-946

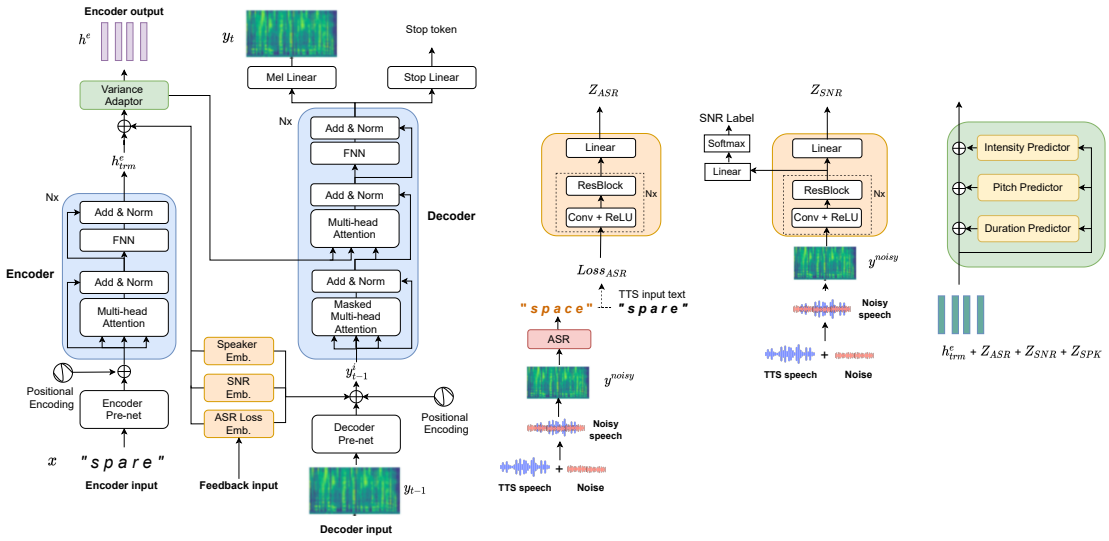


Figure 2: *Architecture: (a) proposed TTS with a Transformer-based encoder-decoder structure, extended with (b) ASR-loss embedding, (c) SNR embedding, and (d) variance adaptor [10] modules.*

a standard TTS trained on clean speech [18]. The transfer learning was done using a small amount of Lombard speech. A work by Hu et al. (2021) [19] recently constructed a multi-style Tacotron-based TTS, which speech styles include normal speech, whispered speech, and Lombard speech. The speech was synthesized by treating the desired style as speaker embedding input to the model.

As can be seen, most previous works require the information of the desired Lombard speech target output to be adapted during TTS training and cannot perform further automatic adaptation during inference. In contrast, our TTS framework dynamically adapts to the noise condition automatically during inference, given SNR and ASR-loss embedding's auditory feedback.

# 3. Proposed TTS in Speech Chain Framework

Figure 2 illustrates the overall structure of the proposed TTS. It is based on the Transformer TTS [2, 3], extended with auditory feedback components (ASR-loss embedding and SNR embedding) and a variance adaptor (Figure 2(a)). Given character sequence $x = [x_1, x_2, ..., x_S]$ with length $S$, the Transformer TTS generates the speech's Mel-spectrogram $y = [y_1, y_2, ..., y_T]$ with length $T$ and the adapted prosody based on the auditory feedback from SNR ($Z_{SNR}$) and ASR loss ($Z_{ASR}$) embedding in an end-to-end manner. We perform a dynamic adaptation in noisy situations with a feedback loop. The loop is performed in several iterations until the ASR loss converges.

We also utilize speaker recognition with DeepSpeaker [20] to enable a multi-speaker TTS, which generates speaker embedding vector $Z_{SPK} = SPKEmbedding(y)$ using a convolution network-based structure. The implementation resembles a previous machine speech chain framework [9]. We pre-trained the DeepSpeaker model, and the model weight is kept during TTS training for the constant embedding. The speaker embedding is then merged with the encoder output and the decoder input following the speaker embedding utilization method in the Multi-Speech framework [3].

From the Mel-spectrogram, we generate a magnitude spectrogram that consists of a CBHG (1-D Convolution Bank + Highway + bidirectional GRU) module that resembles the Tacotron framework [1]. We use the Griffin-Lim algorithm to estimate the phase spectrogram and the inverse short-time Fourier transform (STFT) to reconstruct the signal.

