Improving Intelligibility of Synthesized Speech in Noisy Condition with Dynamically Adaptive Machine Speech Chain

Sashi Novitasari[1,2], Sakriani Sakti [1,3,2], and Satoshi Nakamura [1,2]

[1]Nara Institute of Science and Technology, Japan

[2]RIKEN, Center for Advanced Intelligence Project (AIP), Japan

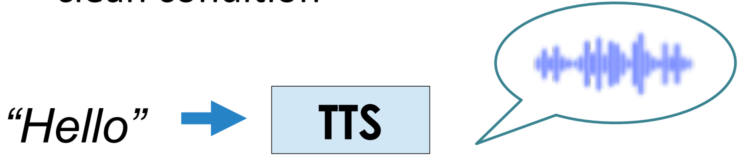
[3]Japan Advanced Institute of Science and Technology, Nomi-shi, 923– 1292 Japan

E-mail: {sashi.novitasari.si3, ssakti, s-nakamura}@is.naist.jp

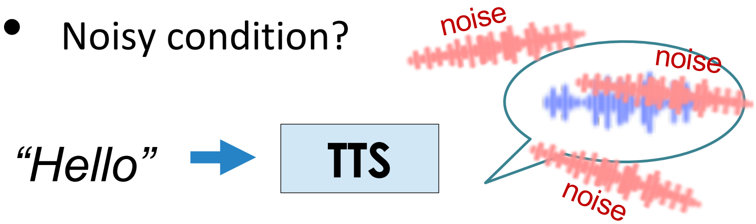
SPEECH PRODUCTION

State-of-the-art: End-to-end neural TTS

- Synthesizes a human-like speech in clean condition



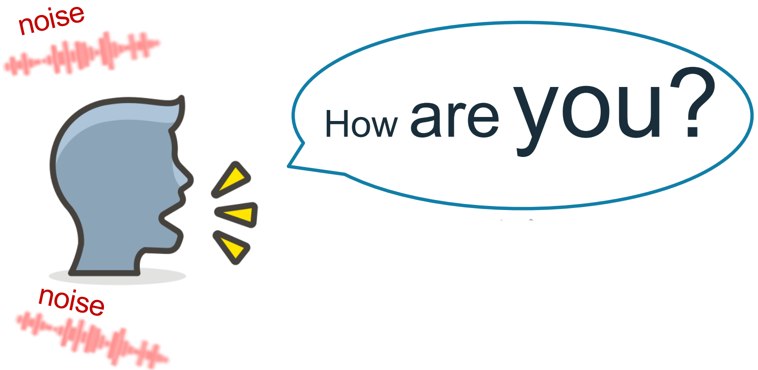
- Noisy condition?



Cannot perform well!

How about humans?

In noisy situation, we tend to speak louder (**Lombard effect**)

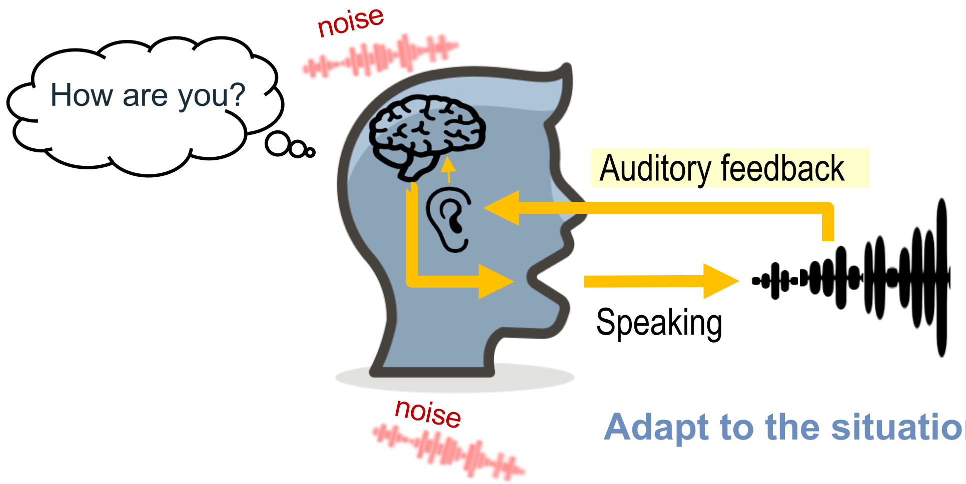


- Existing work with neural TTS:
  Fine-tuning to certain noise [Paul et al., 2020]
- Human:
  No fine-tuning before speaking in noisy place
  → **How?**

