

Deep Learning for Audio

Lecture 7

Pavel Severilov

Moscow Institute of Physics and Technology

2023

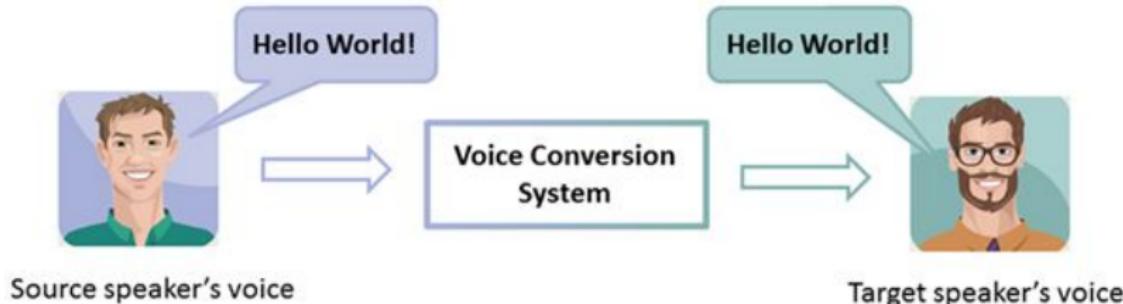
Outline

1. Voice Conversion: basics
2. AutoVC
3. CycleGAN-VC
4. StarGAN-VC

Outline

1. Voice Conversion: basics
2. AutoVC
3. CycleGAN-VC
4. StarGAN-VC

Voice Conversion: task



- ▶ Types:
 - ▶ One-to-One
 - ▶ Many-to-One
 - ▶ Many-to-Many
 - ▶ Any-to-Any (aka Zero Shot)
- ▶ Data: parallel/not parallel. Parallel data hard to obtain
- ▶ Quality: Naturalness (aka MOS), Speaker Similarity, Any Other Product Metric

VC old solution

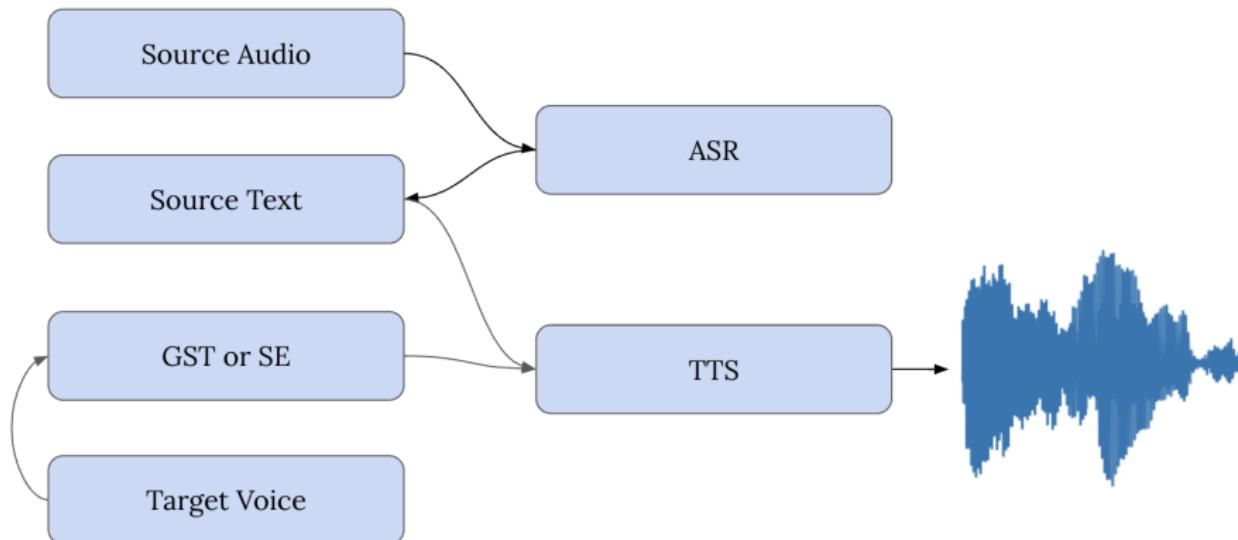
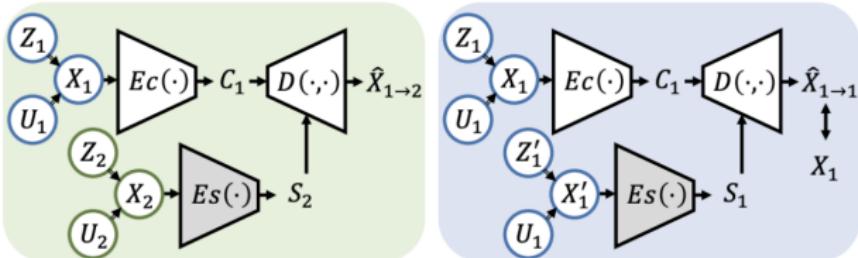


