Neural Voice Conversion and it's recent advancements

Dipjyoti Paul

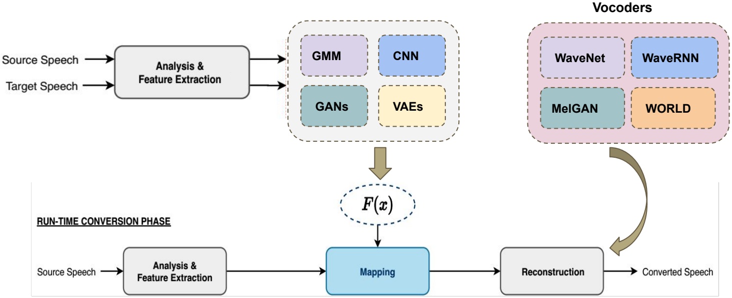
University of Crete, Computer Science Dept., Greece,

HY578: Digital Speech Signal Processing
Nov 2025

1. Introduction to VC
2. Neural Voice Conversion
   - Sequence based VC
   - VAE VC
   - GAN VC
   - zero-shot/few-shot VC
3. TTS to VC
4. References
5. THANK YOU

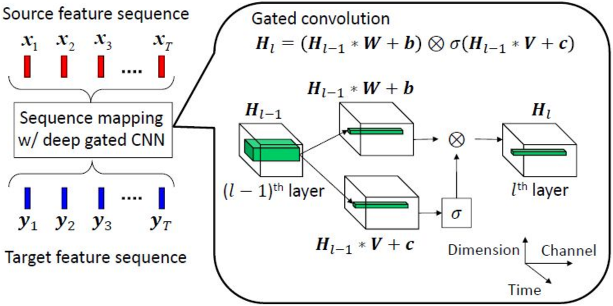
Voice Conversion

- Voice conversion framework.



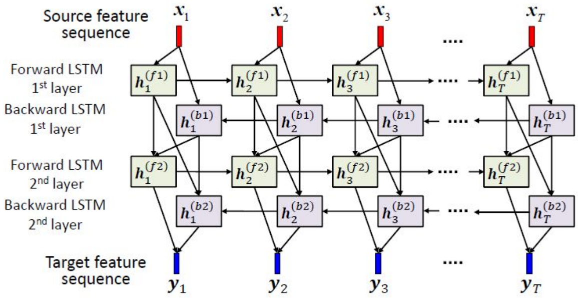
Sequence based VC

- The proposed method uses Gated CNN to model long-term dependencies [Kaneko et. al. 2017].
- Training with fixed length of sequence.
- Inference with arbitrary length of sequence.



Sequence based VC

- This method uses Bi-Directional LSTM which considers inter-frame correlation [Sun et. al. 2015].



Variational Autoencoder (VAE)-VC

- The core of VAE-VC is an encoder-decoder network.
- During training, given an observed (source or target) spectral frame $\mathbf{x}$, a speaker-independent encoder E_θ with parameter set θ encodes $\mathbf{x}$ into a latent code:

$$\hat{\mathbf{z}} = E_\theta(\mathbf{x})$$

- The speaker code $\hat{\mathbf{y}}$ of the input frame is then concatenated with the latent code, and passed to a conditional decoder G_ϕ with parameter set ϕ to reconstruct the input.

$$\hat{\mathbf{x}} = G_\phi(\hat{\mathbf{z}}, \hat{\mathbf{y}}) = G_\phi(E_\theta(\mathbf{x}), \hat{\mathbf{y}})$$

Variational Autoencoder (VAE)-VC

- The core of VAE-VC is an encoder-decoder network.
- During training, given an observed (source or target) spectral frame $\mathbf{x}$, a speaker-independent encoder E_θ with parameter set θ encodes $\mathbf{x}$ into a latent code:

$$\hat{\mathbf{z}} = E_\theta(\mathbf{x})$$

- The speaker code $\hat{\mathbf{y}}$ of the input frame is then concatenated with the latent code, and passed to a conditional decoder G_ϕ with parameter set ϕ to reconstruct the input.