SPEECH PRODUCTION

Human

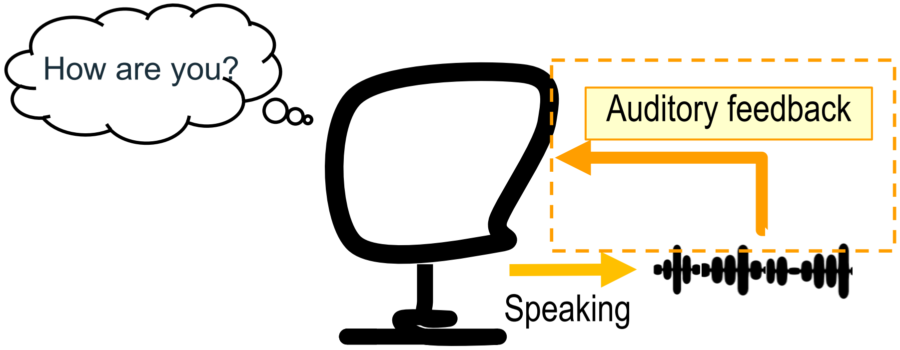
- Humans speak while listen to their own speech
  Speech chain[Denes, 1993]



Adapt to the situation

TTS

- Computers only learn how to speak
- Cannot hear their own voice



Only speaking

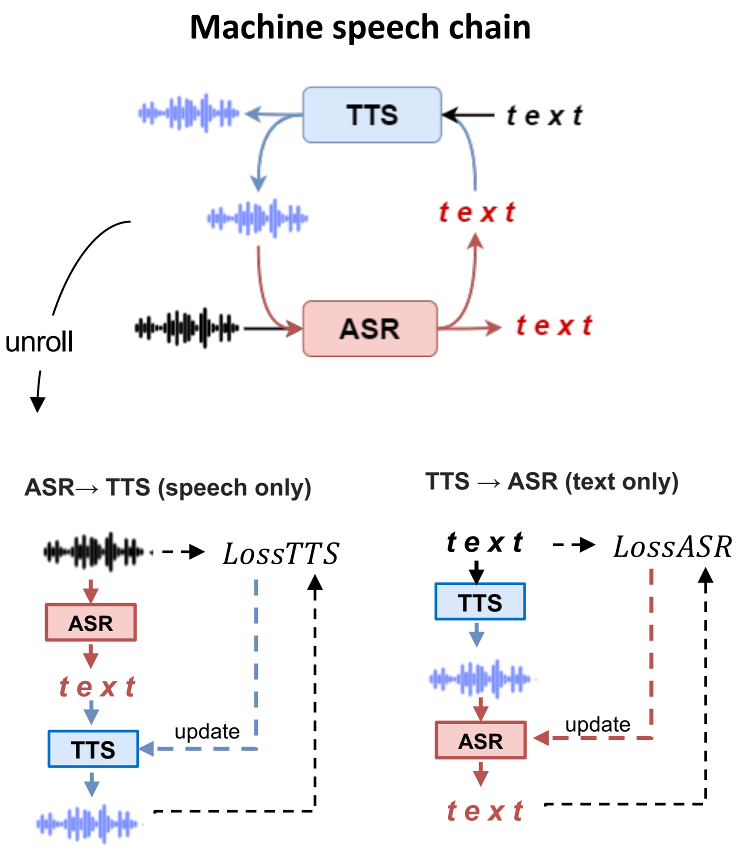
MACHINE SPEECH CHAIN

- Introduced in 2017 [Tjandra et al., 2017]
- ASR and TTS are connected via closed feedback loop during training

  → Support each other and improve together

Limitation: Only for training mechanism

- In inference, ASR and TTS perform separately as in the standard manner
- Unable to dynamically adapt based on various conditions (unlike humans)

