SONY

SL1 stuttgart laboratory 1

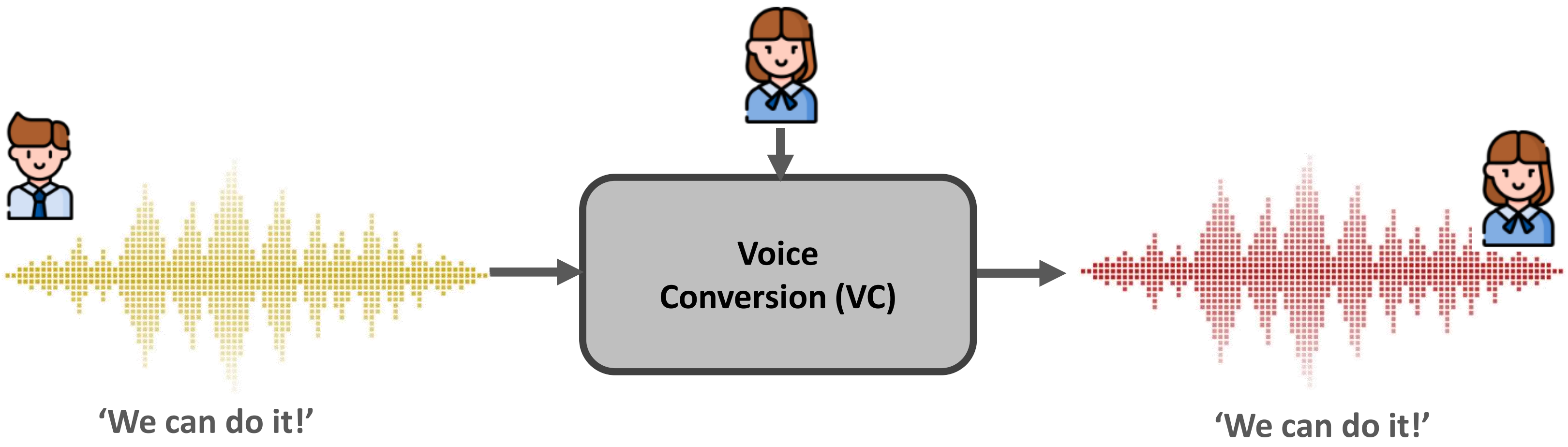
NVC-Net: End-to-End Adversarial Voice Conversion

Bac Nguyen[*], Fabien Cardinaux

Sony Europe B.V., R&D Center, Stuttgart Laboratory 1, Germany

Contents

Problem definition



Transform a recording:

- Converting **non-linguistic information** (speaker identity)
- Preserving **linguistic information** (content)

Why voice conversion?

- Speaker-identity modification
  - Voice dubbings for movies
  - Pronunciation conversion

- Personalized Text-to-Speech systems
  - Provide a simple solution
  - The same sentence said by different people has different effect

- Entertainment
  - Gamming: avatar voices
  - Singing voice conversion

Contents

VC Challenge: Non-parallel training data

- Parallel training data
  - Very sensitive to **misalignment**
  - Expensive to collect

‘We can do it!’