Figure: Straightforward way to solve VC task (has problems)

Outline

1. Voice Conversion: basics
2. AutoVC
3. CycleGAN-VC
4. StarGAN-VC

AutoVC



$$L_{\text{recon_post}} = \left\| \hat{X}_{1 \rightarrow 2} - X_1 \right\|_2^2$$

$$L_{\text{content}} = \left\| E_c(\hat{X}_{1 \rightarrow 1}) - C_1 \right\|_1$$

$$L_{\text{recon_pre}} = \left\| \tilde{X}_{1 \rightarrow 1} - X_1 \right\|_2^2$$

$$L = L_{\text{recon_post}} + \mu L_{\text{recon_pre}} + \lambda L_{\text{content}}$$

Figure: The style transfer autoencoder framework. The grey boxes denote pre-trained modules. During training, the content encoder and the decoder minimize the self-reconstruction error.

AutoVC

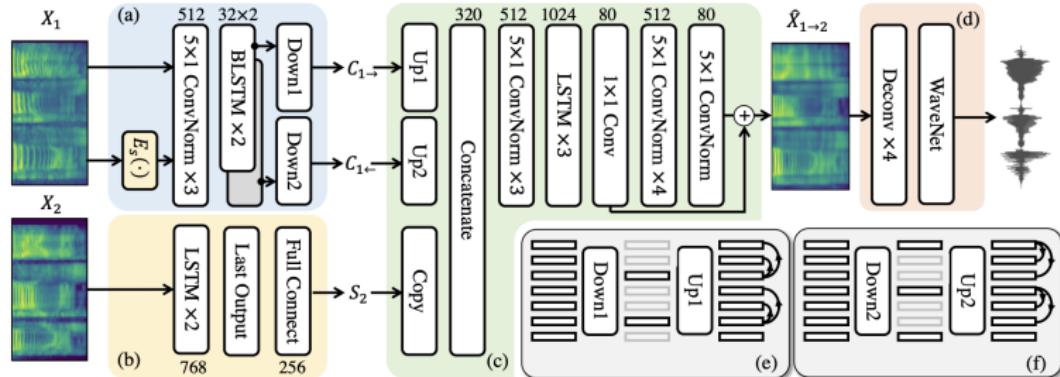


Figure: The style transfer autoencoder framework. The grey boxes denote pre-trained modules. During training, the content encoder and the decoder minimize the self-reconstruction error.

ConVoice

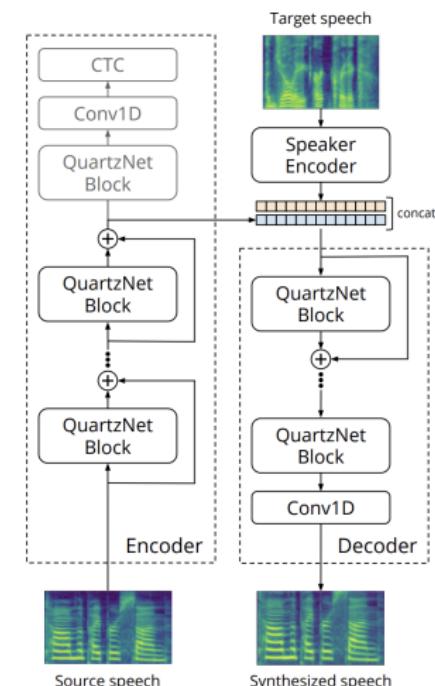


Figure: The pre-trained ASR model extracts acoustic features from the source utterance and the pre-trained speaker encoder produce the target speaker embedding from the target audio. Embeddings are concatenated and fed into the decoder that generates the melscale log spectrogram.