In this study, we construct three TTS systems with different feedback configurations to investigate the Lombard effect within the machine speech chain. Each system is trained using normal speech and Lombard speech of various noise conditions. The following are the details of each system.

### 3.1. TTS with SNR feedback

TTS generates speech waveform based on text input and feedback from SNR embedding. The SNR feedback represents the power or intensity measurement of how well the TTS speech can be heard in noisy environments. Given an SNR embedding feedback that represents a discretized SNR level, TTS attempts to re-synthesize speech with higher SNR ($\geq$ 20 dB).

We implement the SNR embedding module using convolution network layers (Figure 2(c)) and generate an embedding $Z_{SNR}$ from noisy speech features $y^{noisy}$:

$$Z_{SNR} = SNR\ Embedding(y^{noisy}). \quad (1)$$

We defined several SNR classes and pre-trained the SNR recognition model to generate SNR embedding vectors by learning to classify the SNR given noisy speech utterances. The learning is done with cross-entropy loss:

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

where $l$ is the reference SNR label, $p_l$ is the predicted SNR probability, and $C_l$ is the number of classes of the SNR recognition model.

SNR embedding vector $Z_{SNR}$ is then combined with TTS encoder transformer output $h^e_{trm}$ and speaker embedding $Z_{SPK}$, where final TTS encoder output $h^e$ becomes

$$h^e = h^e_{trm} + Z_{SPK} + Z_{SNR}. \quad (3)$$

Embedding vectors $Z_{SPK}$ and $Z_{SNR}$ are also combined with the TTS decoder's first transformer layer input ($y^i_{t-1}$) along with the positional encoding $PE$:

$$y^i_{t-1} = prenet(y_{t-1}) + Z_{SPK} + Z_{SNR} + PE. \quad (4)$$

Therefore in the TTS decoder's multi-head attention, the attention query, key, and value are the encoder output and decoder

input that have been embedded with the feedback information. During TTS training, the model updates are done using the standard Transformer TTS loss function.

### 3.2. TTS with SNR-ASR feedback

The second version of TTS generates speech waveform based on text input and feedback from the SNR and ASR-loss embedding. The ASR-loss embedding (Figure 2(b)) represents the speech intelligibility measurement of how well the noisy TTS speech can be recognized. The generation of ASR-loss embedding vector $Z_{ASR}$ is done by first performing noisy TTS speech recognition using an ASR and then loss calculation between text sequence hypothesis $p_x$ predicted by ASR and TTS input text $x$:

$$Z_{ASR} = \textit{ASR Loss Embedding}(Loss_{ASR}(x, p_x)), \quad (5)$$

$$p_x = p(x|y^{noisy}), \quad (6)$$

$$Loss_{ASR}(x_s, p_{x_s}) = -\sum_{c=1}^{C} \mathbb{1}(x_s = c) * \log p_{x_s}[c], \quad (7)$$

where $Loss_{ASR}(x, p_x)$ is the sentence-level loss and $Loss_{ASR}(x_s, p_{x_s})$ is the character-level loss. Here $C$ is the number of ASR output classes, $s$ is the character's index in sequence $x$. The ASR embedding module is trained directly during TTS training without a pre-training step.

Similar to the proposed TTS in Section 3.1, the ASR-loss embedding is combined into the TTS encoder output and the decoder input along with the speaker and the SNR embedding vectors:

$$h^e = h^e_{trm} + Z_{SPK} + Z_{SNR} + Z_{ASR}, \quad (8)$$

$$y^i_{t-1} = prenet(y_{t-1}) + Z_{SPK} + Z_{SNR} + Z_{ASR} + PE. \quad (9)$$

The TTS is also trained using the same loss function as the standard Transformer TTS.