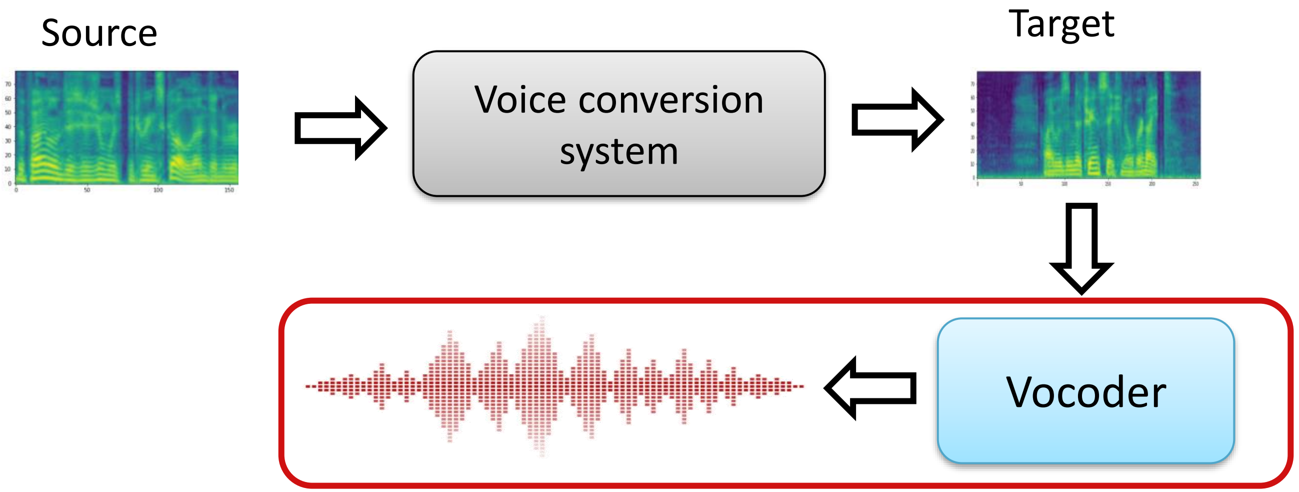
‘We can do it!’

- Non-parallel training data
  - Easy to collect
  - **Difficult to deal** with non-parallel data

‘We can do it!’

‘Hello world!’

