Style Equalization: Unsupervised Learning of Controllable Generative Sequence Models

- ICML 2022
- Apple
- Paper, Demo

Overview

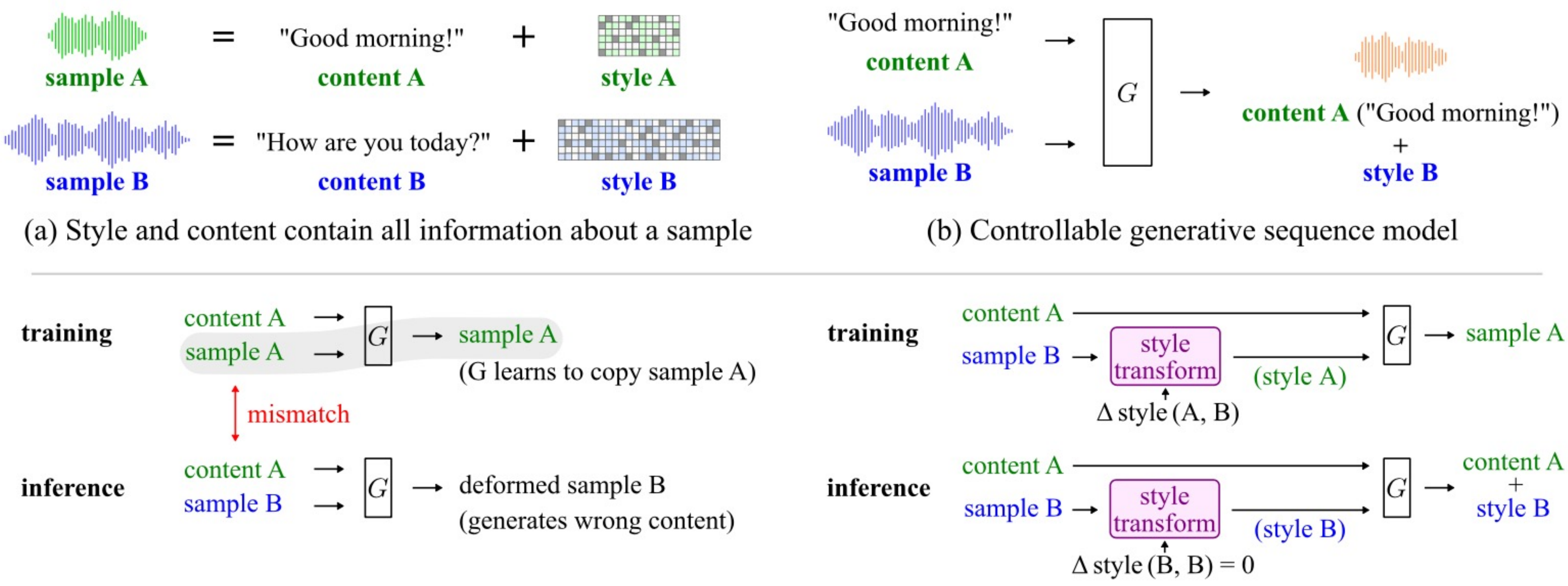
- Motivation
  - Content-leakage problem due to training-inference mismatch
  - ⇒ Unparallel setting training method for style transfer
  - Challenge: no GT output

- Input / output Design
  - Text A + style B => style A with Text A (GT audio)
  - Text와 style을 다른 source에서 입력받도록 디자인
    - Text A + Style A => style B with Text A -> no GT audio

- Variational RNN을 활용한 Controllable Generative Sequence Model
- Style Equalization
  - Style B와 Style A의 차이를 먼저 구하고, style transform을 시행

