

Trustworthy AI Systems

-- Voice Conversion

Instructor: Guangjing Wang

guangjingwang@usf.edu

Last Lecture

- Speech Recognition
- Speaker Recognition
 - Speaker Verification
 - Speaker Identification
- Humans as Deepfake Audio Detectors

This Lecture

Voice Conversion

- One-shot Voice Conversion by Separating Speaker and Content Representations with **Instance Normalization** (InterSpeech'2019)
- AGAIN-VC: A one-shot voice conversion using activation guidance and adaptive **instance normalization**. (ICASSP'2021)
- VQVC+: one-shot voice conversion by **vector quantization** and U-Net architecture. (ICSA'2020)
- AVQVC: One-shot voice conversion by **vector quantization** with applying contrastive learning. (ICASSP'2022)
- FREEVC: Towards High-quality Text-Free **One-Shot** Voice Conversion (ICASSP'2023)

Voice Conversion (VC)

- Voice = Content Information (e.g., semantics) + Speaker Features (e.g., timbre, accent, and tones)
- Voice Conversion: transforms the source speaker's into another one's while preserving the linguistic content

None Disentangle-based Methods

- Statistics-based Methods: need parallel data
 - Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory (ICASSP'2007)
 - Voice conversion using partial least squares regression (IEEE Trans. ASLP 2010)
- Generative Models: unparalleled data (domain mapping, target speak in training)
 - GAN-based Method: around 2018
 - Cyclegan-vc: **Non-parallel** voice conversion using cycle-consistent adversarial network
 - Stargan-vc: **Non-parallel** many-to-many voice conversion using star generative adversarial networks
 - VAE-based Method: around 2018
 - **Non-parallel** many-to-many voice conversion with auxiliary classifier variational autoencoder
 - Voice conversion from unaligned corpora using variational autoencoding Wasserstein GAN

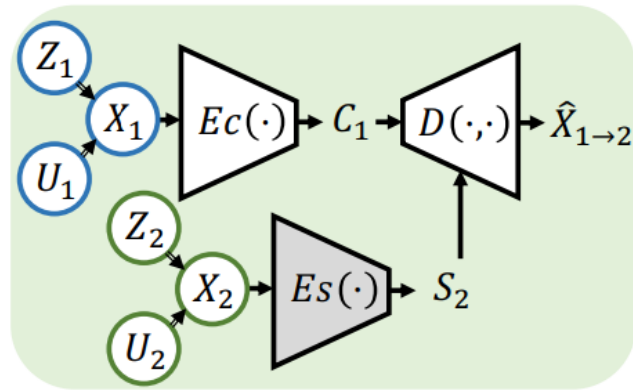
Disentangle-based Methods

- Instance normalization (IN)
 - One-shot voice conversion by separating speaker and content representations with **instance normalization** (InterSpeech'2019)
 - AGAIN-VC: A **one-shot** voice conversion using activation guidance and adaptive **instance normalization**. (ICASSP'2021)
- Vector quantization (VQ)
 - VQVC+: **One-shot** voice conversion by **vector quantization** and **u-net** architecture. (ICSA'2020)
 - AVQVC: **One-shot** voice conversion by **vector quantization** with applying **contrastive learning**. (ICASSP'2022)

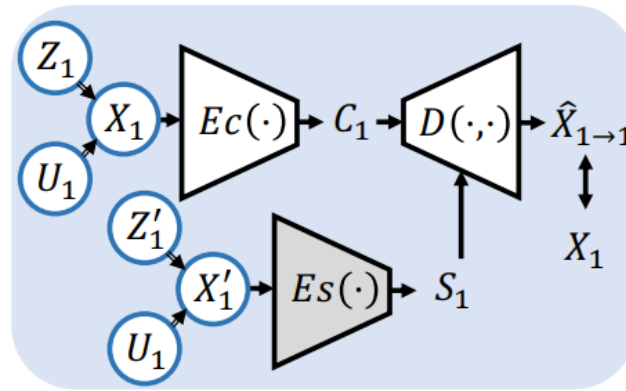
Disentangle-based Methods: Encoder-Decoder Structure

- Encoder: Speech => Latent Representation
 - Encoding the **source** speech's **content**
 - Encoding the source speech's speaker features
 - Encoding the target speech's content
 - Encoding the **target** speech's **speaker features**
- Decoder: Latent Representation => Speech
 - Fusing the source speech's content and target speaker features
 - Decoding the Fused Representation to Speech Data

An Example: AutoVC (ICML'19): Layer Dimension



(a) Conversion



(b) Training

E(S): Pretrained Speaker Verification Model

E(C): Content Extraction Model: to be trained
Autoencoder with a carefully designed bottleneck

D(): Decoder

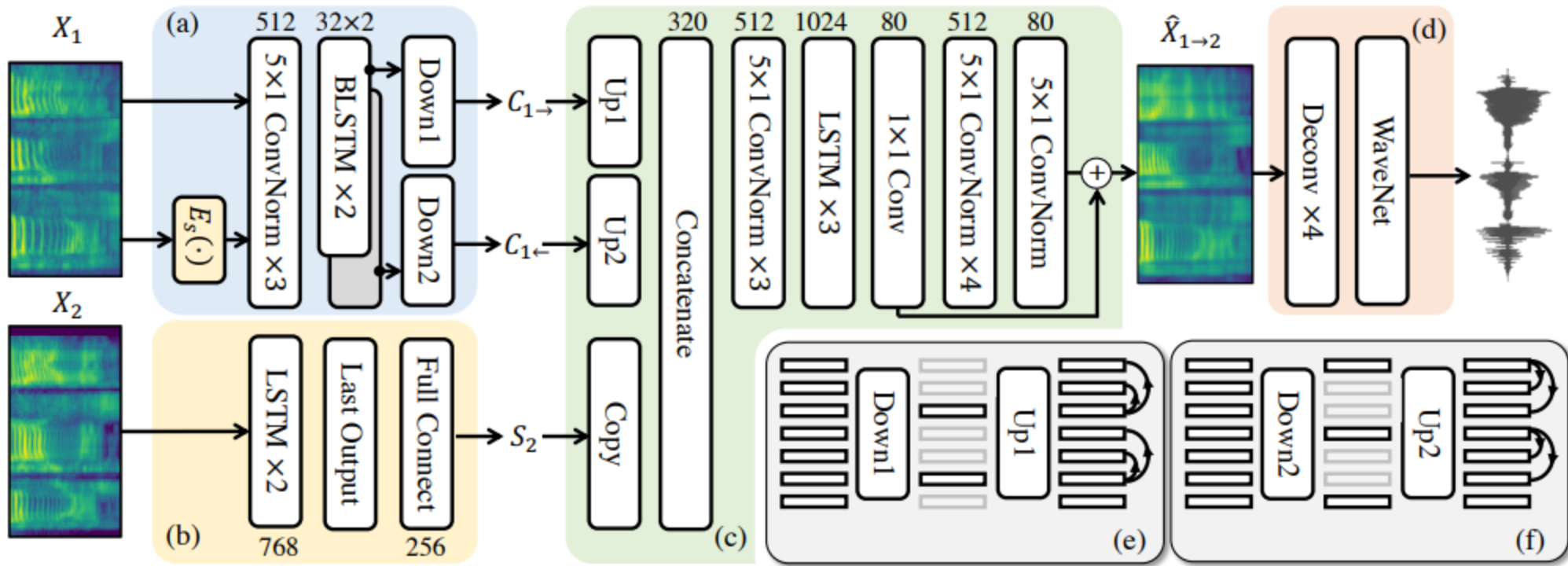
where

$$\min_{E_c(\cdot), D(\cdot, \cdot)} L = L_{\text{recon}} + \lambda L_{\text{content}},$$

$$L_{\text{recon}} = \mathbb{E}[\|\hat{X}_{1 \rightarrow 1} - X_1\|_2^2],$$

$$L_{\text{content}} = \mathbb{E}[\|E_c(\hat{X}_{1 \rightarrow 1}) - C_1\|_1].$$

An Example: AutoVC (ICML'19)