VC Challenge: Vocoder dependence

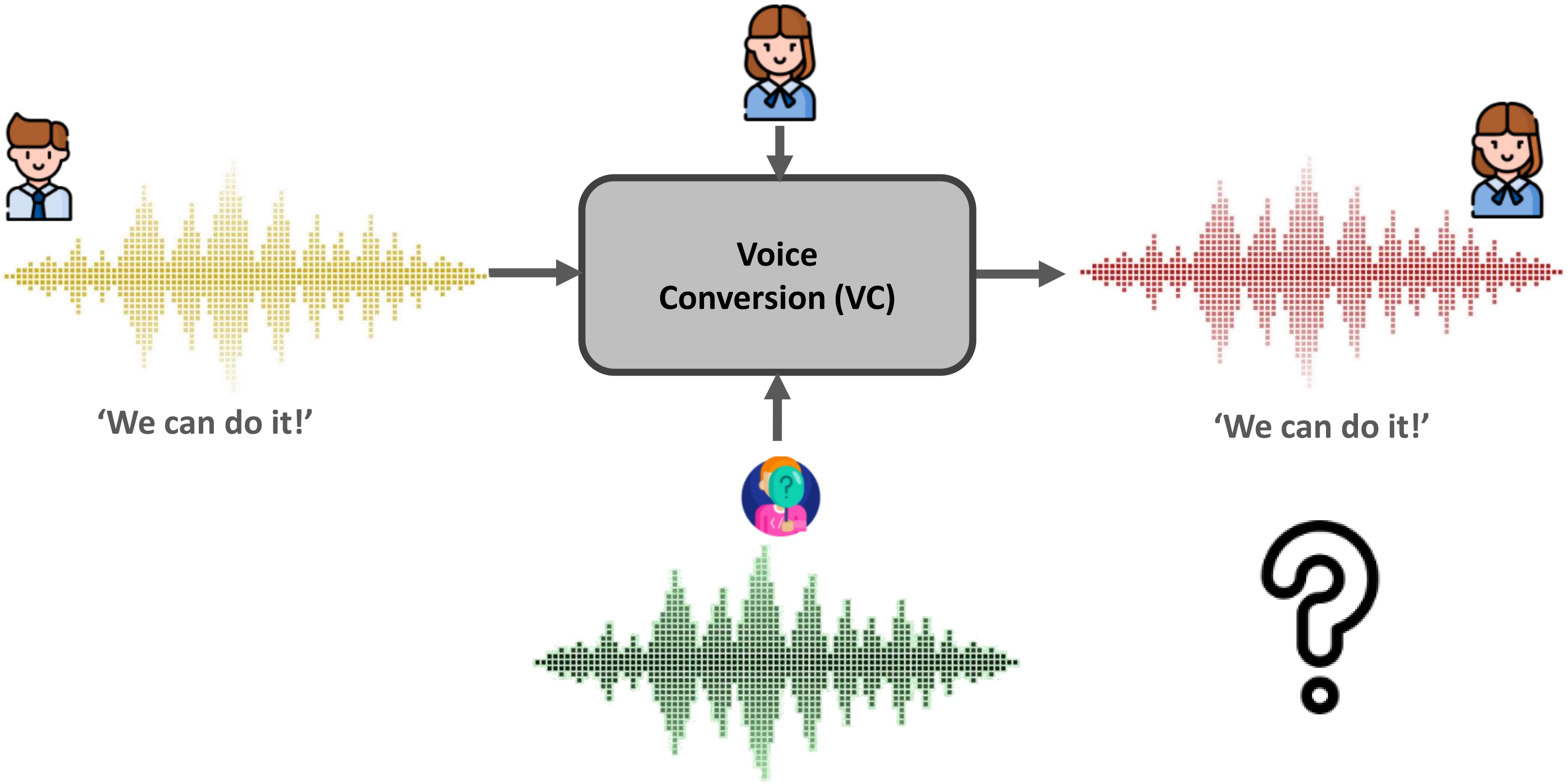


➢ Most of VC systems rely on a vocoder to produce audio waveforms

- Slow at inference time
- Quality of audio is vocoder-dependent
- Feature mismatch problem when training data are limited