Outline

1. Voice Conversion: basics
2. AutoVC
3. CycleGAN-VC
4. StarGAN-VC

CycleGAN-VC

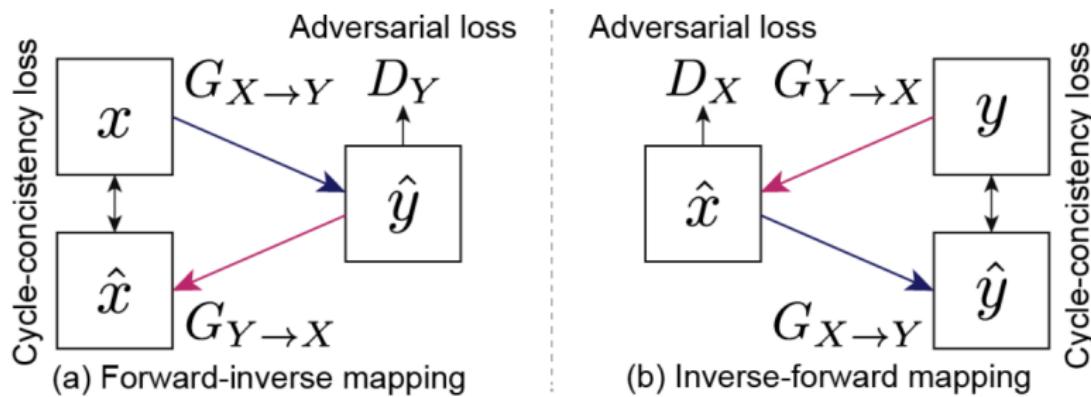


Figure: Adversarial Loss + Cycle-consistency Loss idea for VC

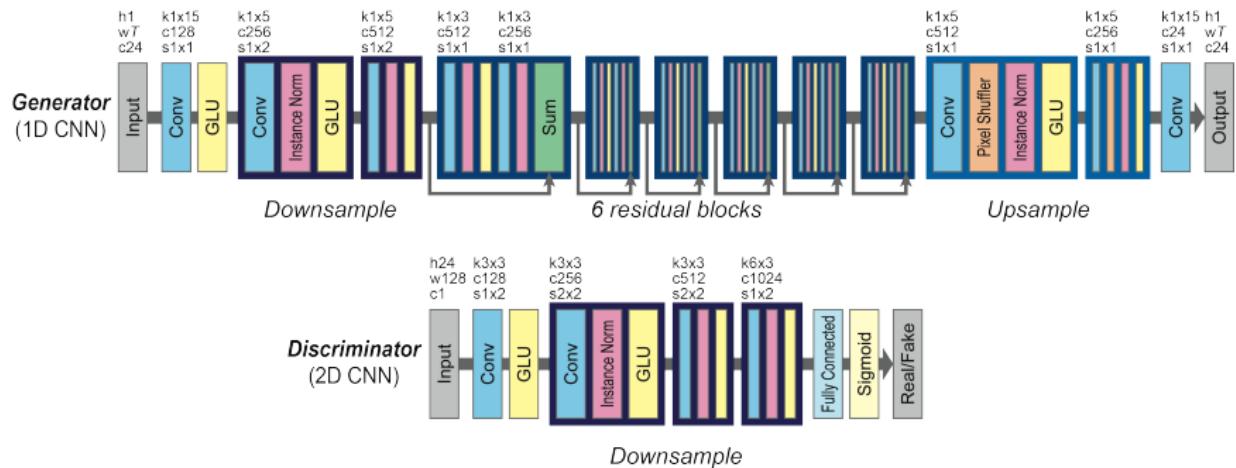
CycleGAN-VC: loss

$$\begin{aligned}\mathcal{L}_{cyc}(G_{X \rightarrow Y}, G_{Y \rightarrow X}) \\ = & \mathbb{E}_{x \sim P_{\text{Data}}(x)} [\|G_{Y \rightarrow X}(G_{X \rightarrow Y}(x)) - x\|_1] \\ & + \mathbb{E}_{y \sim P_{\text{Data}}(y)} [\|G_{X \rightarrow Y}(G_{Y \rightarrow X}(y)) - y\|_1]\end{aligned}$$

$$\begin{aligned}\mathcal{L}_{id}(G_{X \rightarrow Y}, G_{Y \rightarrow X}) = & \mathbb{E}_{y \sim P_{\text{Data}}(y)} [\|G_{X \rightarrow Y}(y) - y\|_1] \\ & + \mathbb{E}_{x \sim P_{\text{Data}}(x)} [\|G_{Y \rightarrow X}(x) - x\|_1]\end{aligned}$$