Es and (b):The style encoder

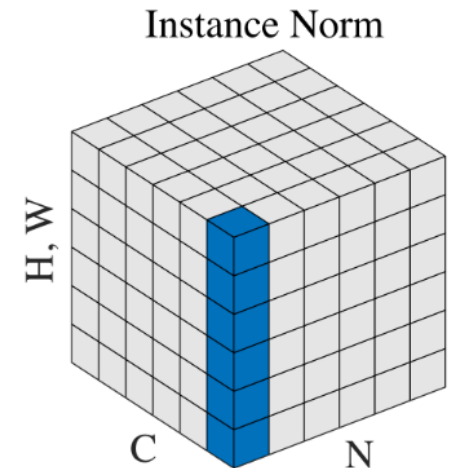
AutoVC Architecture

Instance Normalization based Disentanglement (1)

Instance Normalization

$$\mu_{ti} = \frac{1}{HW} \sum_{l=1}^W \sum_{m=1}^H x_{tilm}, \quad \sigma_{ti}^2 = \frac{1}{HW} \sum_{l=1}^W \sum_{m=1}^H (x_{tilm} - \mu_{ti})^2, \quad y_{tijk} = \frac{x_{tijk} - \mu_{ti}}{\sqrt{\sigma_{ti}^2 + \epsilon}},$$

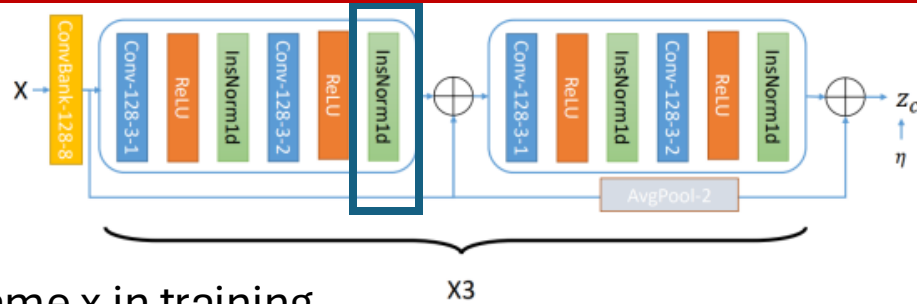
based on the description in Section 2.1. In this paper, we find that simply adding Instance normalization (IN) without affine transformation to E_c can remove the speaker information while preserving the content information. Similar idea has been verified to be effective for style transfer in computer vision [28].



Instance Normalization based Disentanglement (2)

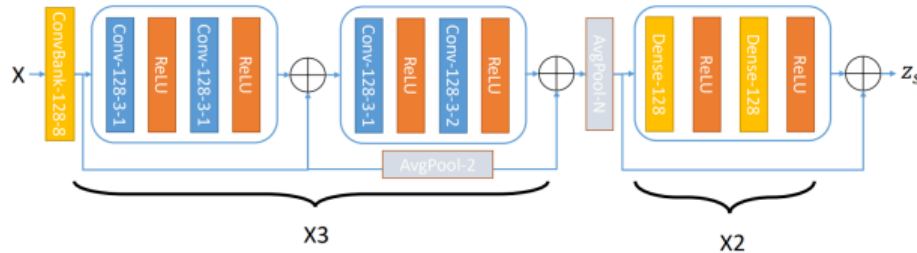
Mirror Structure

Content encoder

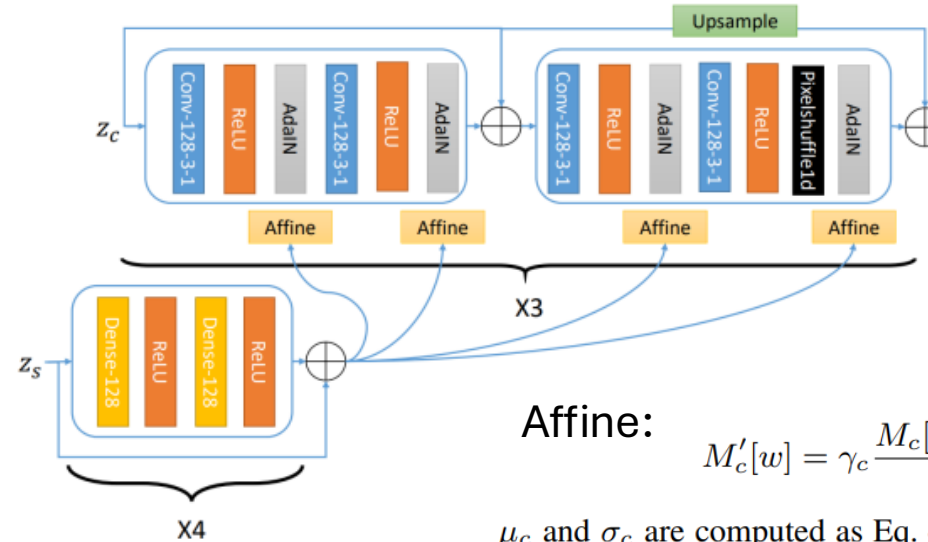


Same x in training

Speaker encoder



Decoder



Adaptive instance normalization (AdaIN) layer aligns the mean and variance of the content features with those of the style features.

Affine:

$$M'_c[w] = \gamma_c \frac{M_c[w] - \mu_c}{\sigma_c} + \beta_c. \quad (6)$$

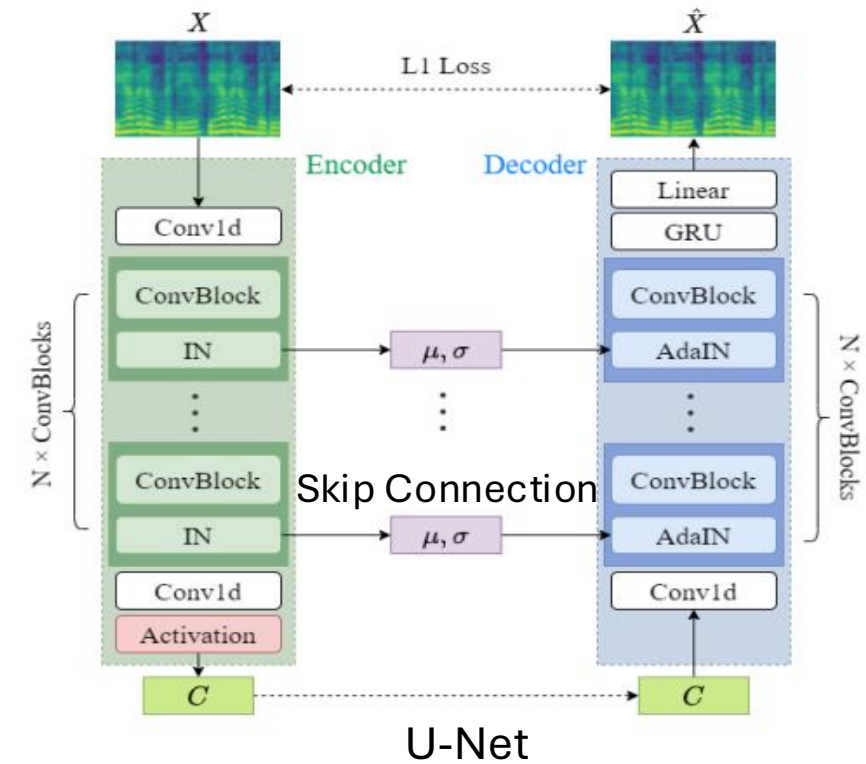
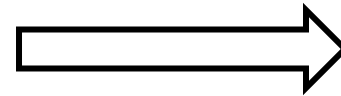
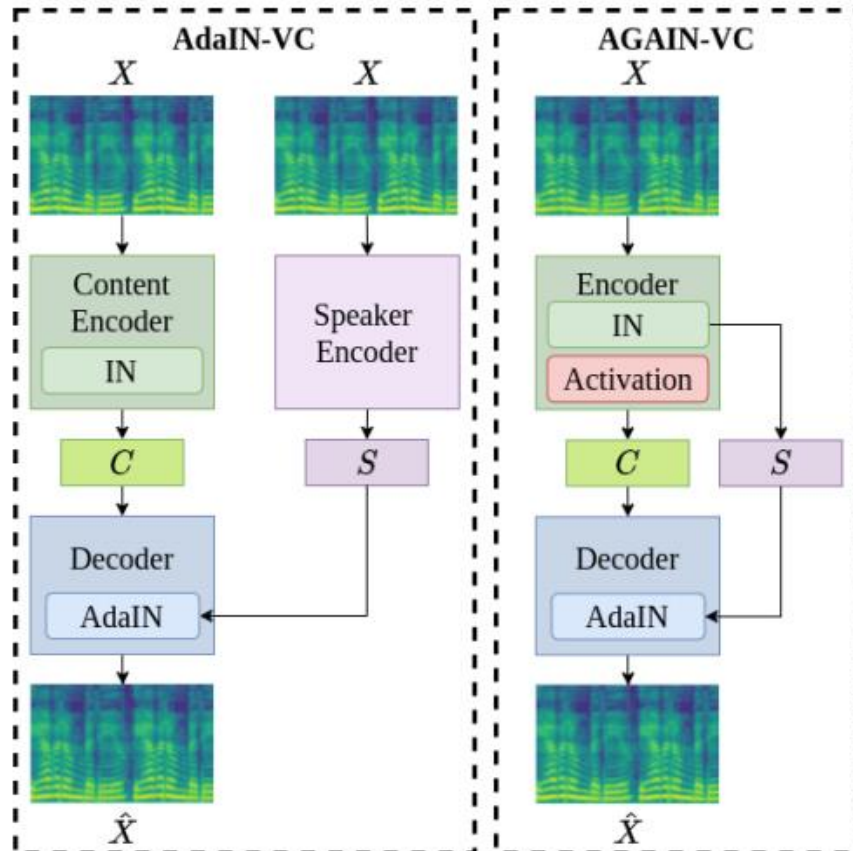
μ_c and σ_c are computed as Eq. 4. γ_c and β_c for each channel are the linear transformation of the output of speaker encoder E_s .

$$\min_{\theta_{E_s}, \theta_{E_c}, \theta_D} L(\theta_{E_s}, \theta_{E_c}, \theta_D) = \lambda_{rec} L_{rec} + \lambda_{kl} L_{kl}$$

$$L_{rec}(\theta_{E_s}, \theta_{E_c}, \theta_D) = \mathbb{E}_{x \sim p(x), z_c \sim p(z_c|x)} [\|D(E_s(x), z_c) - x\|_1].$$

$$L_{kl}(\theta_{E_c}) = \mathbb{E}_{x \sim p(x)} [\|E_c(x)^2\|_2^2].$$

Instance Normalization based Disentanglement (3)



$$\text{AdaIN}(\mathbf{H}, \mu(\mathbf{Z}), \sigma(\mathbf{Z})) = \sigma(\mathbf{Z})\text{IN}(\mathbf{H}) + \mu(\mathbf{Z}).$$

AGAIN-VC: A one-shot voice conversion using activation guidance and adaptive instance normalization. (ICASSP'2021)

Instance Normalization based Disentanglement (4)

Activation Guidance: With an extra activation function, the range of content embeddings is somewhat restricted.

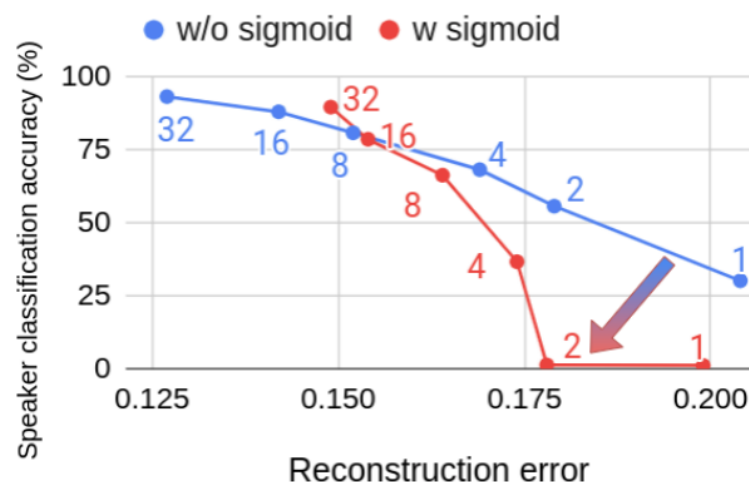


Fig. 3: The trade-off between the reconstruction error and the speaker classification accuracy on C . The numbers here represent the channel size of C . Besides, the arrow stands for the direction of improvement of the models we do care.

Both of them should be as low as possible

Table 1: Comparison between the models with different activation functions. C and S are the speaker classification accuracy on content embeddings and speaker embeddings, respectively, and Rec. represents the reconstruction error. * is our proposed method.

Activation	C (Acc.%) ↓	S (Acc.%) ↑	Rec. ↓
None	68.6	92.6	0.161
ReLU	51.9	92.2	0.174
ELU	69.2	91.5	0.167
Tanh	57.3	91.7	0.165
Sigmoid ($\alpha = 1$)	30.5	90.0	0.167
Sigmoid ($\alpha = 0.1$) *	1.7	93.2	0.151
Sigmoid ($\alpha = 0.01$)	1.7	91.1	0.222

Vector Quantization based Disentanglement (1)

Vector Quantization:

- The content information can be represented by discrete codes;
- The speaker information can be viewed as the difference between the continuous representations and the discrete codes.

$$\begin{aligned} V &= \text{enc}(X), \\ C &= \text{Quantize}(V), \\ s &= \mathbb{E}_t[V - C], \quad S = \underbrace{\{s, s, \dots, s\}}_{T \text{ times}}, \end{aligned}$$

$$\text{Quantize}(V) = \{q_0, q_1, \dots, q_T\}, \quad q_j = \arg \min_{q \in Q} (\|v_j - q\|_2^2).$$

Q: quantization codebook

VQVC+: one-shot voice conversion by **vector quantization** and U-Net architecture. (ICSA'2020)

Vector Quantization based Disentanglement (2)

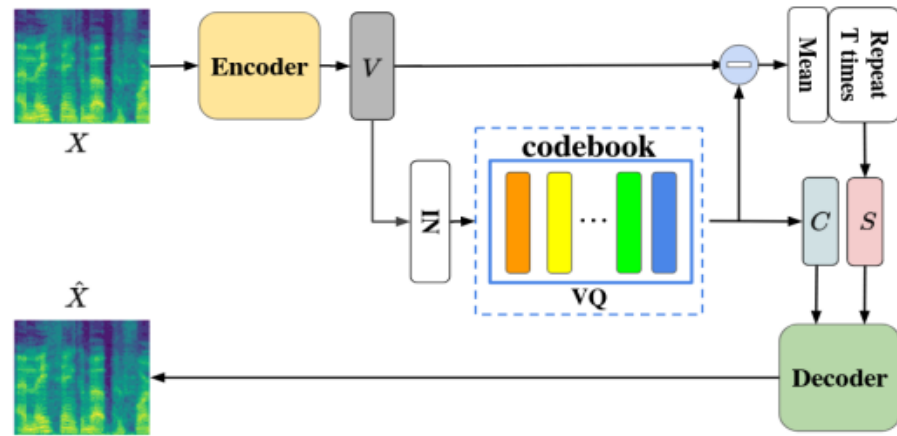


Figure 1: The VQVC architecture. VQ is the vector quantization layer, and IN is the instance normalization layer. VQVC applies IN+VQ layers to separate the content and the speaker information to achieve voice conversion.