### 3.3. TTS with SNR-ASR feedback and variance adaptor

Humans tend to increase their speech intensity and pitch in noisy environments and also speak slower [21]. Therefore, in addition to the SNR and ASR-loss embedding feedback, we applied a variance adaptor module within the proposed TTS with a similar approach as in FastSpeech2 [10] that guides the prosody adaptation. The variance adaptor, shown in Figure 2(d), consists of three components: a pitch predictor, an intensity predictor, and a duration predictor. Each component predicts the pitch, the intensity, and the duration of the target speech in character-level details by taking the encoder output that was combined with feedback embedding. This module is applied in the TTS encoder and provides the following output:

$$h^e = \textit{Var Adaptor}(h^e_{trm} + Z_{SPK} + Z_{SNR} + Z_{ASR}). \quad (10)$$

The decoder input follows Eq. 9. In our duration predictor, instead of predicting the token duration as an integer to extend the encoder output length like in the original FastSpeech2 framework, our duration predictor estimates the duration as a real value, similar to the other predictors. The encoder output length in our model follows the standard Transformer TTS.

With the variance adaptor, the TTS is trained with the standard TTS loss function combined with the variance predictor losses. The variance predictor loss is calculated with the mean squared error (MSE) loss function:

$$Loss_{pred}(v, \hat{v}) = \frac{1}{S}\sum_{s=1}^{S}(v_s - \hat{v}_s)^2, \quad (11)$$

where $v$ is the normalized reference value and $\hat{v}$ is the output of the predictor. The TTS training loss function becomes

$$Loss_{TTS}(\boldsymbol{y}, \hat{\boldsymbol{y}}) =$$
$$\frac{1}{T}\sum_{t=1}^{T}((y_t - \hat{y}_t)^2 - (b_t \log(\hat{b}_t) + (1 - b_t)\log(1 - \hat{b}_t))) +$$
$$Loss_{pred}(v^P, \hat{v}^P) + Loss_{pred}(v^G, \hat{v}^G) + Loss_{pred}(v^D, \hat{v}^D), \quad (12)$$

where $\boldsymbol{y} = [y, b, v^P, v^G, v^D]$ and $\hat{\boldsymbol{y}} = [\hat{y}, \hat{b}, \hat{v}^P, \hat{v}^G, \hat{v}^D]$. Here $v^P$, $v^G$, and $v^D$ are the reference pitch, the intensity, and the duration, and $\hat{v}^P$, $\hat{v}^G$, and $\hat{v}^D$ are the pitch, the intensity, and the duration predicted by the respective predictor. $b$ and $\hat{b}$ are the reference and the predicted probability of the stop tokens in the decoder.

## 4. Experiments

### 4.1. Data

We used in our experiment the Wall Street Journal (WSJ) corpus [22] whose dataset consists of multi-speaker English speeches recorded by reading news text, sampled in 16 kHz. We utilized the *SI-284*, *dev93*, and *eval92* sets as the training, development, and test sets. The *SI-284* set consists of 81 hours of speech.

To learn how human vocalization changes in noisy conditions, we also recorded natural Lombard speech with a single male speaker who read the WSJ development and test sets in noise conditions. We first simulated a noisy environment with additive white[1] and babble[2] noise of SNR 0 dB and SNR -10 dB based on WSJ clean speech data. The noise level is considered constant within an utterance. Then, given only the noise signals, the speaker read the WSJ text as if it were aimed for someone in a noisy condition. For comparison, we also recorded the clean speech version.

Next we constructed synthetic Lombard speech of a full-set of WSJ data by modifying the pitch, the intensity, and the duration of the normal WSJ speech based on the vocal realization changes observed in our natural Lombard speech[3]. Here the SNR between our synthetic Lombard speech and the noise was 20 dB.

Both the original WSJ clean speech and the synthetic Lombard WSJ speech were used for TTS training and testing. From both the clean and Lombard speech data, we extracted the character-level speech timing using the Montreal forced-alignment toolkit [24] to estimate the speech pitch, the intensity, and the duration in character-level details with which we then calculated the variance predictor losses during the TTS training.

### 4.2. Model

Our TTS model consists of a Transformer-based encoder and decoder. The TTS input was the character sequence, and the output was the 80 dimensions of the log 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 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 speaker, the ASR loss, and the SNR feedback embedding modules shared a similar configuration. Each consisted of four stacks of convolution and residual blocks and a linear layer. Before training the TTS, the SNR recognition model was pre-trained to predict the noisy speech SNR labels: SNR 0 dB, SNR

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

[2]From the noise sounds dataset in AURORA-2 corpus [23]