$$\hat{\mathbf{x}} = G_\phi(\hat{\mathbf{z}}, \hat{\mathbf{y}}) = G_\phi(E_\theta(\mathbf{x}), \hat{\mathbf{y}})$$

Variational Autoencoder (VAE)-VC

- The core of VAE-VC is an encoder-decoder network.
- During training, given an observed (source or target) spectral frame $\mathbf{x}$, a speaker-independent encoder E_θ with parameter set θ encodes $\mathbf{x}$ into a latent code:

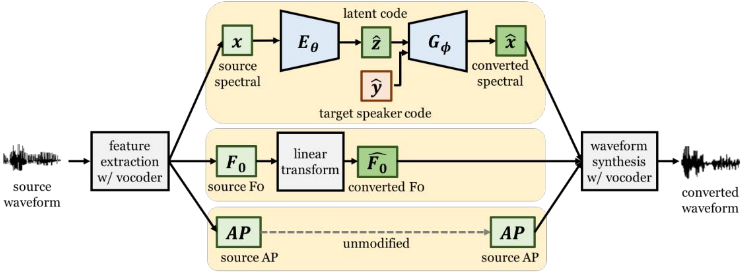
$$\hat{\mathbf{z}} = E_\theta(\mathbf{x})$$

- The speaker code $\hat{\mathbf{y}}$ of the input frame is then concatenated with the latent code, and passed to a conditional decoder G_ϕ with parameter set ϕ to reconstruct the input.

$$\hat{\mathbf{x}} = G_\phi(\hat{\mathbf{z}}, \hat{\mathbf{y}}) = G_\phi(E_\theta(\mathbf{x}), \hat{\mathbf{y}})$$

VAE-VC

- This framework is based on variational auto-encoder which exploits non-parallel corpora. [Hsu et. al. 2016]



VAE-VC

- The final approximated objective function of an individual frame:

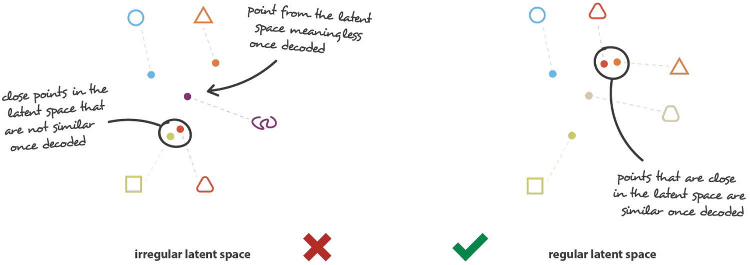
$$\mathcal{L}(\theta, \phi; \mathbf{x}_n) = -D_{KL}(q_\phi(\mathbf{z}_n|\mathbf{x}_n)||p(\mathbf{z}_n)) + \mathbb{E}_{q_\phi(\mathbf{z}_n|\mathbf{x}_n)}[log p_\theta(\mathbf{x}_n|\mathbf{z}_n, \hat{\mathbf{y}}_n)]$$

  where, $q_\phi(.)$ is the variational posterior.
  $p(.)$ is the true posterior.
  $D_{KL}(.||.)$ is the Kullback-Leibler divergence (KLD) of the approximate from the true posterior.

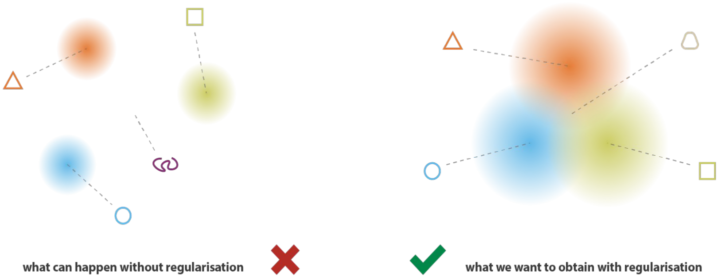
- Training is equivalent to iteratively finding the parameters that maximize the variational lower bound:

$$\{\theta^*, \phi^*\} = argmax_{\theta,\phi} \mathcal{L}(\theta, \phi; \mathbf{X})$$