- TTS와 handwritining의 style transfer에 적용

Motivation



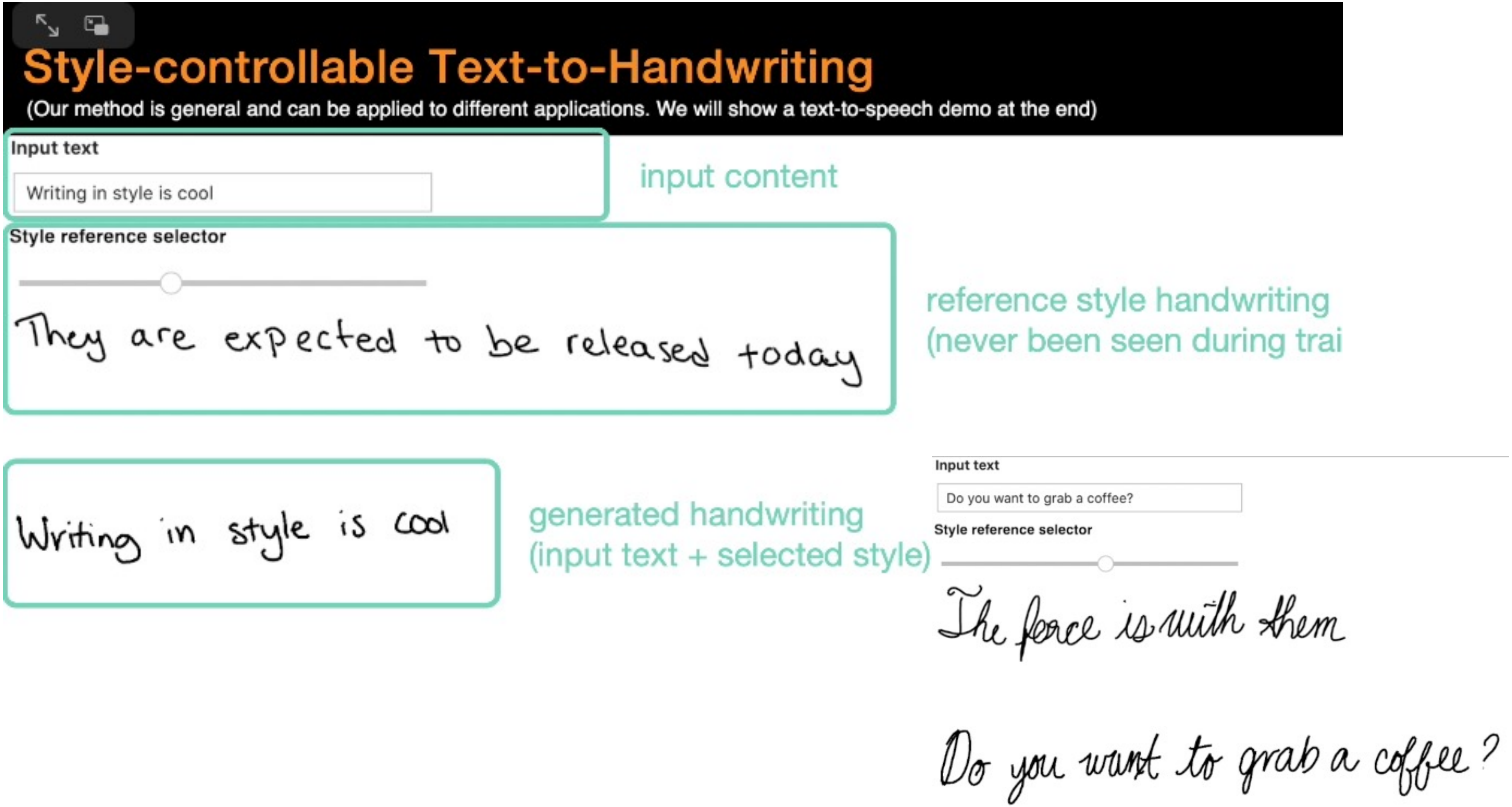
(a) Style and content contain all information about a sample

(b) Controllable generative sequence model

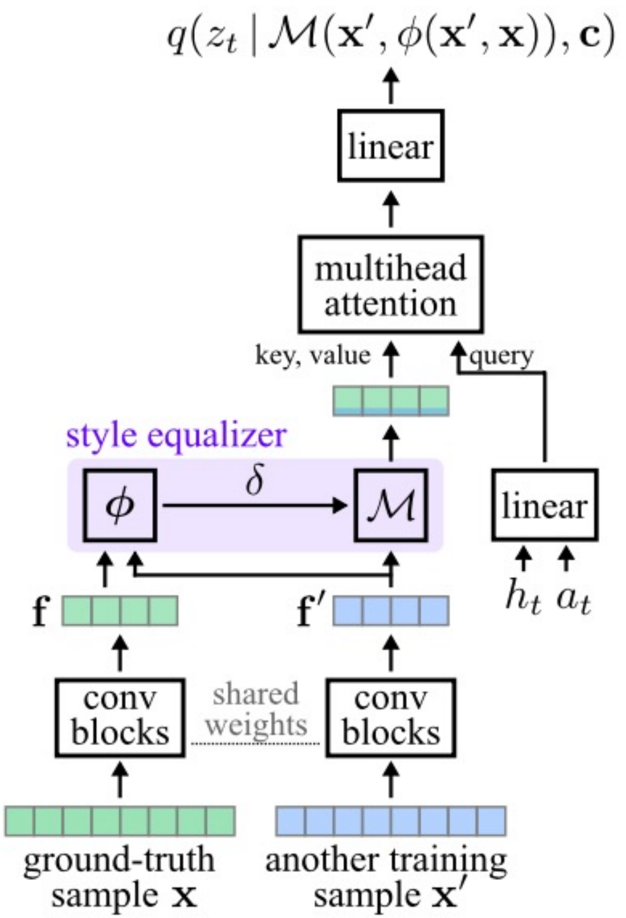
(c) Training-inference mismatch leads to generation errors