Figure: Adversarial Loss + Cycle-consistency Loss + Identity Loss

CycleGAN-VC: architecture



CycleGAN-VC2

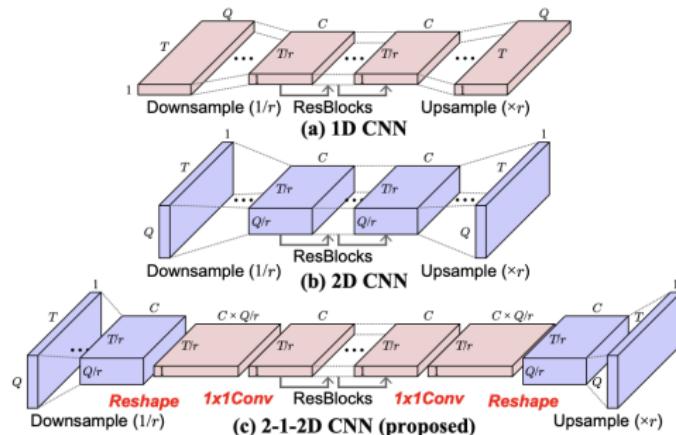


Fig. 2. Comparison of generator network architectures. Red and blue blocks indicate 1D and 2D convolution layers, respectively. r indicates a downsampling or upsampling rate.

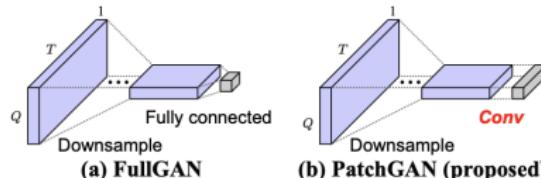


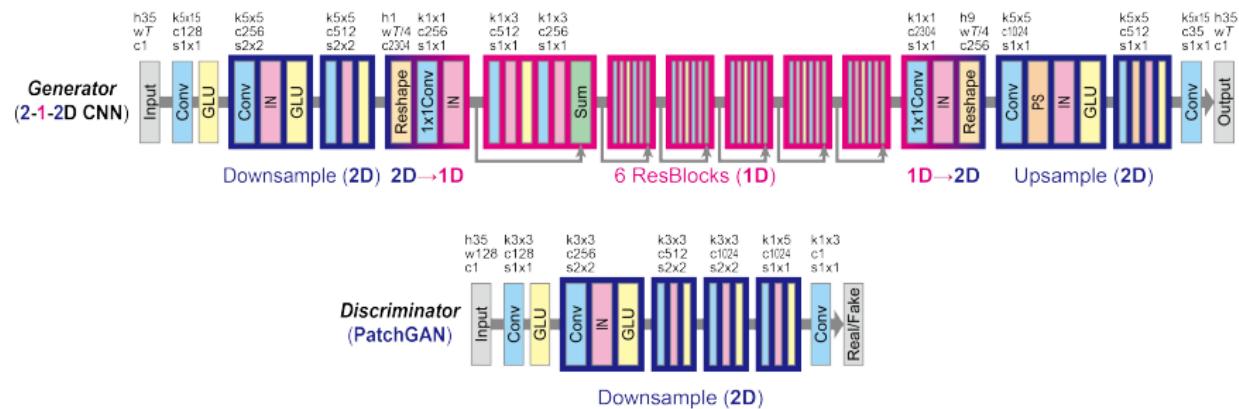
Fig. 3. Comparison of discriminator network architectures

CycleGAN-VC2

$$\begin{aligned}\mathcal{L}_{adv2}(G_{X \rightarrow Y}, G_{Y \rightarrow X}, D'_X) &= \mathbb{E}_{x \sim P_X(x)} [\log D'_X(x)] \\ &+ \mathbb{E}_{x \sim P_X(x)} [\log(1 - D'_X(G_{Y \rightarrow X}(G_{X \rightarrow Y}(x))))]\end{aligned}$$