VC Challenge: Zero-shot voice conversion

Perform VC from/to speakers that are unseen during training



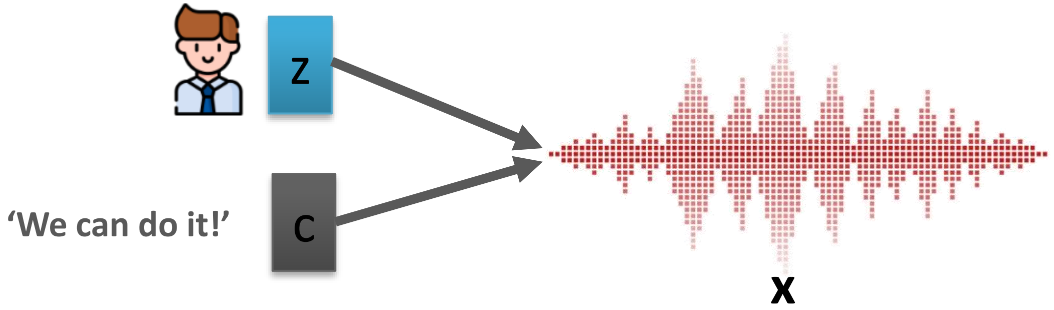
Contents

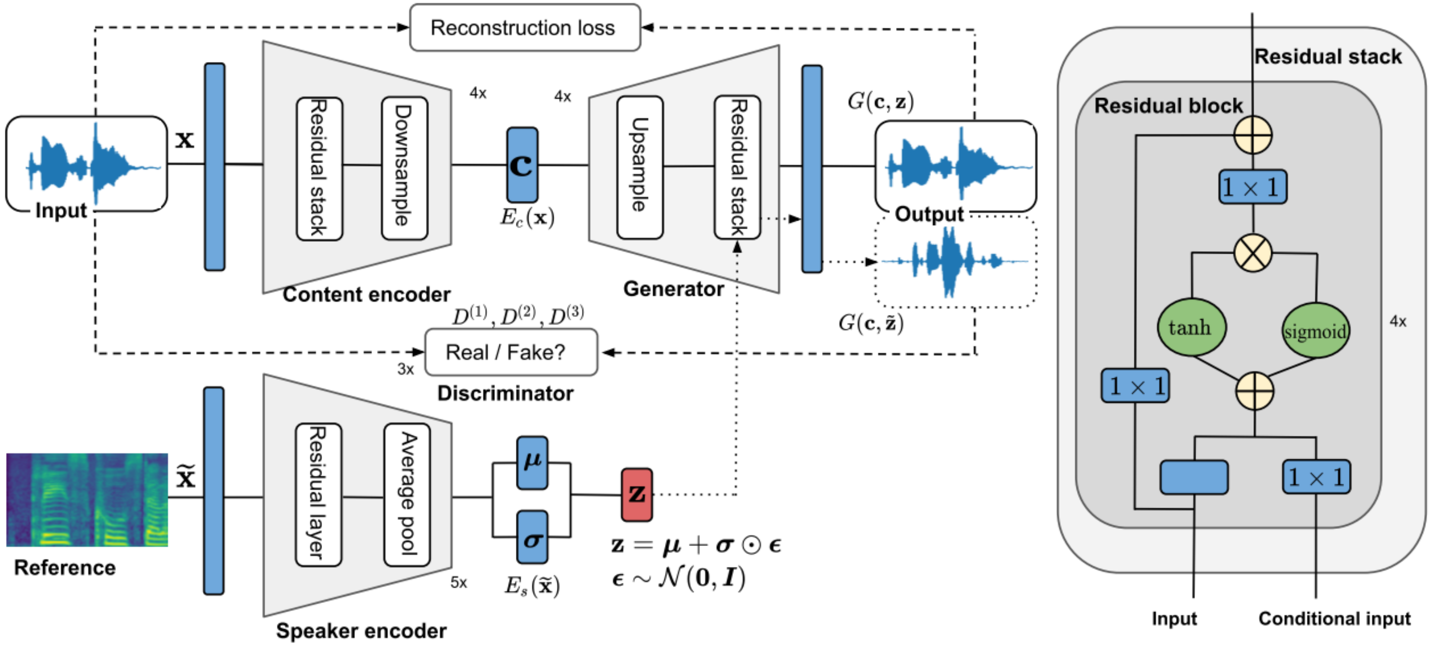
NVC-Net: End-to-end adversarial voice conversion

- **Contributions**
  - NVC-Net can directly generate raw audio without vocoder
  - NVC-Net is very fast at inference
  - NVC-Net supports zero-shot voice conversion

NVC-Net: Network architecture

- An utterance **x** is generated from two latent embeddings
  - Speaker identity **z**
  - Speech content **c**





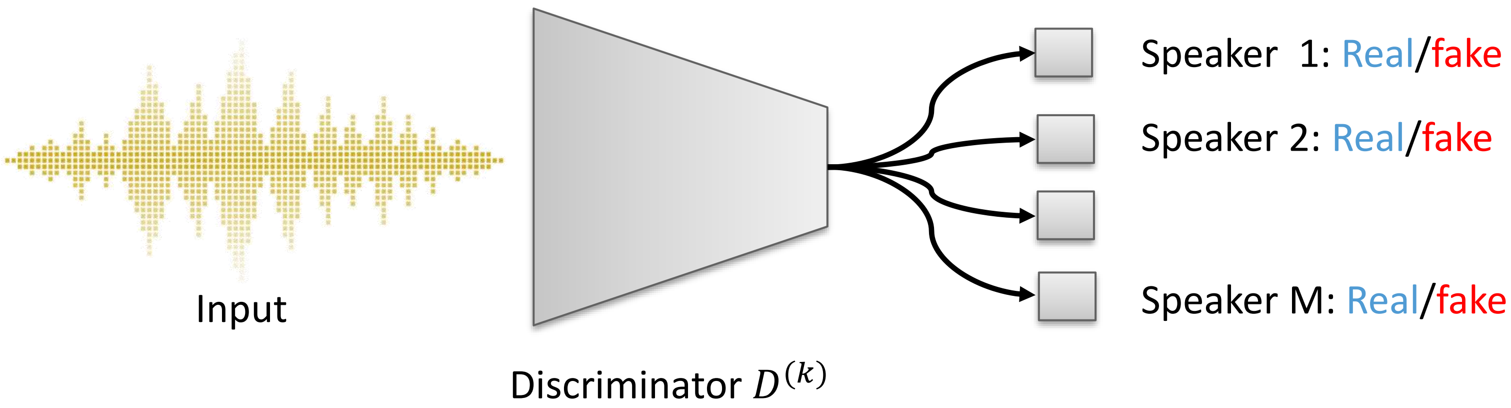
How to disentangle the speaker identity from the speech content?