(d) Proposed style equalization

Handwriting Style Transfer Example



Style Equalization



(c) proposed style encoder with style equalization

- **Training Scheme**

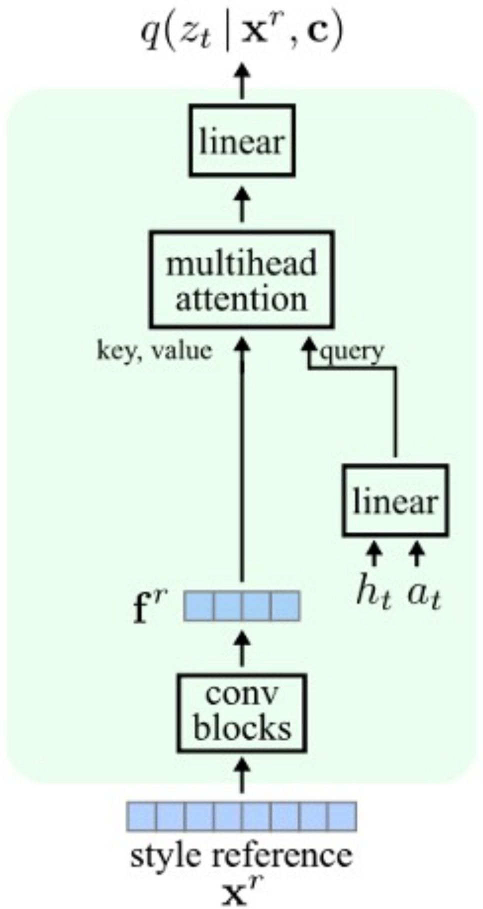
- **$\mathbb{x}'$: style input / $\mathbb{x}$: target style**
- **먼저, 각 style을 encoding -> style feature f′, f**
- **$\boldsymbol{\phi}$: f′, f 간의 차이를 구하는 function**
  - $\boldsymbol{\phi}(\mathbb{x}', \mathbb{x}) = \boldsymbol{avg}(\boldsymbol{Af}) - \boldsymbol{avf}(\boldsymbol{Af}')$
  - **$\boldsymbol{A}$: linear layer, $\boldsymbol{avg}$: average pooling**
- **$\mathcal{M}$: Learnable style transformation function**
  - **$\mathcal{M}(\mathbb{x}', \boldsymbol{\phi}(\mathbb{x}', \mathbb{x}))$: style difference와 input style 을 받아서, target style x로 변환**
  - $\mathcal{M}(\mathbb{x}', \boldsymbol{\phi}(\mathbb{x}', \mathbb{x})) = \mathbf{f}' + \mathbf{A}^{\mathbf{T}}\boldsymbol{\phi}(\mathbb{x}', \mathbb{x})$
    - **Style difference를 every time step에 더해줌**
- **Q : from content / K, V : style**
- **Content, target style, style diffence로부터 posterior distribution을 approximate**