$$L_{rec}(\mathcal{Q}, \theta_{enc}, \theta_{dec}) = \mathbb{E}_{\mathbf{X} \in \mathcal{X}} [\|\hat{\mathbf{X}} - \mathbf{X}\|_1^1].$$

$$L_{latent}(\theta_{enc}) = \mathbb{E}_t [\|IN(\mathbf{V}) - \mathbf{C}\|_2^2].$$

$$L = L_{rec} + \lambda L_{latent}.$$

Two 3X1 1D-convolution layers, an IN layer, and a vector quantization layer

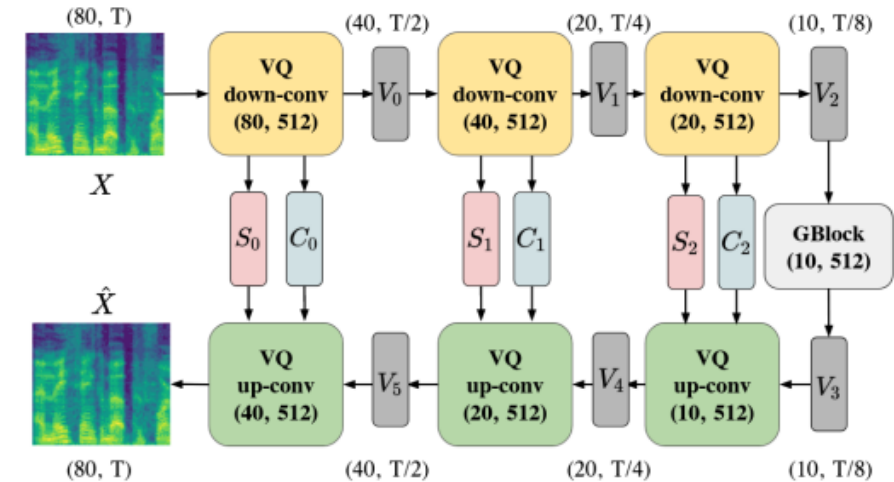


Figure 2: The VQVC+ architecture. VQVC+ applies the U-Net architecture to improve quality, and each sub-module in the encoder is a variant of the VQVC encoder. Quantized output \mathbf{C} and the speaker embedding \mathbf{S} are skip-connected to the decoder instead of the continuous embedding \mathbf{V} .

Up-conv: Two 3X1 1D-convolution layers, and time upsampling, frequency upsampling

Vector Quantization based Disentanglement (3)

The experiment on the output of each encoder layer C_0 , C_1 , C_2 :

Method	$C_0 / C_1 / C_2$ (%)	L1Loss
VQVC	16.0	0.262
Q32	19.5 / 11.8 / 6.8	0.210
Q64	23.2 / 16.6 / 7.0	0.188
Q128	33.3 / 17.0 / 10.3	0.180
Q256	35.8 / 18.1 / 12.5	0.165
IN-only	71.2 / 36.8 / 5	0.145

Table 1: Accuracy of identifying speakers on the content embedding and the speaker embedding with different methods. VQVC is the model without skip-connection design. QN means that the size of codebook, Q , in VQVC+ is N . IN-only means no quantization in U-Net. L1Loss is the L1 reconstruction loss.

Method	$S_0 / S_1 / S_2$ (%)
VQVC	96.6
Q64	98.3 / 72.2 / 45.4
IN-only	97.4 / 80.1 / 23.1

Table 2: Accuracy of identifying speakers on the speaker embedding S .

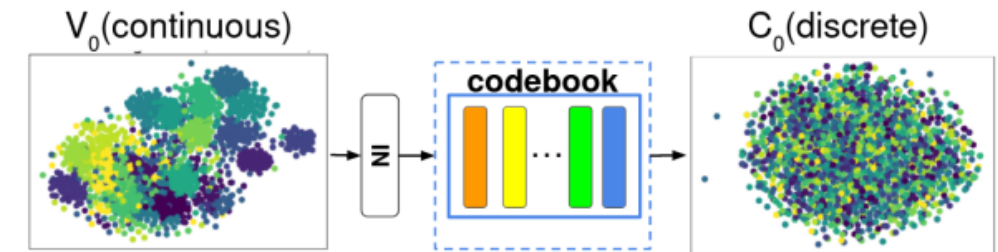
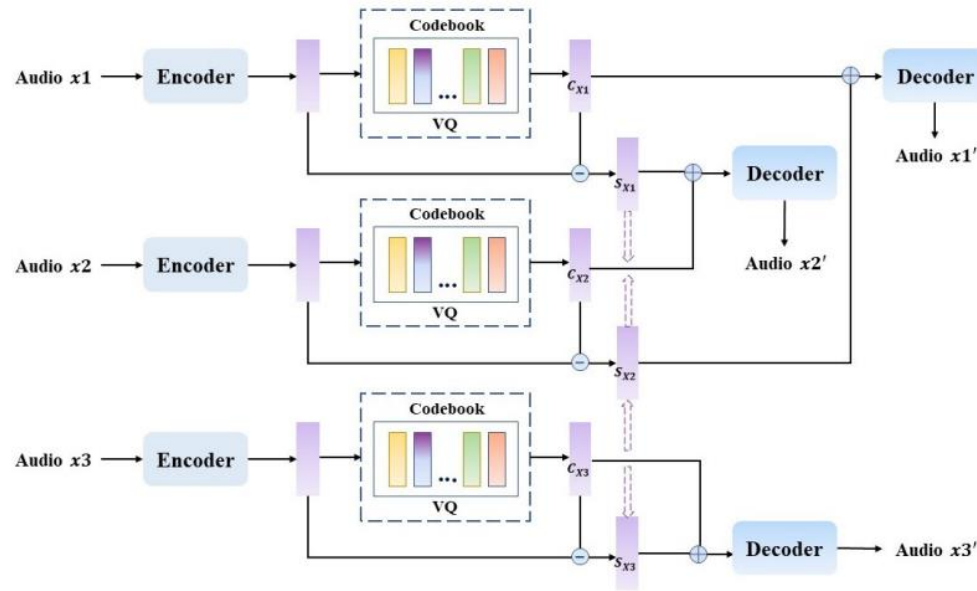


Figure 6: quantization

Vector Quantization based Disentanglement (4)



$$\mathcal{L}_{\text{recon}} = \|x'_1 - x_1\|_1^1 + \|x'_2 - x_2\|_1^1 + \|x'_3 - x_3\|_1^1$$

$$\begin{aligned} \mathcal{L}_{\text{latent}} = & \|enc(x_1) - C_{x_1}\|_2^2 \\ & + \|enc(x_2) - C_{x_2}\|_2^2 + \|enc(x_3) - C_{x_3}\|_2^2 \end{aligned}$$

$$\mathcal{L}_{\text{speaker}} = \|S_{x_2} - S_{x_1}\|_1^1$$

$$\mathcal{L}_{\text{diff}} = -(\|S_{x_2} - S_{x_3}\|_1^1 + \|S_{x_1} - S_{x_3}\|_1^1)$$

Fig. 2. Framework of AVQVC. Both x_1 and x_2 are produced by the same speaker, but their text content are different, while x_3 belongs to another speaker. C_X is a discrete variable generated by looking up the *codebook*. And, S_X denotes speaker embedding, which is produced by the mean difference between encoder output and C_X .

$$L = \mathcal{L}_{\text{recon}} + \alpha \mathcal{L}_{\text{latent}} + \beta \mathcal{L}_{\text{speaker}} + \lambda \mathcal{L}_{\text{diff}}$$

AVQVC: **One-shot** voice conversion by **vector quantization** with applying **contrastive learning**.