Figure: Additional adv loss: Two-step Adversarial Loss

CycleGAN-VC2: architecture



Outline

1. Voice Conversion: basics
2. AutoVC
3. CycleGAN-VC
4. StarGAN-VC

StarGAN-VC

$$\begin{aligned}\mathcal{L}_{\text{adv}}^D(D) &= -\mathbb{E}_{c \sim p(c), \mathbf{y} \sim p(\mathbf{y}|c)} [\log D(\mathbf{y}, c)] \\ &\quad - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), c \sim p(c)} [\log(1 - D(G(\mathbf{x}, c), c))]\end{aligned}$$
$$\mathcal{L}_{\text{adv}}^G(G) = -\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), c \sim p(c)} [\log D(G(\mathbf{x}, c), c)]$$

$$\begin{aligned}\mathcal{L}_{\text{cls}}^C(C) &= -\mathbb{E}_{c \sim p(c), \mathbf{y} \sim p(\mathbf{y}|c)} [\log p_C(c \mid \mathbf{y})]\end{aligned}$$
$$\mathcal{L}_{\text{cls}}^G(G) = -\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), c \sim p(c)} [\log p_C(c \mid G(\mathbf{x}, c))]$$

- ▶ Based on CycleGAN-VC
- ▶ Many-to-many
- ▶ Domain Classification Loss

CycleGAN-VC vs StarGAN-VC

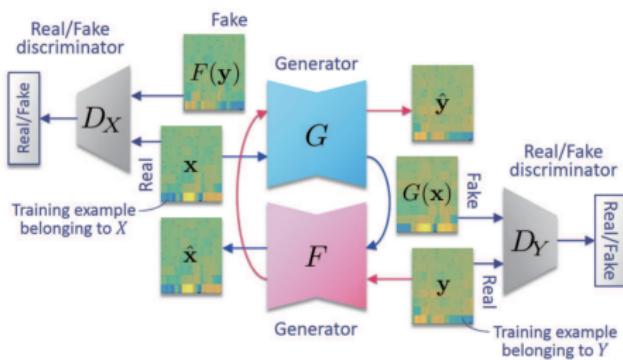


Fig. 1. Concept of CycleGAN training.

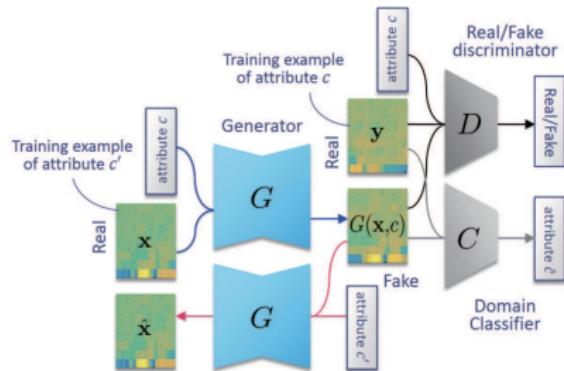
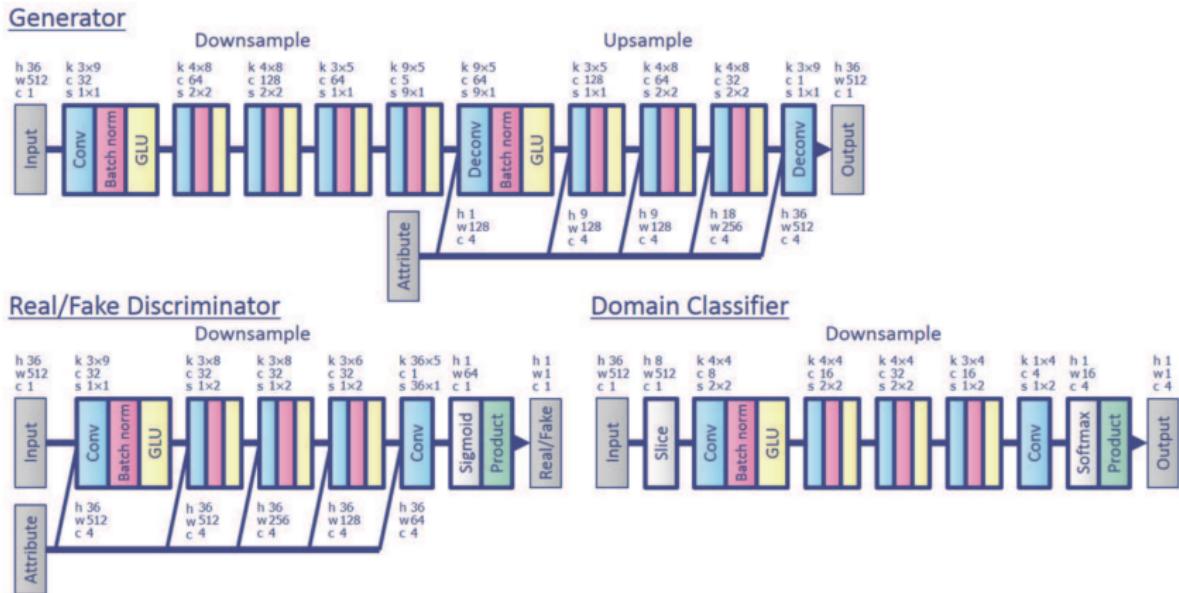