Style Equalization

- **inference Scheme**

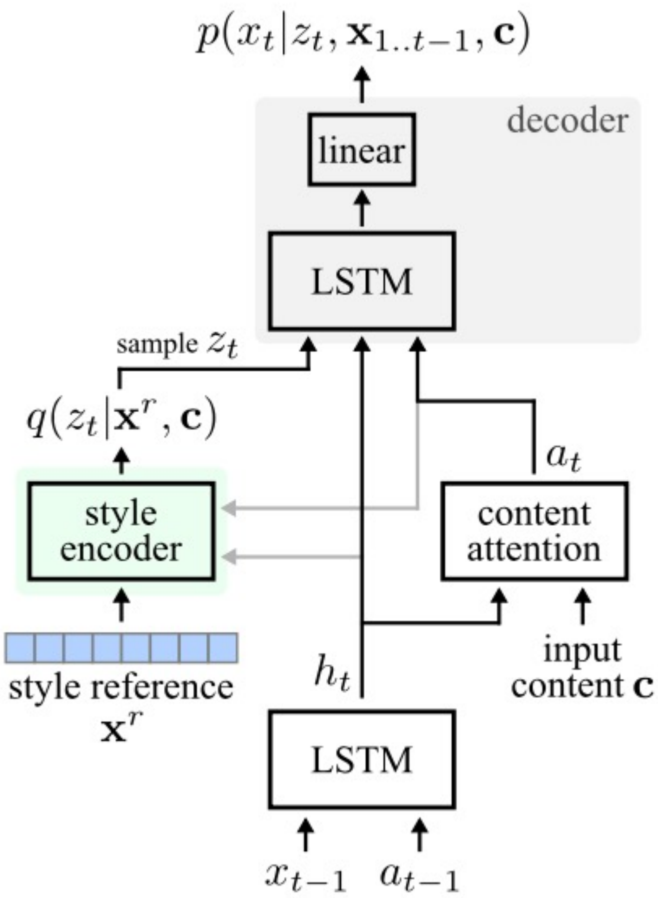
- **cf) Training**
  - **$\mathbb{x}'$: style input / $\mathbb{x}$: target style**
  - **$\boldsymbol{\phi}$: f′, f 간의 차이를 구하는 function**
    - $\boldsymbol{\phi}(\mathbb{x}', \mathbb{x}) = \boldsymbol{avg}(\boldsymbol{Af}) - \boldsymbol{avf}(\boldsymbol{Af}')$
    - **$\boldsymbol{A}$: linear layer, $\boldsymbol{avg}$: average pooling**
  - **$\mathcal{M}$: Learnable style transformation function**
    - $\mathcal{M}\left(\boldsymbol{x}', \boldsymbol{\phi}(\mathbb{x}', \mathbb{x})\right) = \mathbf{f}' + \mathbf{A}^{\mathbf{T}}\boldsymbol{\phi}(\mathbb{x}', \mathbb{x})$
      – **Style difference를 every time step에 더해줌**

- ***Input Style = Target Style***
- $\boldsymbol{\phi}(\mathbb{x}', \mathbb{x}) = \mathbf{0}$ => $\mathcal{M}\left(\boldsymbol{x}', \boldsymbol{\phi}(\mathbb{x}', \mathbb{x})\right) = \mathbf{f}'$



(b) proposed style encoder

Model Structure



(a) overview of the entire model

- **Variational RNN**
  - **VAE with Recurrence**

- $\boldsymbol{c}$ **: content embedding (output of phoneme encoder)**
- $\mathbb{x} = [\boldsymbol{x1}, \boldsymbol{x2}, \ldots \boldsymbol{xT}]$ **: GT audio**
- $\mathbb{z} = [\boldsymbol{z1}, \boldsymbol{z2}, \ldots, \boldsymbol{zT}]$ **: Reference style information**

Experiment

- **Training dataset : VCTK, LibriTTS**

- **Baseline**
  - **Tacotron**
  - **Tacotron-S**
  - **GST-16 / GST-64**
  - **GST-16S / GST-64S**
  - **GST/Tactron-S : reference encoder 훈련 데이터에 Voxceleb dataset도 추가**
  - **Token 개수 : capacity of reference encoder**

- **Metrics**
  - **SMOS**
  - **WER, cos-sim, sRank**
    - **sRank : 평가셋에 있는 모든 speaker와 cos-sim을 계산한 뒤, Target 음성의 rank를 구함**

Result - Quantitative

Table 1: Quantitative results on VCTK dataset. The reference style inputs are seen (randomly selected from the training set). WER measures content accuracy; cosine-similarity (cos-sim) and sRank measure style similarity.

Method	Parallel text			Nonparallel text		
	WER (%)	cos-sim ↑	sRank ↓	WER (%)	cos-sim ↑	sRank ↓
Tacotron	16.0 ± 1.7	0.05 ± 0.13	53.1 ± 29.1	16.4 ± 1.2	0.05 ± 0.12	53.9 ± 27.8
Tacotron-S	13.6 ± 0.7	0.24 ± 0.18	16.4 ± 20.9	16.3 ± 0.4	0.22 ± 0.18	18.0 ± 21.9
GST-16	18.6 ± 0.9	0.23 ± 0.15	21.4 ± 21.9	18.5 ± 1.1	0.23 ± 0.16	21.1 ± 22.4
GST-64	16.9 ± 0.5	0.23 ± 0.17	24.4 ± 23.1	27.5 ± 0.4	0.22 ± 0.16	25.2 ± 24.4
GST-16S	8.3 ± 0.1	0.34 ± 0.18	10.8 ± 15.2	17.7 ± 0.8	0.31 ± 0.17	13.0 ± 20.0
GST-64S	14.1 ± 0.3	0.33 ± 0.18	11.4 ± 16.3	24.7 ± 1.0	0.32 ± 0.18	12.7 ± 18.1
Proposed	$\mathbf{7.4 \pm 0.2}$	$\mathbf{0.73 \pm 0.12}$	$\mathbf{1.5 \pm 2.1}$	$\mathbf{9.5 \pm 0.4}$	$\mathbf{0.64 \pm 0.14}$	$\mathbf{1.9 \pm 4.2}$
Oracle	6.6 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	6.6 ± 0.0	0.57 ± 0.16	1.6 ± 4.1