Vector Quantization based Disentanglement (5)

Table 1. Comparison of different models in traditional VC and one-shot vc.

Methods	Traditional VC			One-Shot VC			MODEL-SIZE
	MCD	MOS	VSS	MCD	MOS	VSS	
VQVC	8.16 ± 0.31	2.28 ± 0.99	3.47 ± 0.82	8.12 ± 0.14	2.06 ± 0.84	2.97 ± 0.75	5.71M
VQVC+	7.08 ± 0.22	3.31 ± 0.90	3.42 ± 0.85	8.41 ± 0.08	2.75 ± 0.84	3.11 ± 0.88	388M
AutoVC	4.34 ± 0.12	3.81 ± 1.14	3.45 ± 0.76	7.66 ± 0.17	2.61 ± 0.73	2.91 ± 0.72	339M
StarGAN-VC2	6.28 ± 0.09	3.45 ± 1.01	3.59 ± 0.87	—	—	—	56.45M
AVQVC(512)	5.19 ± 0.29	3.57 ± 0.91	3.70 ± 0.71	5.04 ± 0.13	3.20 ± 0.91	3.29 ± 0.64	5.77M

The Mel-Cepstral Distortion(MCD) between converted speech and the ground truth target speech

$$\frac{10\sqrt{2}}{\ln 10} \frac{1}{T} \sum_{t=1}^T \sqrt{\sum_i (C_{ti} - \hat{C}_{ti})^2}.$$

The mean opinion score(MOS) test, which is used to evaluate the quality of converted speech

The voice similarity score(VSS) test, which measures how similar the timbre of the converted voice is to that of the ground truth.

FreeVC

Limitations of existing work:

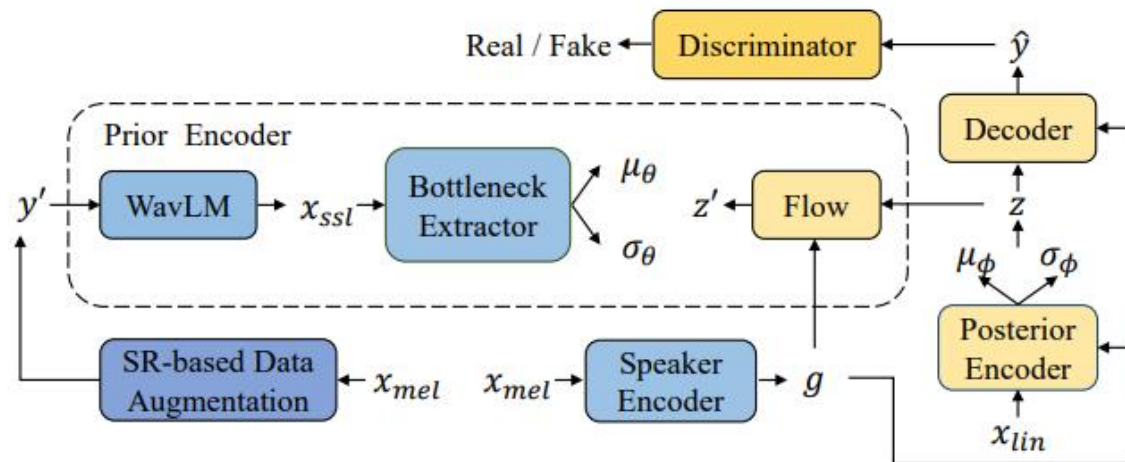
- Extract dirty content information with speaker information leaked in (text-free/disentanglement)
- Demand a large amount of annotated data for training (text-based VC)
- The quality of reconstructed waveform can be degraded

Two steps of VC:

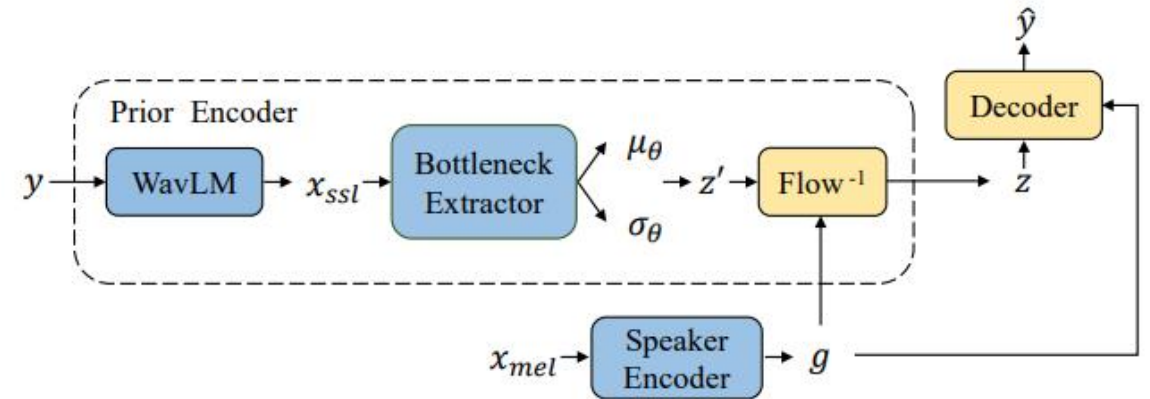
1. A conversion model converts the source acoustic features into target speaker's voice
2. A pre-trained **vocoder** transforms the converted features into waveform
 - A different distribution from that the **vocoder** uses during training → degrading the quality

FreeVC (1)