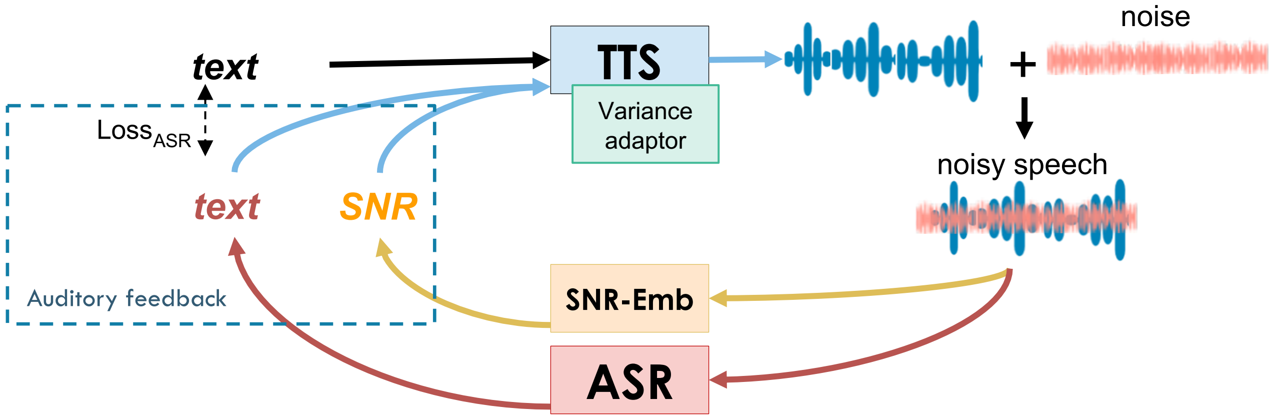


PROPOSED METHOD

New Generation of Machine Speech Chain

Dynamically Adaptive Machine Speech Chain Inference for TTS

TTS speaks louder in noisy environment by taking auditory feedback



RELATED WORKS
TTS IN NOISY CONDITION

Parametric TTS in noise

- HMM TTS speech modification to increase speech intelligibility in noise while keeping the speech energy fixed [Valentini-Botinhao et al., 2014; Schepker et al., 2015]
- HMM TTS adapted to Lombard speech data [Raitio et al., 2014]

Neural network-based TTS in noise

- Transfer learning from a standard end-to-end TTS (clean) to an end-to-end Lombard TTS [Paul et al., 2020]
    - Lombard TTS is trained on a small Lombard dataset
- End-to-end multi-style TTS [Hu et al., 2021]
    - Synthesizable speech styles: Normal speech, whispered speech, Lombard speech

Offline fine-tuning

Our focus

End-to-end Lombard TTS with dynamic adaptation using auditory feedback, similar to human

PROPOSED METHOD

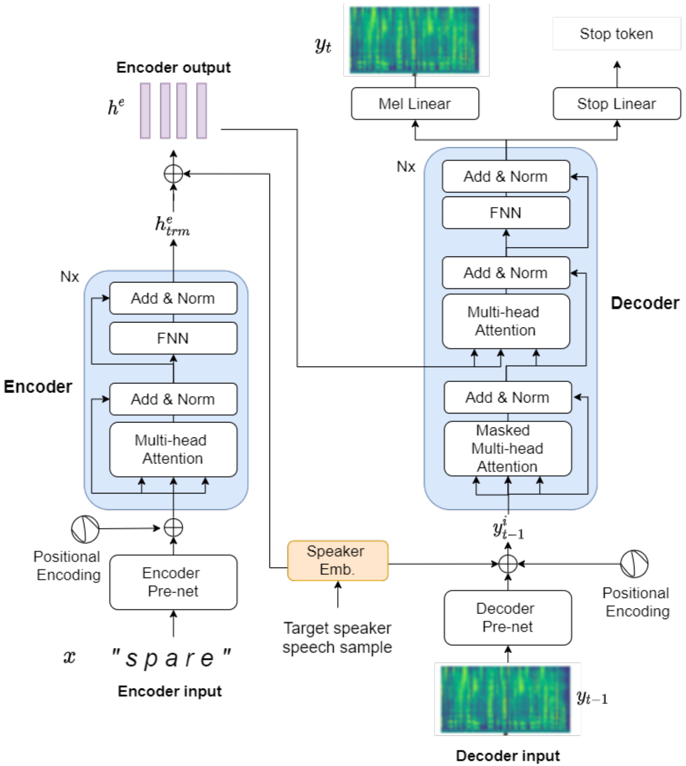
PROPOSED TTS

TRANSFORMER TTS WITH AUDITORY FEEDBACK

- Basic TTS structure: Transformer TTS [Li et el., 2018]
  - Input : Characters
  - Output : Speech features (80 dims. Mel-spectrogram)