NVC-Net: Objective functions (I)

➢ Generating high-fidelity audio for a target speaker

$$\mathcal{L}_{\text{adv}}(D^{(k)}) = -\mathbb{E}_{\mathbf{x},y}\left[\log D^{(k)}(\mathbf{x})[y]\right] - \mathbb{E}_{\mathbf{c},\tilde{\mathbf{z}},\tilde{y}}\left[\log\left(1 - D^{(k)}(G(\mathbf{c},\tilde{\mathbf{z}}))[\tilde{y}]\right)\right],$$

$$\mathcal{L}_{\text{adv}}(E_c, E_s, G) = \sum_{k=1}^{3}\mathbb{E}_{\mathbf{c},\tilde{\mathbf{z}},\tilde{y}}\left[\log\left(1 - D^{(k)}(G(\mathbf{c},\tilde{\mathbf{z}}))[\tilde{y}]\right)\right].$$

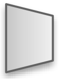


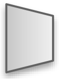
NVC-Net: Objective functions (II)

- Reconstructing highly-perceptually-similar audio waveform from latent embeddings

  - Feature matching loss

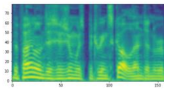
$$\mathcal{L}_{\text{fm}}^{(k)}(E_c, E_s, G) = \mathbb{E}_{\mathbf{c},\mathbf{z},\mathbf{x}}\left[\sum_{i=1}^{L} \frac{1}{N_i}\left\|D_i^{(k)}(\mathbf{x}) - D_i^{(k)}\big(G(\mathbf{c},\mathbf{z})\big)\right\|_1\right]$$

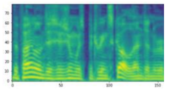




  - Spectral loss

$$\mathcal{L}_{\text{spe}}^{(w)}(E_c, E_s, G) = \mathbb{E}_{\mathbf{c},\mathbf{z},\mathbf{x}}\left[\left\|\theta(\mathbf{x}, w) - \theta\big(G(\mathbf{c},\mathbf{z}), w\big)\right\|_2^2\right]$$





NVC-Net: Objective functions (III)

- Preserving the speaker-invariant information during the conversion
  - Converted utterance preserves the speaker-invariant characteristics of its input audio

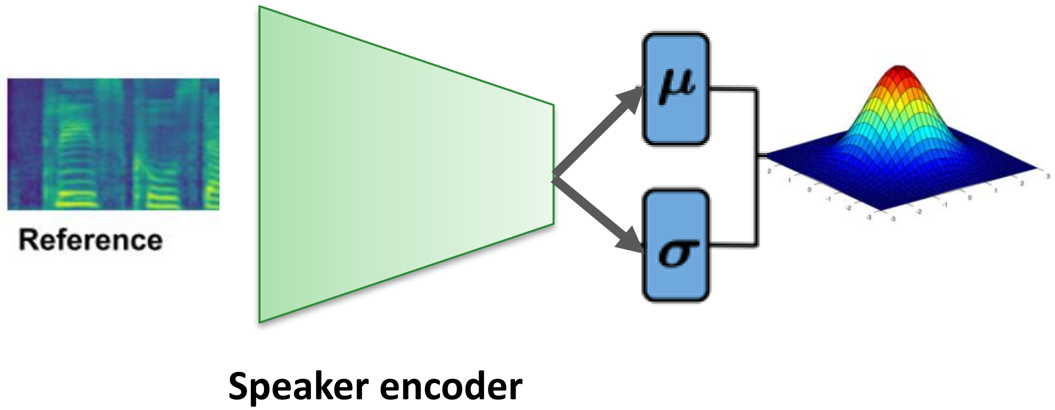
$$\mathcal{L}_{\text{con}}(E_c, G) = \mathbb{E}_{\mathbf{x},\widetilde{\mathbf{z}}}\left[\left\|E_c(\mathbf{x}) - E_c\big(G(E_c(\mathbf{x}), \widetilde{\mathbf{z}})\big)\right\|_2^2\right]$$

- There are two benefits:
  - This allows cycle conversion
  - Disentangling the speaker identity from the speech content

NVC-Net: Objective functions (IV)