Bottleneck extractor extracts content information



(a) Training procedure

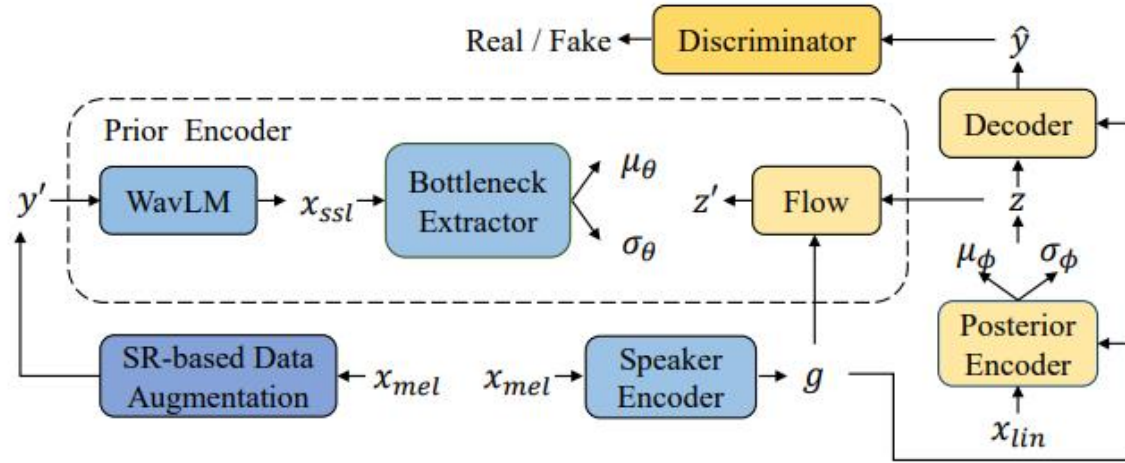


(b) Inference procedure

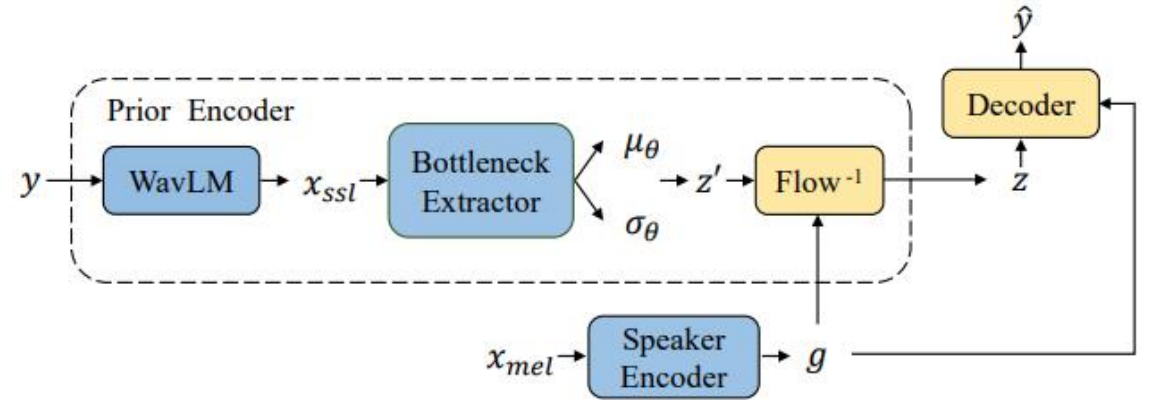
Fig. 1: Training and inference procedure of FreeVC. Here y denotes source waveform, y' denotes augmented waveform, \hat{y} denotes converted waveform, x_{mel} denotes mel-spectrogram, x_{lin} denotes linear spectrogram, x_{ssl} denotes SSL feature, and g denotes speaker embedding.

FreeVC (2)

μ_θ and d -dim σ_θ . The normalizing flow, which conditions on speaker embedding g , is adopted to improve the complexity of prior distribution. Following VITS, it is composed of multiple affine coupling layers [18] and is made to be volume-preserving with the Jacobian determinant $|\det \frac{\partial z'}{\partial z}|$ of 1.



(a) Training procedure



(b) Inference procedure

Fig. 1: Training and inference procedure of FreeVC. Here y denotes source waveform, y' denotes augmented waveform, \hat{y} denotes converted waveform, x_{mel} denotes mel-spectrogram, x_{lin} denotes linear spectrogram, x_{ssl} denotes SSL feature, and g denotes speaker embedding.

FreeVC (3): SR-based Data Augmentation

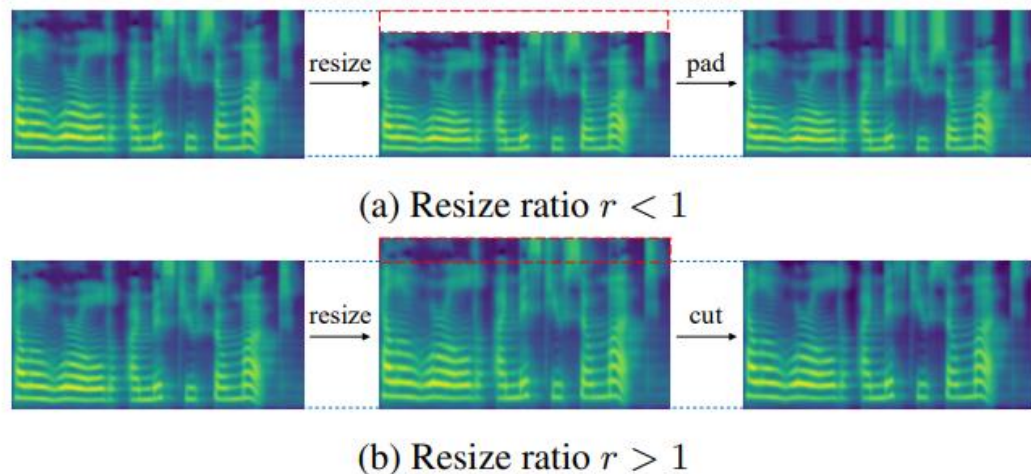
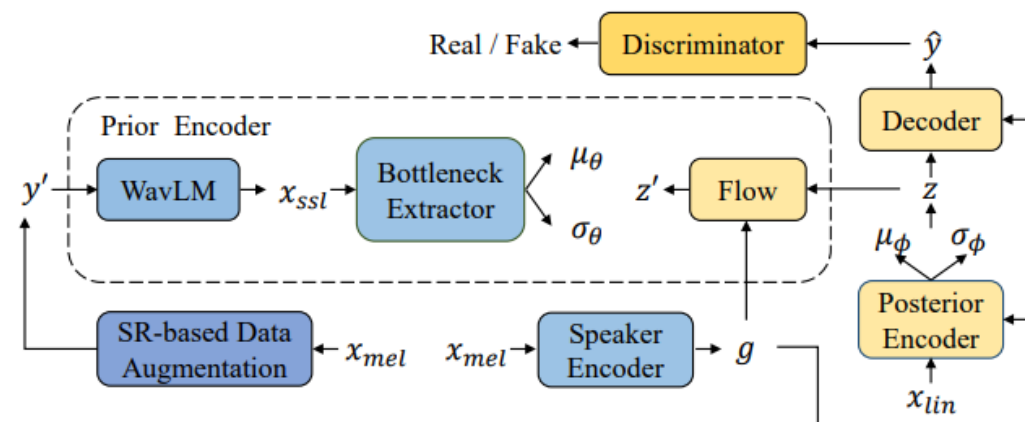


Fig. 2: Vertical spectrogram-resize operation.

L_{fm} is the feature matching loss



$$q_{\phi}(z|x_{lin}) = N(z; \mu_{\phi}, \sigma_{\phi}^2),$$

$$p_{\theta}(z|c) = N(z'; \mu_{\theta}, \sigma_{\theta}^2) \left| \det \frac{\partial z'}{\partial z} \right|.$$