  - Multi-speaker experiment: Multi-speaker TTS Transformer [Chen et al., 2020]
  - Speaker embedding: Deep Speaker [Li et al., 2017] (similar to TTS in the basic machine speech chain)

- **Proposed TTS structure**:
  a) TTS + SNR embedding
  b) TTS + ASR-SNR embedding
  c) TTS + ASR-SNR embedding + Variance adaptor



Multi-speaker Transformer TTS with Deep Speaker speaker embedding

A. TTS with SNR embedding

Auditory feedback

- **SNR embedding**(Z_{SNR}): SNR of noisy speech (y^{noisy})

$$Z_{SNR} = SNR\ Emb\ (y^{noisy})$$

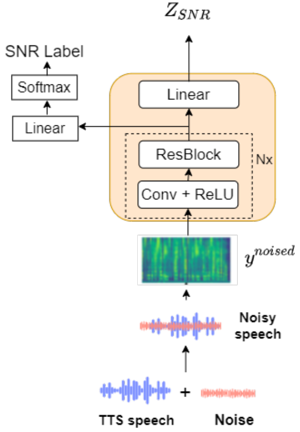
  - Trained as SNR recognition model first
  - Utilized in:
    - Encoder output (h^e)

$$h^e = h^e_{trm} + Z_{SPK} + Z_{SNR}$$

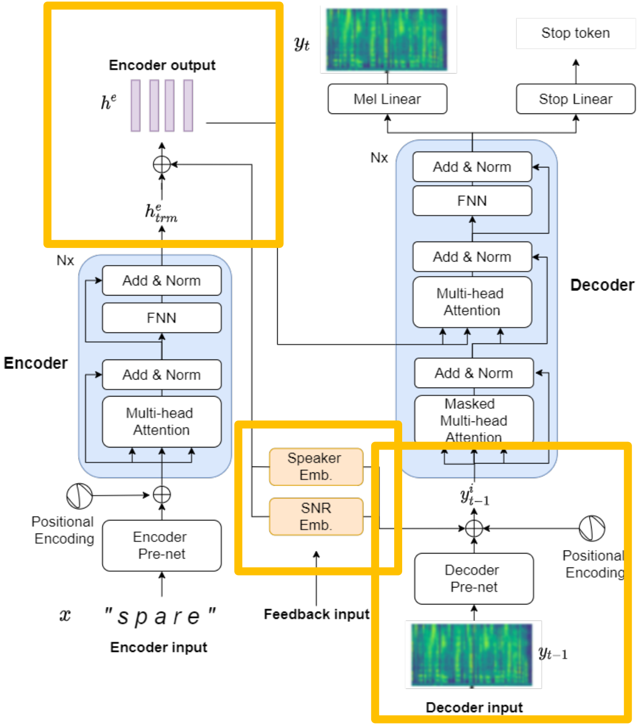
    - Decoder first layer Input (y^i_{t-1})

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

Z_{SPK} : speaker embedding
PE : positional encoding



SNR emb. module



Transformer TTS with SNR emb.

B. TTS with SNR and ASR-loss embedding

Auditory feedback:

- **SNR embedding**
- **ASR-loss embedding** (Z_{ASR}): Maps the ASR MSE loss into embedding space

$$Z_{ASR} = ASR\ Loss\ Emb\ \left(Loss_{ASR}(x, p_x)\right)$$
$$p_x = p(x|y^{noisy})$$

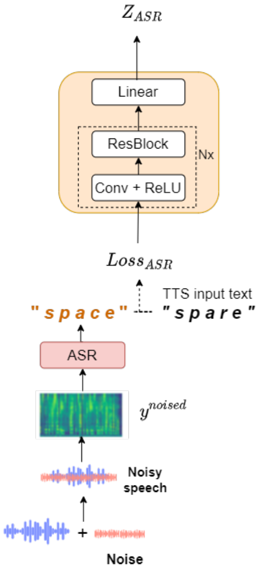
x : TTS input text (correct text)