Intuitions about Regularization

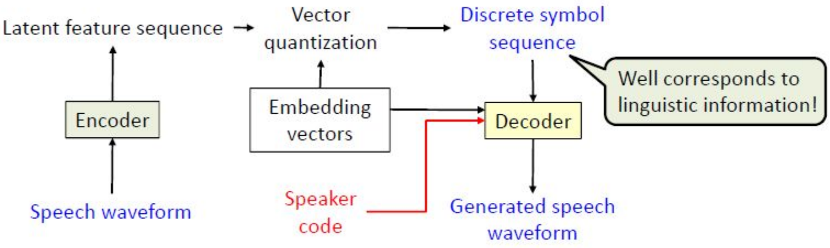


Intuitions about Regularization



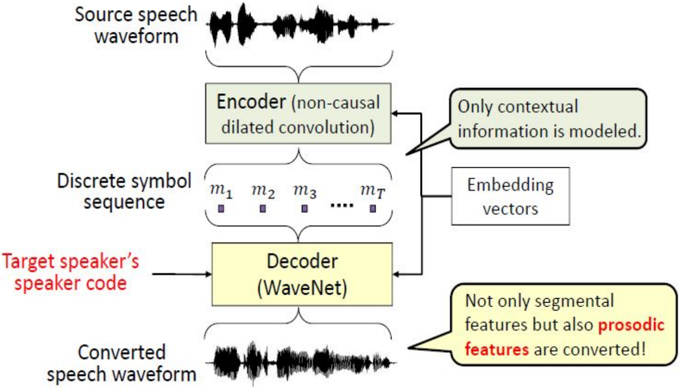
Vector Quantization VAE (VQ-VAE) VC

- Directly encodes speech waveform into a discrete symbol sequence capturing long-term dependencies. [Van den Oord et. al. 2017]
- The posterior and prior distributions are categorical
- The samples drawn from these distributions index an embedding table. These embeddings are then used as input into the decoder network.



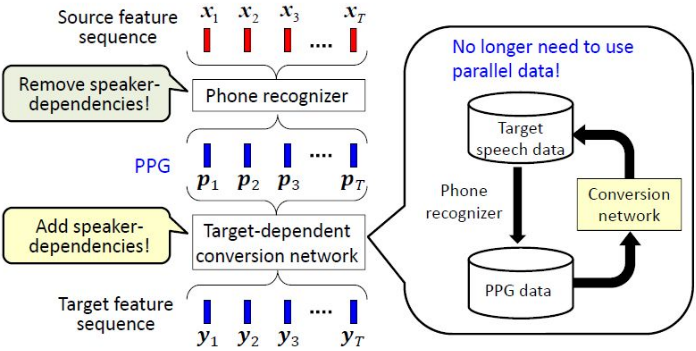
VQ-VAE VC

- Extract discrete symbol sequences as speaker independent contextual features.



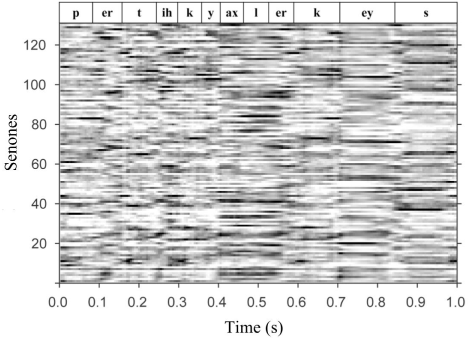
Phoneme Posteriogram VC

- Extract phoneme posteriorgram (PPG) as speaker independent contextual features. [Sun et. al. 2016]



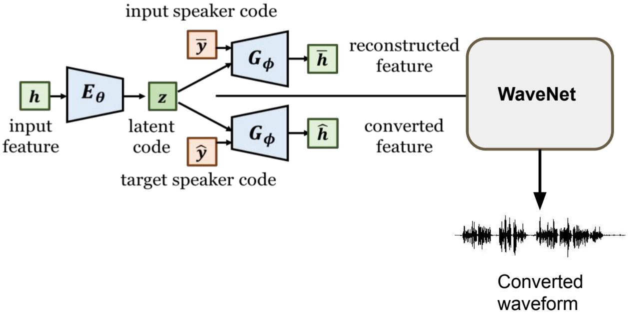
Phoneme Posteriogram VC

- PPG representation of the spoken phrase "particular case".
- The horizontal axis (time in seconds), the vertical (indices of phonetic classes). The number of senones is 131. Darker shade implies a higher posterior probability.