GAN

CVAE

$$L(G) = L_{rec} + L_{kl} + L_{adv}(G) + L_{fm}(G), \quad (3)$$

$$L(D) = L_{adv}(D). \quad (4)$$

FreeVC (3): SR-based Data Augmentation

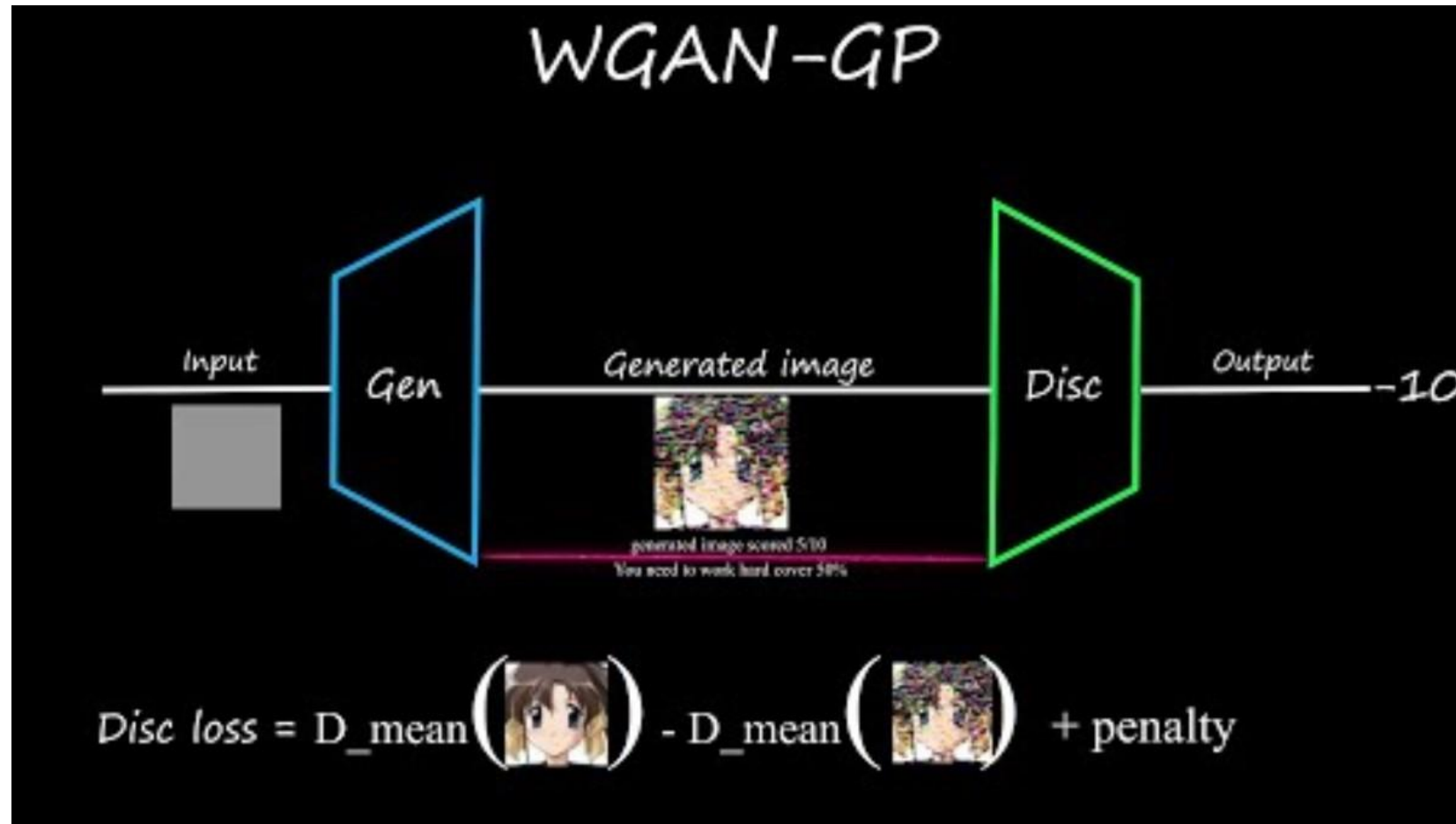
Table 1: Subjective evaluation results in terms of 5-scale MOS and SMOS with 95% confidence intervals under seen-to-seen, unseen-to-seen and unseen-to-unseen scenarios. For reference, we also report scores of source utterances.

	seen-to-seen		unseen-to-seen		unseen-to-unseen	
	MOS	SMOS	MOS	SMOS	MOS	SMOS
VQMIVC	2.31±0.09	2.10±0.08	1.50±0.08	1.71±0.08	1.49±0.08	1.29±0.05
BNE-PPG-VC	2.80±0.12	2.95±0.12	2.89±0.10	2.83±0.10	3.44±0.08	2.63±0.10
YourTTS	3.46±0.10	3.25±0.09	2.54±0.10	2.50±0.10	2.87±0.09	1.97±0.09
FreeVC	3.99±0.09	3.80±0.09	4.06±0.08	3.77±0.09	4.06±0.08	2.83±0.08
FreeVC (w/o SR)	3.85±0.10	3.50±0.10	3.88±0.08	3.58±0.08	3.97±0.09	2.80±0.09
FreeVC-s	4.01±0.09	3.75±0.09	4.08±0.08	3.68±0.09	4.02±0.09	2.78±0.09
Source	4.32±0.08	-	4.11±0.10	-	4.17±0.09	-

The mean opinion score(MOS) test, which is used to evaluate the quality of converted speech.

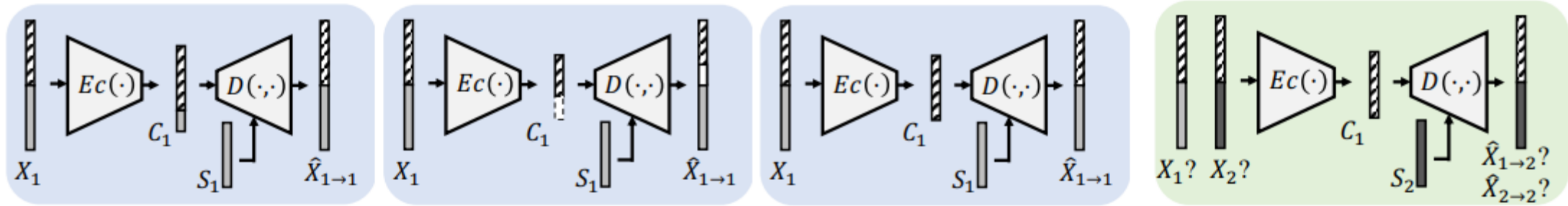
The Sage Instruments Mean Opinion Score (SMOS) test line provides an accurate assessment of how telephone users perceive speech quality.

Wasserstein GAN (Take a break)



<https://www.youtube.com/watch?v=QJOEmwvnmTM>

Is there perfect disentanglement?



(a) Bottleneck too wide

(b) Bottleneck too narrow

Content is
damaged!
Already bad

(c) Bottleneck just right

Did not see the perfect
conversion in real life.

(d) Conversion

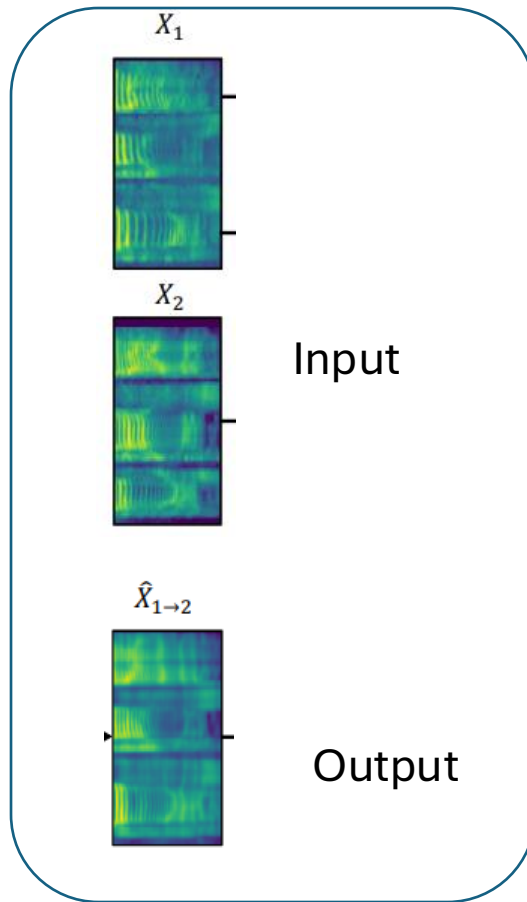
1. Perfect reconstruction is achieved.
2. The content embedding C_1 does not contain any information about the source speaker U_1 , which we refer to as *speaker disentanglement*.