p_x : ASR hypothesis

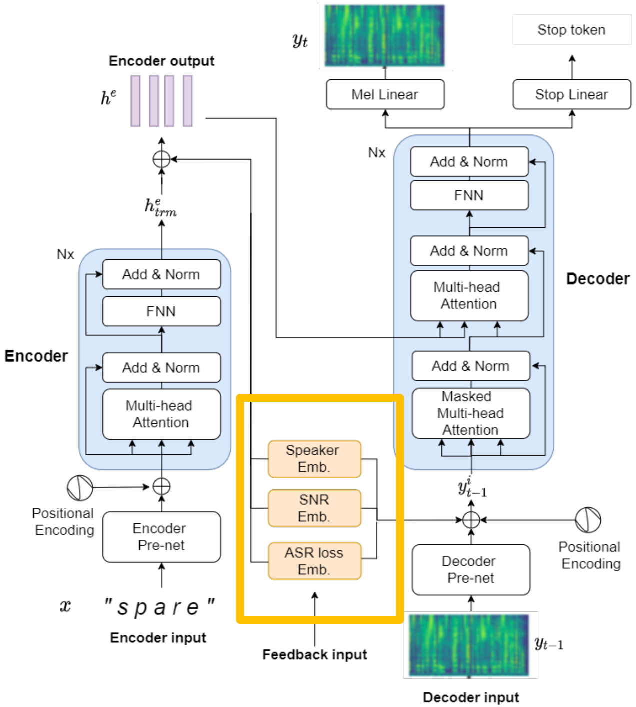
- Utilized in encoder output and decoder input:

$$h^e = h^e_{trm} + Z_{SPK} + Z_{SNR} + Z_{ASR}$$

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



ASR-loss emb. module



Transformer TTS with SNR and ASR-loss emb.

C. TTS with SNR, ASR-loss embedding, and variance adaptor

Auditory feedback:

- SNR embedding
- ASR-loss embedding

Prosody guide: Variance adaptor

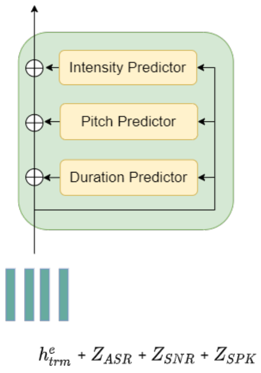
- Based on variance adaptor in Fast Speech [Ren et al., 2020], modified for autoregressive Transformer decoder
- 3 components → predict character-level speech prosody:

$$v^X = Predictor^X(h^e_{trm} + Z_{SPK} + Z_{SNR} + Z_{ASR})$$

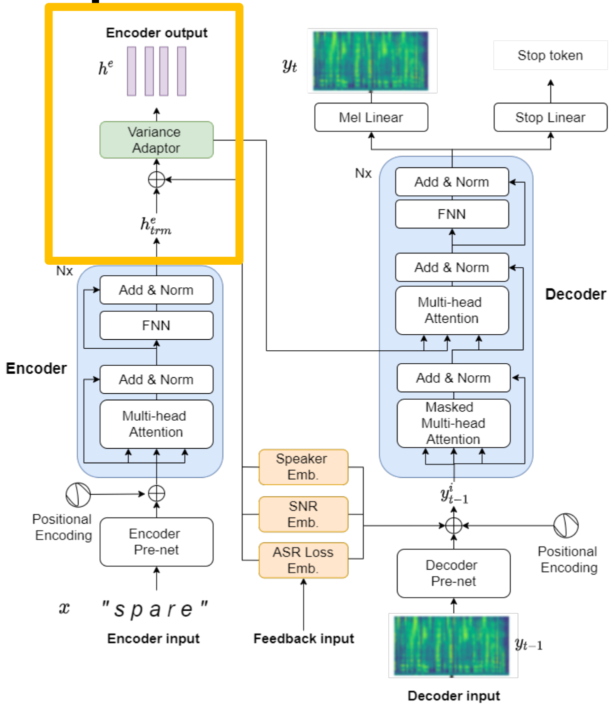
  - Intensity predictor ($X = G$)
  - Pitch predictor ($X = P$)
  - Duration predictor ($X = D$)