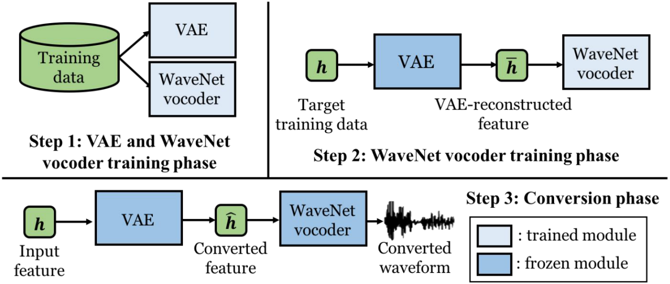
WaveNet Vocoder in VAE-VC

- A general framework of WaveNet vocoder in voice conversion. [Huang et. al. 2019]



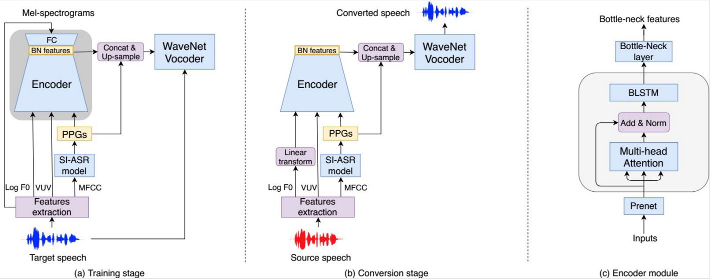
WaveNet Vocoder in VAE-VC

- Training and Inference



Jointly Trained Conversion Model and Vocoder

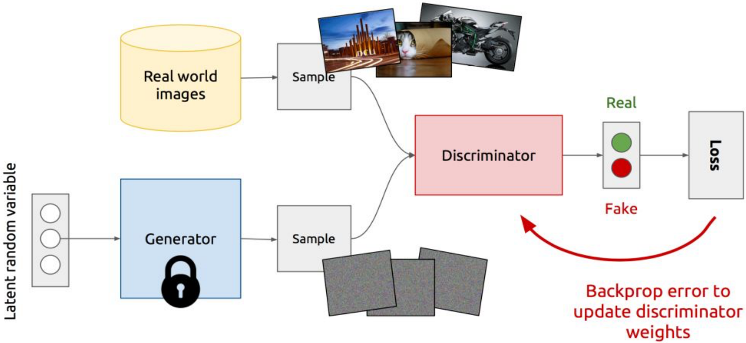
- The proposed model jointly trains a conversion model that maps phonetic posteriorgrams (PPGs) to Mel-spectrograms and a WaveNet vocoder. [Liu et. al. 2019]



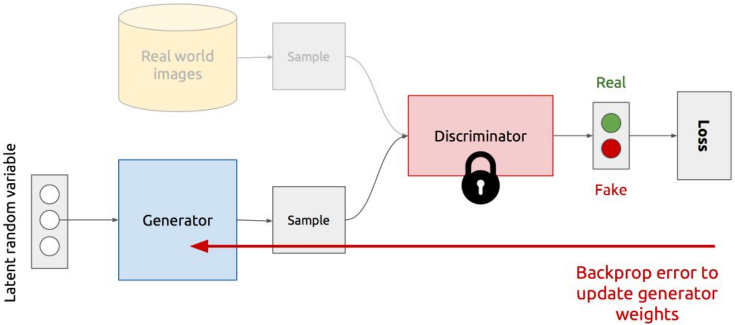
GAN Formulation

- An adversarial process which simultaneously trains two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. [Goodfellow et. al. 2017]