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

Table 1: *Speech intelligibility measure (CER %) at different SNR levels using clean- and multi-condition training ASR.*

| System | Clean condition training ASR | | | Multi-condition training ASR | | |
|---|---|---|---|---|---|---|
| | Clean | SNR 0 | SNR -10 | Clean | SNR 0 | SNR -10 |
| **Baseline TTS** | | | | | | |
| Standard TTS | 18.92 | 118.72 | 106.25 | 18.32 | 70.54 | 77.07 |
| + Rule-based modification into Lombard speech | 18.92 | 102.96 | 104.69 | 18.32 | 44.68 | 57.86 |
| + Fine tuning with Lombard speech (SNR 0) | 10.76 | 93.19 | 105.01 | 13.19 | 32.71 | 53.35 |
| + Fine tuning with Lombard speech (SNR -10) | 11.73 | 71.88 | 99.36 | 14.26 | 24.47 | 40.62 |
| + Fine tuning with Lombard speech (SNR 0 + SNR -10) | 11.25 | 79.94 | 100.44 | 13.40 | 28.12 | 46.13 |
| **Proposed TTS** | | | | | | |
| TTS in speech chain framework | 18.92 | 118.72 | 106.25 | 18.32 | 70.54 | 77.07 |
| + SNR feedback | 10.21 | 83.15 | 101.41 | 11.58 | 22.82 | 42.00 |
| + SNR-ASR feedback | 10.76 | 52.51 | 87.72 | 12.55 | 16.11 | 25.61 |
| + SNR-ASR feedback + variance adaptor | 10.47 | 55.70 | 92.75 | 11.99 | 14.70 | 24.96 |
| **Topline (human natural speech)** | | | | | | |
| Natural speech | 5.77 | 92.56 | 98.98 | 7.43 | 22.17 | 58.81 |
| + Rule-based modification into Lombard speech | 5.77 | 58.40 | 67.78 | 7.43 | 13.24 | 15.15 |
| Natural Lombard speech | 5.77 | 25.38 | 59.25 | 7.43 | 11.46 | 20.46 |

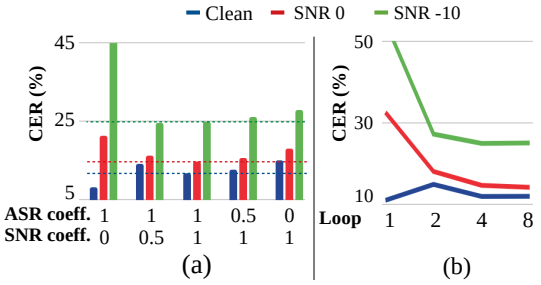


Figure 3: *The effect of auditory feedback on the TTS speech intelligibility: (a) the embedding coefficients and (b) the number of feedback loop.*

-10 dB, and clean (no noise). Our ASR model follows the same configuration as the Speech-Transformer [25].

### 4.3. Experiment Results and Discussion

In this study, we focused on evaluating TTS speech intelligibility. To get objective measurements for all the systems, we used an ASR system to recognize the TTS-generated speech and calculated the character error rate (CER). We prepared two ASR systems, which were trained in a clean condition only and a multi-condition (mixed clean and noisy speech data).

Our experiment result is shown in Table 1. The clean condition testing was done using TTS output without noise, and the noise condition testing was done by adding noise signals of the corresponding SNR condition to the generated speech. As described earlier, we investigated the Lombard effect within the machine speech chain with three different feedback configurations: (1) SNR feedback only, (2) SNR-ASR feedback, and (3) SNR-ASR feedback with variance adaptor.

In the comparison, we had several baselines: (1) the standard TTS in which the generated speech 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 with the same method as the synthetic Lombard WSJ speech construction; (3) three TTS systems that were fine-tuned to Lombard speech, following the approach of previous work by Paul et al. [18]. The topline is the natural clean and Lombard speech by a human. We also included synthetic modifications from natural human speech.

From the baseline results, we found that CER could be reduced by post-processing the speech into Lombard speech. Still, the fine-tuned baseline systems resulted in a better performance than the post-processing system. However, our experiment results show that our proposed TTS outperformed the baselines. By incorporating SNR and ASR feedback together, the proposed models significantly outperformed the fine-tuned baseline models and more closely approached the CER of topline human speech intelligibility measure. The SNR feedback guided the TTS to synthesize a louder speech in noise, while ASR feedback improved the speech intelligibility further and resulted in lower CER than only modifying TTS speech or fine-tuning the TTS with Lombard speech. Here the best performance was achieved by TTS with a variance adaptor.