Table 2: Quantitative results on LibriTTS-all-960 dataset.

Method	Seen speakers, parallel text			Seen speakers, nonparallel text			Unseen speakers, nonparallel text		
	WER (%)	cos-sim ↑	sRank ↓	WER (%)	cos-sim ↑	sRank ↓	WER (%)	cos-sim ↑	sRank ↓
Tacotron	64.4 ± 4.1	0.00 ± 0.10	1218 ± 671	52.2 ± 1.7	0.01 ± 0.10	1140 ± 651	52.2 ± 0.6	0.00 ± 0.10	847 ± 563
Tacotron-S	14.9 ± 0.0	0.23 ± 0.22	430 ± 584	18.7 ± 0.2	0.24 ± 0.22	370 ± 562	13.5 ± 0.0	0.16 ± 0.16	289 ± 436
GST-64	38.2 ± 2.2	0.12 ± 0.19	706 ± 701	33.3 ± 3.4	0.12 ± 0.19	700 ± 739	30.4 ± 2.2	0.09 ± 0.16	535 ± 610
GST-192	19.3 ± 0.3	0.10 ± 0.17	786 ± 725	17.8 ± 0.6	0.09 ± 0.16	823 ± 719	18.0 ± 0.7	0.07 ± 0.14	587 ± 582
GST-64S	19.7 ± 0.7	0.39 ± 0.23	150 ± 305	20.4 ± 1.6	0.40 ± 0.23	143 ± 334	16.5 ± 0.2	0.28 ± 0.17	121 ± 259
GST-192S	13.8 ± 0.7	0.39 ± 0.23	137 ± 309	15.4 ± 1.1	0.41 ± 0.22	126 ± 317	13.4 ± 0.2	0.29 ± 0.18	139 ± 316
Proposed	$\mathbf{6.2 \pm 0.5}$	$\mathbf{0.82 \pm 0.14}$	$\mathbf{1.7 \pm 4.1}$	$\mathbf{9.4 \pm 0.3}$	$\mathbf{0.78 \pm 0.14}$	$\mathbf{1.8 \pm 6.0}$	$\mathbf{7.6 \pm 0.9}$	$\mathbf{0.57 \pm 0.15}$	$\mathbf{7.4 \pm 42.6}$
Oracle	6.5 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	6.5 ± 0.0	0.85 ± 0.06	1.0 ± 0.0	6.5 ± 0.0	0.50 ± 0.23	3.6 ± 24.9

Table 3: Style opinion scores of speech synthesizers.

VCTK, seen speakers				LibriTTS, seen speakers				LibriTTS, unseen speakers			
GST-64	GST-16S	Proposed	Oracle	GST-192	GST-192S	Proposed	Oracle	GST-192	GST-192S	Proposed	Oracle
2.1 ± 1.0	3.3 ± 0.9	3.8 ± 0.4	3.8 ± 0.4	1.4 ± 0.6	2.8 ± 1.0	$\mathbf{3.6 \pm 0.6}$	3.5 ± 0.9	1.2 ± 0.5	2.6 ± 0.9	$\mathbf{3.5 \pm 0.7}$	3.5 ± 0.9

- 제안하는 방식이 기존 방식보다 좋음
- **Quantitative**
  - **GST-16 vs GST-64**
    - **Non-parallel text 상황에서 WER**
    - **Lower capacity -> content leakage problem에 더 강함**
    - **Cosine similarity 차이는 별로 없음**
- **Qualitative**
  - **GT 와 비슷하거나 좋은 성능**

Conclusion

- **Unsupervised style transfer learning with style equalization**

- **GT audio 를 사용하는 방법**
- **Style feature 를 그대로 이용하지않고, 두 style 간의 차이를 style transformation 함수의 입력으로 이용**

끝

감사합니다.