We will now show by contradiction how these two properties imply an ideal conversion. Suppose when AUTOVC is

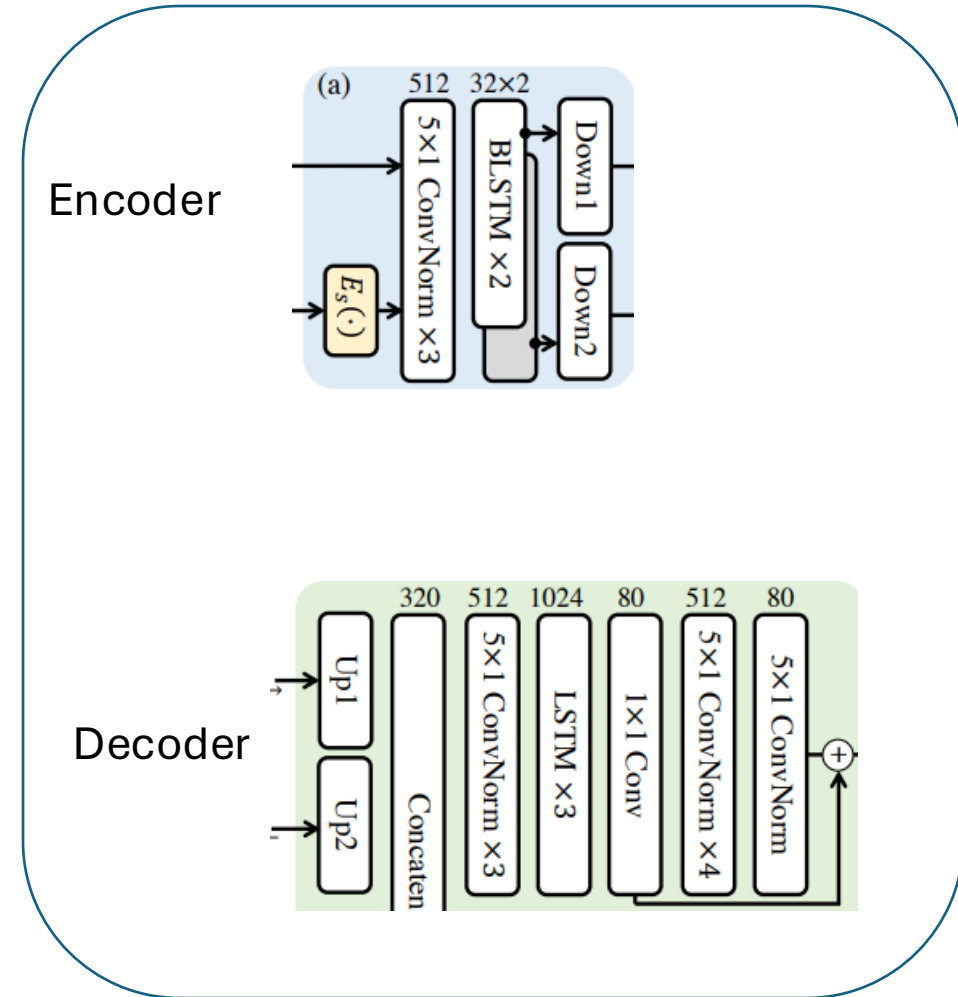
A Research Problem

- It is widely recognized that the content encoder may encode **partial** speaker identity-related information.
- Reverse Engineering the VC Model: the **disentangle-based** ones.
- Given the converted voice, how to know the voice conversion model?
- How to get the source voice to provide more powerful evidence than the classification scores?

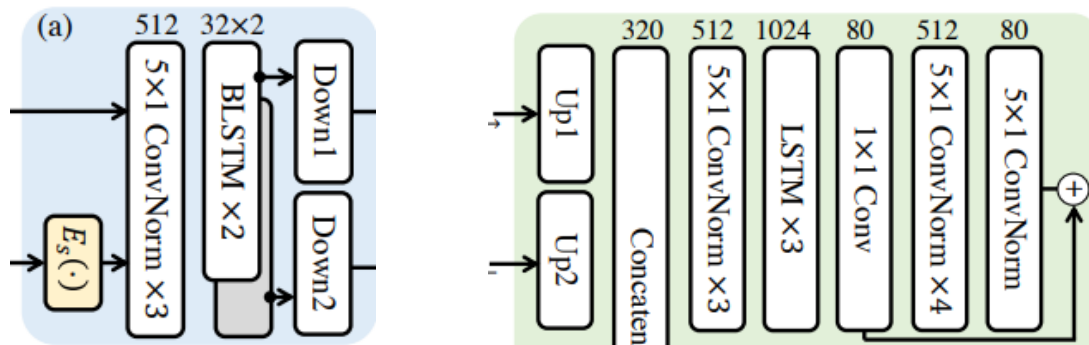
VC Model Stealing Attack



Suppose we know



VC Source Voice Recovery

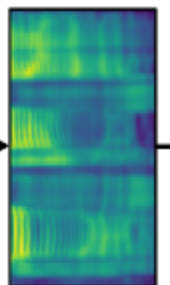


Encoder

Decoder

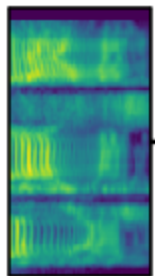
Output

$\hat{X}_{1 \rightarrow 2}$



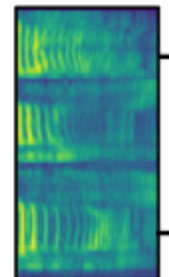
Target

X_2



Suppose we know

X_1



Can we get the Input?

$$X(\text{hat}) = \text{Decoder}\{X_2 (\text{speaker}) + X_1 (\text{content} + \text{weak speaker})\}$$

Can we get an Encoder to achieve:

$$\text{Encoder}(X(\text{hat})) = [X_2 (\text{speaker}) + X_1 (\text{weak Speaker})] + X_1 (\text{content})?$$

Suppose we directly train an Encoder to do the reverse engineering... contrastive loss?

VC Source Voice Recovery

Table 7: Voice conversion dataset.

Method	Dataset [†]	Alias	Lang.	#Speaker	#Sample
VQVC	<i>train-clean-100</i>	VQ-Train	English	251	62,750
	<i>train-clean-360</i>			921	276,300
	<i>train-other-500</i>			1,166	349,800
	VoxCeleb1			1,251	375,300
	VoxCeleb2			5,994	599,400
	<i>test-clean</i>	VQ-Test	English	40	31,200
VQVC+	<i>train-clean-100</i>	V+-Train	English	251	62,750
	<i>train-clean-360</i>			921	276,300
	<i>train-other-500</i>			1,166	349,800
	VoxCeleb1			1,251	375,300
	VoxCeleb2			5,994	599,400
	<i>test-clean</i>	V+-Test	English	40	31,200
AGAIN	<i>train-clean-100</i>	AG-Train	English	251	62,750
	<i>train-clean-360</i>			921	276,300
	<i>train-other-500</i>			1,166	349,800
	VoxCeleb1			1,251	375,300
	VoxCeleb2			5,994	599,400
	<i>test-clean</i>	AG-Test	English	40	31,200
BNE	<i>train-clean-100</i>	BN-Train	English	251	62,750
	<i>train-clean-360</i>			921	276,300
	<i>train-other-500</i>			1,166	349,800
	VoxCeleb1			1,251	375,300
	VoxCeleb2			5,994	599,400
	<i>test-clean</i>	BN-Test	English	40	31,200
BNE	MLS German	BN-GE	German	30	17,400
	MLS French	BN-FR	French	18	6,120
	MLS Spanish	BN-SP	Spanish	20	7,600

[†] *train-clean-100*, *train-clean-360* and *train-other-500* are training sets in LibriSpeech [40]. We use the test sets in German, French, and Spanish of Multilingual LibriSpeech (MLS) [44] for evaluation.

1. Get trained voice conversion models;
2. Convert large scale voice data: obtain converted voice, target voice, and source voice
3. Given converted voice and target voice, synthesize the source voice;
 1. Using GAN to get the distribution of source voice;
 2. Generated new voice to match the original voice conversion process.

Voice conversion models and datasets.

References

- One-shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization (InterSpeech'2019)
- AGAIN-VC: A one-shot voice conversion using activation guidance and adaptive instance normalization. (ICASSP'2021)
- VQVC+: one-shot voice conversion by vector quantization and U-Net architecture. (ICSA'2020)
- AVQVC: One-shot voice conversion by vector quantization with applying contrastive learning. (ICASSP'2022)
- FREEVC: Towards High-quality Text-Free **One-Shot** Voice Conversion (ICASSP'2023)