Using the best model with variance adaptor, we further analyzed how each auditory feedback, SNR and ASR-loss embedding, affected the TTS performance as shown in Figure 3(a). We experimented with various coefficient values for both SNR and ASR-loss embedding when they were combined into TTS encoder output $h^e$ and decoder's first transformer layer input $y^i_{t-1}$. The results show that during the clean condition, the best performance is without SNR feedback, but once the environment becomes noisy, the SNR feedback becomes critical. However, using only SNR feedback to improve the intensity, pitch, and duration may not be enough. The optimum performance is when both coefficients equal one, indicating that ASR-loss embedding feedback is also one crucial factor to improve speech intelligibility in producing Lombard speech. On the other hand, Figure 3(b) shows the number of speech chain loops required during dynamic adaptation. Humans usually attempt to produce Lombard speech in several trials when trying to be heard over the noise. The results here reveal that the machine can also dynamically adapt in several loops; listen to its voice in a noisy environment and then speak louder to improve it. For further information on speech samples, see the following reference: https://sites.google.com/view/lombard-dynamic-tts/home.

## 5. Conclusions

We constructed a dynamically adaptive machine speech chain inference framework to support TTS in noisy conditions. 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 closer to human performances. These results reveal that dynamic adaptation with auditory feedback is critical not only for human speech production mechanisms but also in speech generation by machines.

## 6. Acknowledgements

Part of this work is supported by JSPS KAKENHI Grant Numbers JP17H06101 and JP21H03467.

# 7. References


[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Proc. INTERSPEECH*, 2017, pp. 4006–4010.

[2] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with Transformer network," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6706–6713.

[3] M. Chen, X. Tan, Y. Ren, J. Xu, H. Sun, S. Zhao, and T. Qin, "MultiSpeech: Multi-speaker text to speech with Transformer," in *Proc. INTERSPEECH*, 2020, pp. 4024–4028.

[4] J. Donahue, S. Dieleman, M. Bińkowski, E. Elsen, and K. Simonyan, "End-to-end adversarial text-to-speech," in *Proc. ICLR*, 2021.

[5] 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.

[6] H. Lane and B. Tranel, "The Lombard sign and the role of hearing in speech," *Journal of Speech and Hearing Research*, vol. 14, no. 4, pp. 677–709, 1971.

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

[8] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proc. IEEE ASRU*, 2017, pp. 301–308.

[9] ——, "Machine speech chain," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 976–989, 2020.

[10] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," *ArXiv*, vol. abs/2006.04558, 2020.

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

[12] C. Valentini-Botinhao, J. Yamagishi, S. King, and R. Maia, "Intelligibility enhancement of HMM-generated speech in additive noise by modifying Mel cepstral coefficients to increase the glimpse proportion," *Computer Speech and Language*, vol. 28, pp. 665–686, 2014.

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

[14] 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.

[15] 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 on Speech Synthesis*, 2010.

[16] H. Schepker, J. Rennies, and S. Doclo, "Speech-in-noise enhancement using amplification and dynamic range compression controlled by the speech intelligibility index," *The Journal of the Acoustical Society of America*, vol. 138, no. 5, 2015.

[17] F. Bederna, H. Schepker, C. Rollwage, S. Doclo, A. Pusch, J. Bitzer, and J. Rennies, "Adaptive compressive onset-enhancement for improved speech intelligibility in noise and reverberation," in *Proc. INTERSPEECH*, 2020, pp. 1351–1355.

[18] 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.

[19] Q. Hu, T. Bleisch, P. Petkov, T. Raitio, E. Marchi, and V. Lakshminarasimhan, "Whispered and Lombard neural speech synthesis," in *Proc. IEEE SLT*, 2021.

[20] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep Speaker: An end-to-end neural speaker embedding system," *CoRR*, vol. abs/1705.02304, 2017.

[21] L. Stowe and E. Golob, "Evidence that the Lombard effect is frequency-specific in humans," *The Journal of the Acoustical Society of America*, vol. 134, no. 1, pp. 640–647, 2013.

[22] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[23] D. Pearce and H. G. Hirsch, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy condition," in *Proc. ICSLP*, 2000, pp. 29–32.

[24] 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.

[25] L. Dong, S. Xu, and B. Xu, "Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. ICASSP*, 2018, pp. 5884–5888.