- Add the speech prosodies information to encoder output :

$$h^e = v^G + v^P + v^D + (h^e_{trm} + Z_{SPK} + Z_{SNR} + Z_{ASR})$$



Variance adaptor



Transformer TTS with SNR, ASR-loss embedding, and variance adaptor

Experiments

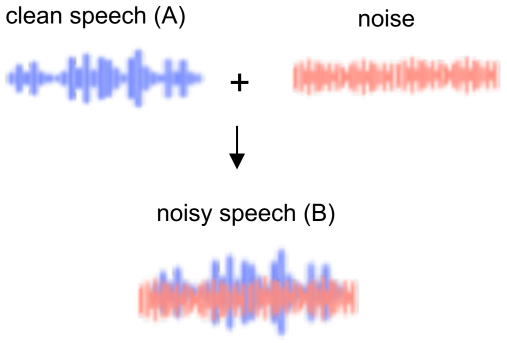
EXPERIMENT SETTING
DATA

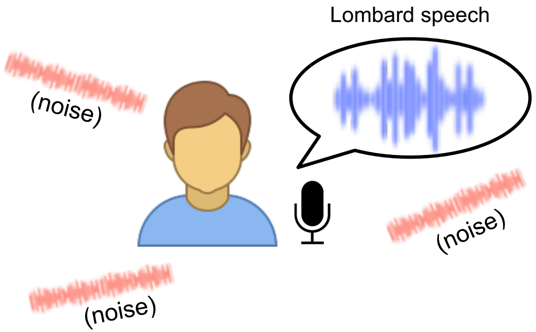
A. Clean Wall Street Journal (WSJ) speech [Paul et al., 1992]
- Multi-speaker English speech, 81 hours of speech
- Training: *SI-284* set, dev: *dev92* set, test: *eval93* set

B. WSJ speech with additive noise
- Clean WSJ speech combined with noisy sound
  - Noise type : white noise and babble noise
  - SNR : SNR 0 and SNR -10

C. Natural Lombard speech
- Clean and noisy speech recorded from single male speaker
- Text: WSJ speech transcription (*dev92 + eval93*)

D. Synthetic Lombard WSJ speech
- Clean WSJ speech with the intensity, pitch, and duration modified into Lombard speech





EXPERIMENT SETTING

SYSTEM CONFIGURATION

Topline: Natural Lombard speech

Models structure and training data configuration

System	Structure	Training Data
TTS		
Baseline standard TTS	Transformer- 6 Enc, 6 Dec	Clean WSJ
Baseline standard TTS + Fine-tuning [Paul et al., 2020]		Clean WSJ + Synthetic Lombard WSJ
Proposed TTS		Clean WSJ + Synthetic Lombard WSJ
Feedback component		
ASR	Transformer- 12 Enc, 6 Dec (Speech-transformer [Dong et al., 2018])	Clean WSJ + Noisy WSJ
SNR recognition	4 convolutional + residual layers	Clean WSJ + Noisy WSJ (class: clean, SNR 0, SNR -10)

RESULT

- Evaluation → Speech intelligibility metric:
  - ASR Character error rate (CER)
  - ASR recognize noisy TTS speech
- Proposed TTS max. feedback loop: 4

- Best performance by TTS + SNR-ASR loss emb. + variance adaptor
  - SNR and ASR feedback improved the speech intelligibility
  - Variance adaptor guided the prosody change well by providing the target prosody information

How the auditory feedback affected the TTS performance?

Speech intelligibility measure (CER %) at different SNR levels using ASR trained on clean and noisy conditions.

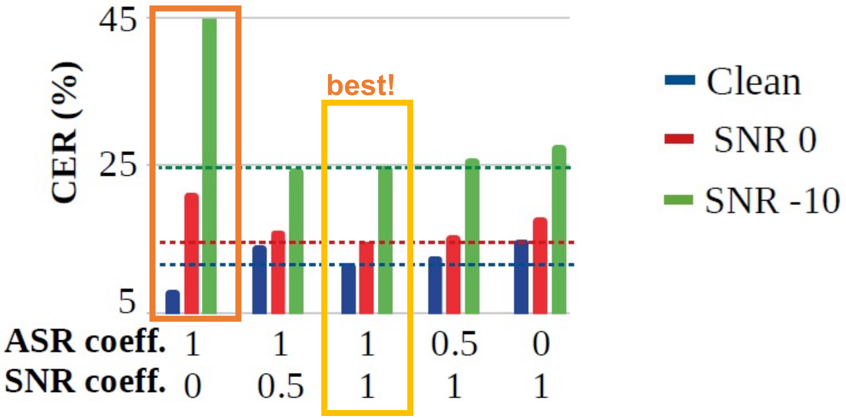
System	Clean	SNR 0	SNR -10
Baseline TTS			
Standard TTS	18.32	70.54	77.07
+ modification into Lombard speech	18.32	44.68	57.86
+ Fine-tuning with Lombard speech	13.40	28.12	46.13
Proposed TTS			
TTS + SNR emb.	**11.58**	22.82	42.00
TTS + SNR-ASR loss emb.	12.55	16.11	25.61
TTS + SNR-ASR loss emb. + var. adaptor	11.99	**14.70**	**24.96**
Topline (human natural speech)			
Natural speech	7.43	22.17	58.81
+ modification into Lombard speech	7.43	13.24	15.15
Natural Lombard speech	7.43	11.46	20.56