- Perform stochastic sampling from the speaker latent space
  - Penalize the deviation of the speaker output distribution from a prior Gaussian

$$\mathcal{L}_{\text{kl}}(E_s) = \mathbb{E}_{\mathbf{x}}\left[\mathbb{D}_{\text{KL}}\left(p(\mathbf{z}|\mathbf{x})\|\mathcal{N}(\mathbf{z}|\mathbf{0},\mathbf{I})\right)\right]$$



Two ways to sample a speaker emebedding:

- from the prior distribution $\boldsymbol{N}(\boldsymbol{z}|\mathbf{0};\boldsymbol{I})$
- from $p(\mathbf{z}/\mathbf{x})$ for a reference utterance **x**

Contents

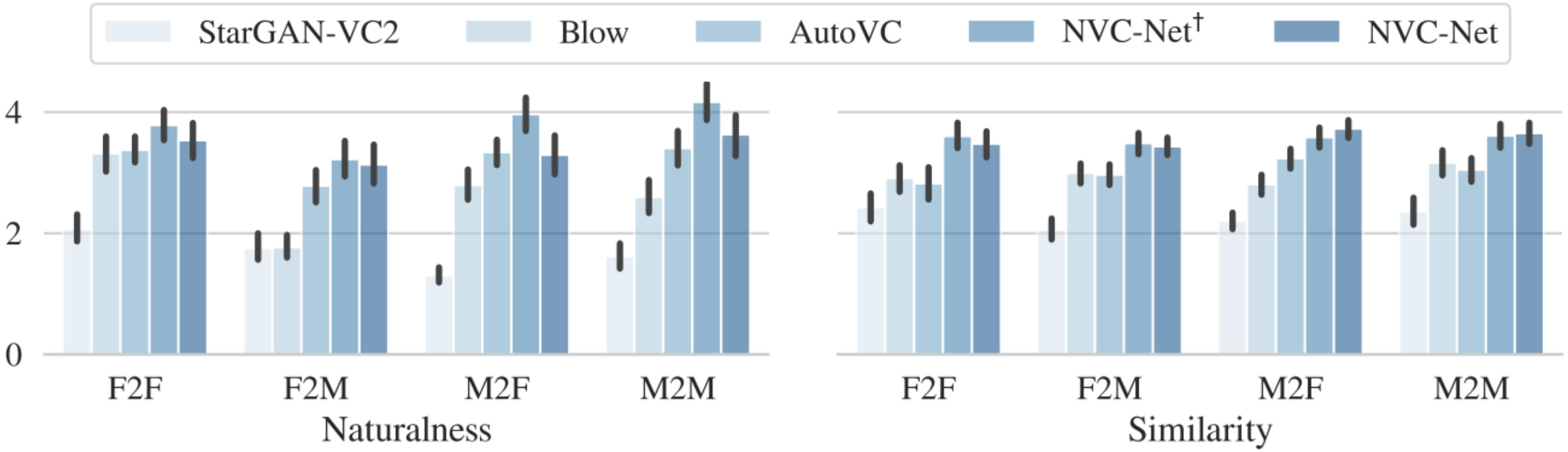
Results: Objective evaluations

- Spoofing (% of correctly classified)
  - The classifier is an Melspectrogram-based convolutional classifier
  - The classifier reaches 99% of accuracy on real speech
    - Training set: 37,508 samples
    - Test set: 4,235 samples