Fig. 2. Concept of StarGAN training.

StarGAN-VC: architecture



StarGAN-VC2: ST Adversarial Loss

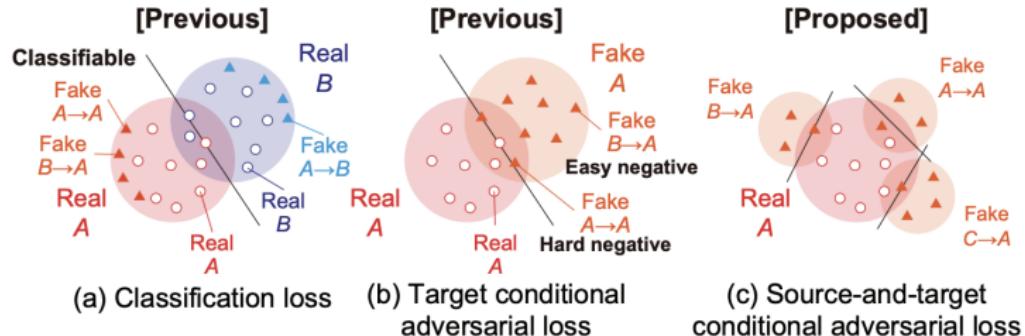


Figure: Comparison of conditional methods in training objectives. “A” and “B” denote the domain codes

$$\begin{aligned}\mathcal{L}_{st\text{-}adv} = & \mathbb{E}_{(\mathbf{x}, c) \sim P(\mathbf{x}, c), c' \sim P(c')} [\log D(\mathbf{x}, c', c)] \\ & + \mathbb{E}_{(\mathbf{x}, c) \sim P(\mathbf{x}, c), c' \sim P(c')} [\log D(G(\mathbf{x}, c, c'), c, c')]\end{aligned}$$

StarGAN-VC2: Conditional IN

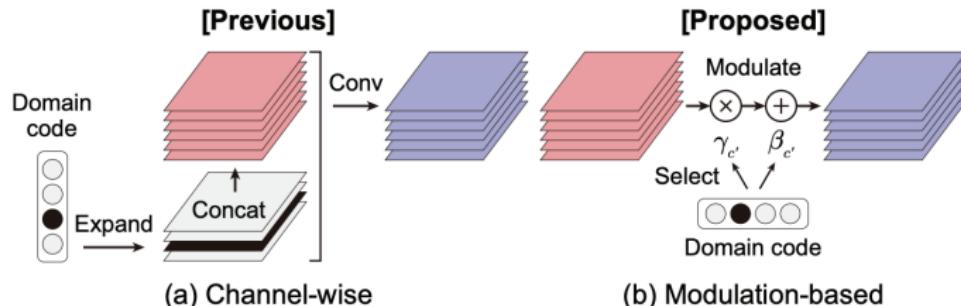


Figure: Comparison of conditional methods in generator networks

$$\text{CIN}(\mathbf{f}; c') = \gamma_{c'} \left(\frac{\mathbf{f} - \mu(\mathbf{f})}{\sigma(\mathbf{f})} \right) + \beta_{c'}$$

StarGAN-VC2:architecture