Result

How the auditory feedback affects TTS speech?

- Experiments by applying a coefficient to SNR embedding and ASR-loss embedding in encoder output and decoder input (default coefficient: 1)

The effect of auditory feedback on speech intelligibility

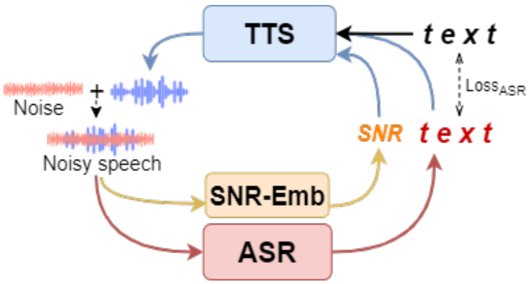


- Clean condition: best performance with ASR feedback only (ASR coeff 1, SNR coeff 0)
- Noisy condition: best performance by equal amount of ASR + SNR feedback (coeff 1)

Both SNR and ASR-loss information are important to synthesize Lombard speech

Result

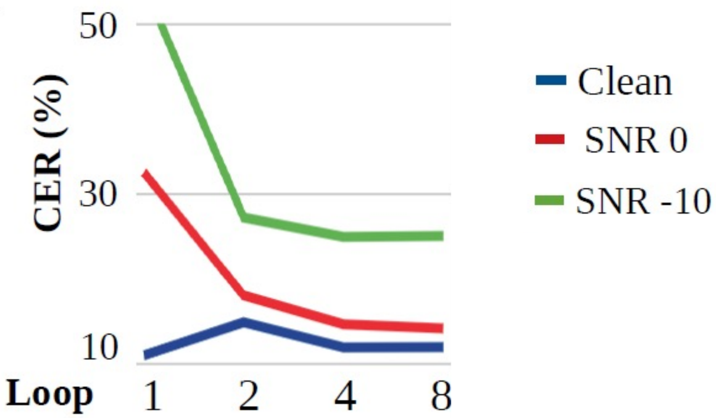
How the feedback loop affects TTS speech?



- Loop 1 : No feedback utilization
- Improvement significantly occurs after the 2nd loop

TTS performed dynamic adapt in several loops; listen to its voice in a noisy environment and then speak louder (similar to humans)

The effect of feedback loop on speech intelligibility



CONCLUSION

- Dynamically adaptive machine speech chain inference framework to support TTS in noisy conditions.
- The proposed systems with auditory feedback and a variance adaptor produced a highly intelligible speech that surpassed a standard TTS with a fine-tuning method and achieved closer to the human performances.
- Dynamic adaptation with auditory feedback is critical not only for human but also in speech generation by machines

THANK YOU

Appendix

RESULT

- Evaluation → Speech intelligibility metric:
  - ASR Character error rate (CER)
  - ASR recognize noisy TTS speech
- Proposed TTS max. feedback loop: 4

- Best performance by TTS + SNR-ASR loss emb. + variance adaptor
  - SNR and ASR feedback improved the speech intelligibility
  - Variance adaptor guided the prosody change well by providing the target prosody information

How the auditory feedback affected the TTS performance?

Speech intelligibility measure (CER %) at different SNR levels using ASR trained on clean and noisy conditions.