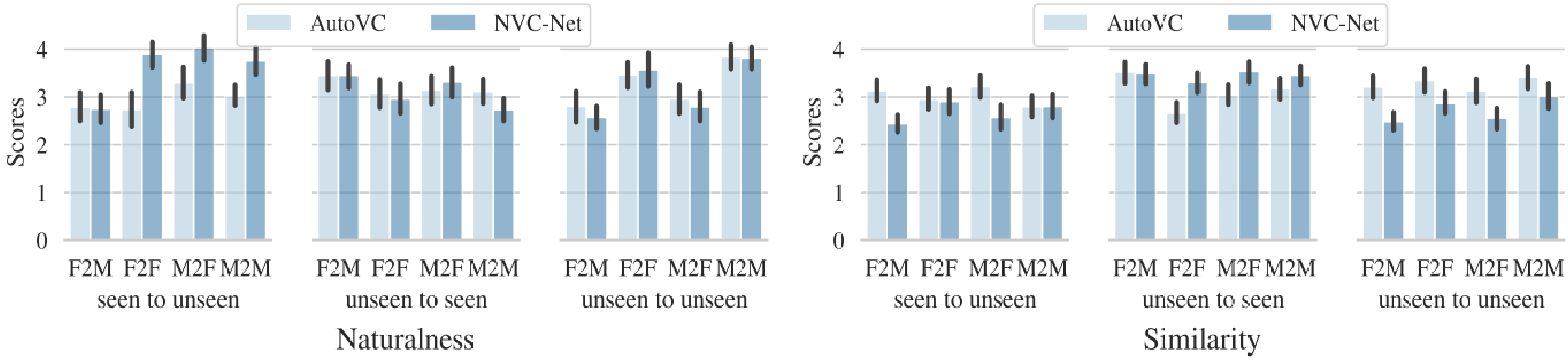
Table 1: Spoofing evaluations of the competing methods

Model	StarGAN-VC2	AutoVC	Blow	NVC-Net†	NVC-Net
Spoofing	19.08	82.46	89.39	**96.43**	93.66

Results: Subjective evaluation



Subjective evaluation for traditional VC settings with 95% confidence intervals



Subjective evaluation for zero-shot VC settings with 95% confidence intervals

Results: Ablation studies

Speaker identification accuracy

Model	Content	Speaker
NVC-Net†	19.21	N/A
NVC-Net	24.15	99.22

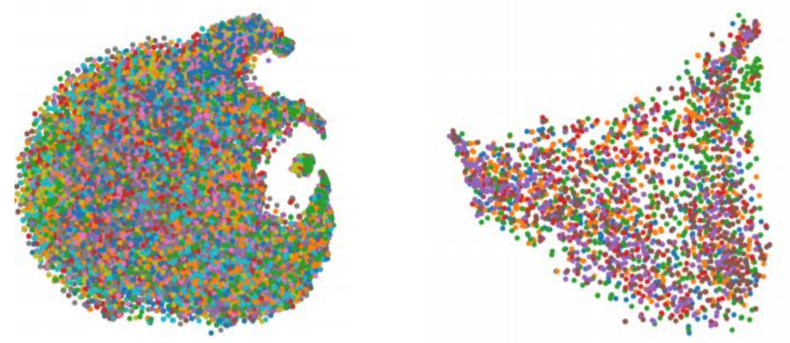
Model size and inference speed comparisons

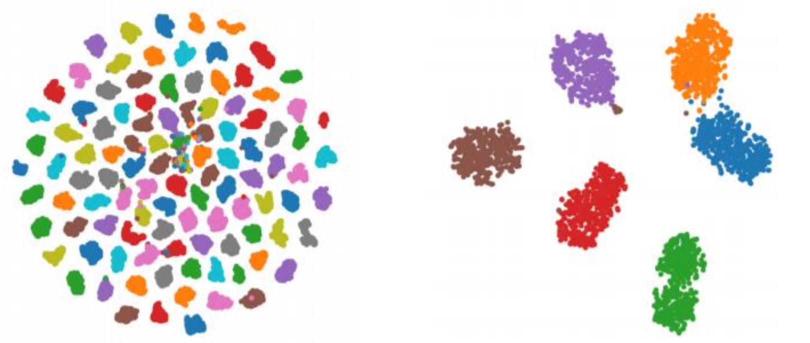
Model	# parameters (in millions)	Inference speed GPU (in kHz)	Inference speed CPU (in kHz)
StarGAN-VC2*	**9.62**	60.47	**35.47**
AutoVC*	28.42	0.11	0.04
Blow	62.11	441.11	2.43
NVC-Net	15.13	**3661.65**	7.49

Less memory footprint

Very fast on GPU

Close to real time on CPU


Content embeddings


Speaker embeddings

Barnes-Hut t-SNE visualization

Demo: https://nvcnet.github.io/

Scan QR code for the code and demo page

SONY