System	Clean	SNR 0	SNR -10
Baseline TTS			
Standard TTS	18.32	70.54	77.07
+ modification into Lombard speech	18.32	44.68	57.86
+ Fine-tuning with Lombard speech	13.40	28.12	46.13
Proposed TTS			
TTS + SNR emb.	**11.58**	22.82	42.00
TTS + SNR-ASR loss emb.	12.55	16.11	25.61
TTS + SNR-ASR loss emb. + var. adaptor	11.99	**14.70**	**24.96**
Topline (human natural speech)			
Natural speech	7.43	22.17	58.81
+ modification into Lombard speech	7.43	13.24	15.15
Natural Lombard speech	7.43	11.46	20.56

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
+ modification into Lombard speech (rule)	18.92	102.96	104.69	18.32	44.68	57.86
+ Fine-tuning with Lombard speech (SNR0)	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 (SNR0 + SNR-10)	11.25	79.94	100.44	13.40	28.12	46.13
Proposed TTS						
TTS + SNR emb	10.21	83.15	101.41	11.58	22.82	42.00
TTS + SNR-ASR loss emb.	10.76	52.51	87.72	12.55	16.11	25.61
TTS + SNR-ASR loss emb. + variance adaptor	10.47	55.70	92.75	11.99	14.70	24.96
Topline (human natural speech)						
Normal speech	5.77	92.56	98.98	7.43	22.17	58.81
+ modification into Lombard speech (rule)	5.77	58.40	67.78	7.43	13.24	15.15
Lombard speech	5.77	25.38	59.25	7.43	11.46	20.56

TTS with SNR, ASR-loss embedding, and variance adaptor

Variance adaptor

- Predictor training loss

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

$\hat{v}$ = predicted prosody
v = prosody label
S = character seq. length

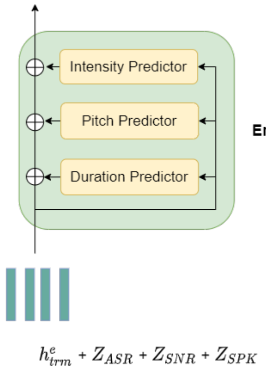
- Label: character-level prosody
  - Char-speech alignment: Force-alignment
  - Prosody label: extracted using FastSpeech open-source code
- TTS training loss

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

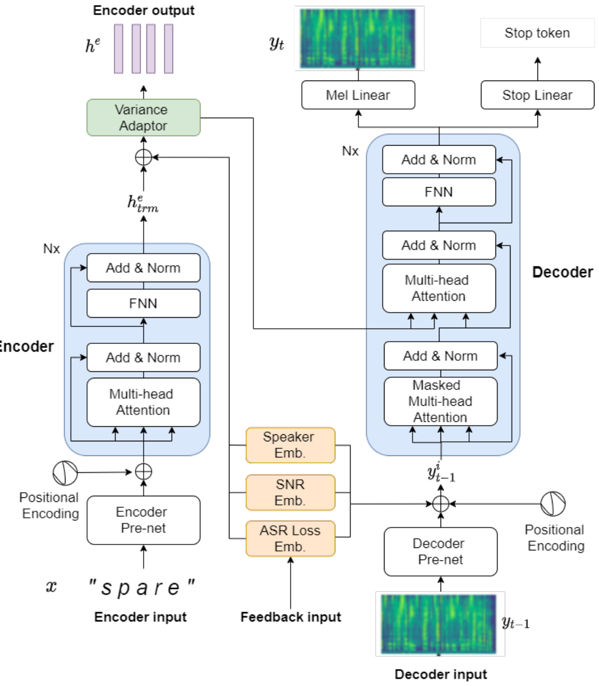
T = speech length
$\hat{y}$= pred. speech
y= ref. speech
$\hat{b}$= pred. stop token
b = stop token label

$\hat{v}^P$ = pred. pitch
v^P = ref. pitch
$\hat{v}^G$ = pred. intensity
v^G= ref. intensity

$\hat{v}^D$ = pred. duration
v^D= ref. duration



Variance adaptor



Transformer TTS with SNR, ASR-loss embedding, and variance adaptor

DATA PREPARATION (3)

D. Synthetic Lombard WSJ speech

- Clean WSJ speech with the modified prosody
  - *Intensity* increased to reach SNR 20
  - *Pitch/duration* were increased using a coefficient based on speech phoneme-level pitch/duration changes in natural Lombard speech (*dev92*) to keep speaker characteristic

Speech examples (noise: from SNR -10)

A. Clean WSJ	B. Clean WSJ + noise	C. Natural Lombard speech	D. Synthetic Lombard WSJ	
			clean	noisy