Discriminator (D) Training



Generator (G) Training



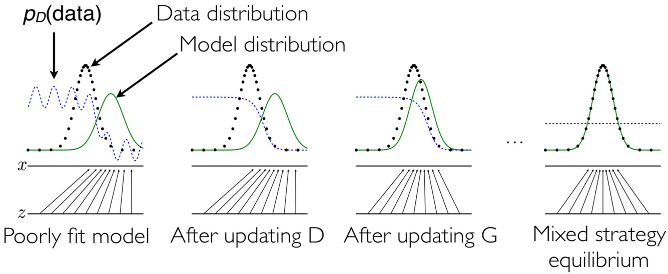
Mathematical Notations

Value of — Expectation — prob. of D(real) — prob. of D(fake)

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

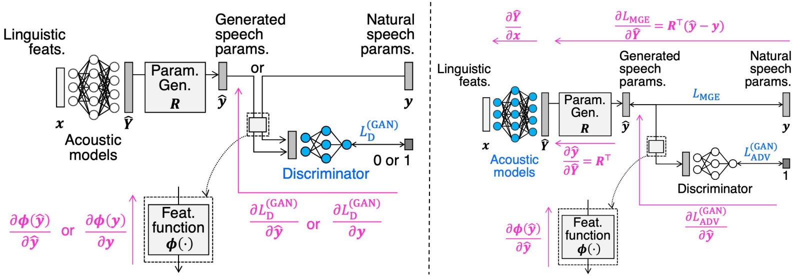
Minimize G — Maximize D — x is sampled from real data — z is sampled from N(0, I) — fake

Learning GANs



GAN-BASED VC

- Training and gradients for updating the discriminator and Generator. ADV is adversarial loss and MGE refers to minimum genration loss. [Saito et. al. 2018]

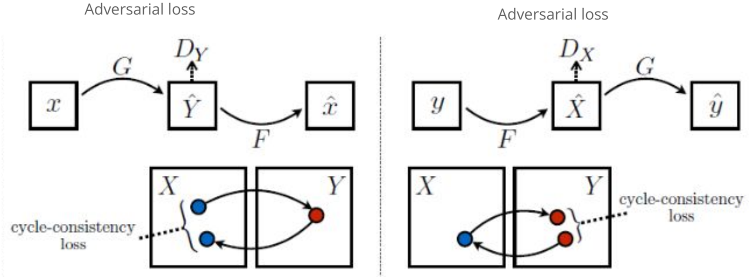


CycleGAN Voice Conversion

- A non-parallel voice-conversion (VC) method that can learn a mapping from source to target speech without relying on parallel data. [Kaneko et. al. 2018]
- Forward and inverse mappings are simultaneously learned using an adversarial loss and cycle-consistency loss.
- Important losses are introduced:
  - Adversarial loss
  - Cycle-consistency loss
  - Identity-mapping loss

CycleGAN losses

- The model learns mapping from source x to target y and vice verse.



CycleGAN losses

- Two mapping function (Adversarial loss): G and F.

$$G : X \rightarrow Y \text{ and } F : Y \rightarrow X$$

- Cycle-consistency loss:
  - Forward: $x \rightarrow G(x) \rightarrow F(G(x)) \rightarrow \hat{x}$
  - Backward: $y \rightarrow F(y) \rightarrow G(F(x)) \rightarrow \hat{y}$
- Adversarial loss + Cycle-consistency loss

$$\mathcal{L}_{adv}(G_{X\rightarrow Y}, D_Y) + \mathcal{L}_{adv}(G_{Y\rightarrow X}, D_X) + \lambda_{cycle}\mathcal{L}_{cycle}(G_{X\rightarrow Y}, G_{Y\rightarrow X})$$

CycleGAN losses

- Adversarial loss:

$$\mathcal{L}_{adv}(G_{X\to Y}, D_Y) = \mathbb{E}_{y\sim P_{Data(y)}}[\log D_Y(y)]$$

$$+ \mathrm{E}_{x\sim P_{Data(x)}}[\log(1 - D_Y(G_{X\to Y}(x)))]$$

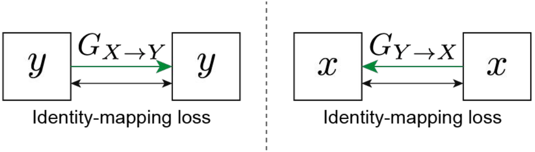
- Cycle-consistency loss:

$$\mathcal{L}_{cycle}(G_{X\to Y}, G_{Y\to X}) = \mathbb{E}_{x\sim P_{Data(x)}}(\| G_{Y\to X}(G_{X\to Y}(x)) - x\|_1)$$

$$+ \mathrm{E}_{y\sim P_{Data(y)}}(\| G_{X\to Y}(G_{Y\to X}(y)) - y\|_1)$$

CycleGAN losses

- Identity-mapping loss:
  To encourage linguistic-information preservation, an identity-mapping loss is implemented.
- It encourages the generator to find the mapping that preserves composition between the input and output.

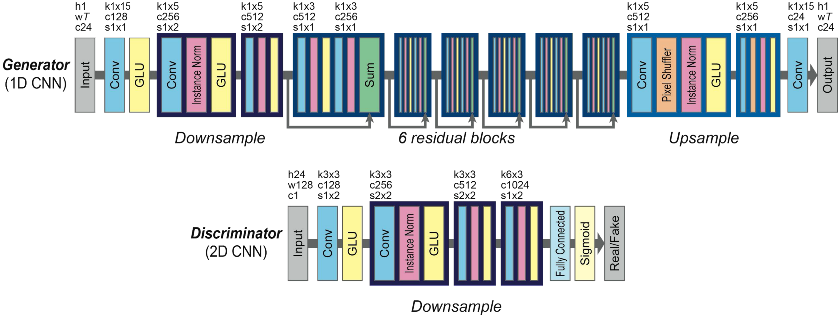


$$\mathcal{L}_{id}(G_{X\to Y}, G_{Y\to X}) = \mathbb{E}_{y\sim P_{Data(y)}}(\| G_{X\to Y}(y) - y\|_1)$$

$$+ \mathrm{E}_{x\sim P_{Data(x)}}(\| G_{Y\to X}(x) - x\|_1)$$

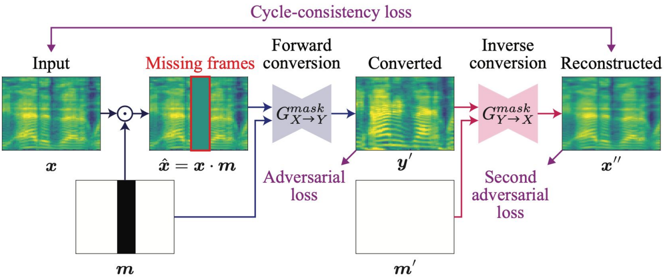
CycleGAN Architecture

- The model architecture.



Masked CycleGAN VC

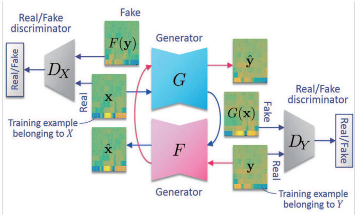
- It is trained using a novel auxiliary task called filling in frames.
- A temporal mask to the input mel-spectrogram and encourage the converter to fill in missing frames.
- Allows the converter to learn time-frequency structures in a self-supervised manner. [Kaneko et. al. 2021]



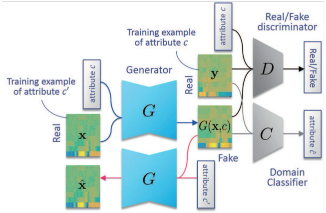
StarGAN VC

- A non-parallel many-to-many voice conversion.
- Generator (G) takes an acoustic feature with an attribute c as the inputs and generates an acoustic feature sequence $\hat{y} = G(x, c)$.
- Discriminator (D) is designed to produce a probability $D(y, c)$ that an input y is a real speech feature. [Kameoka et. al. 2018]

CycleGAN



StarGAN



StarGAN Losses

- The full objectives of StarGAN-VC to be minimized with respect to G, D and C are
  - Generator loss:

$$\mathcal{L}_{adv}(G) + \lambda_{cls}\mathcal{L}_{cls}(G) + \lambda_{cyc}\mathcal{L}_{cyc}(G) + \lambda_{id}\mathcal{L}_{id}(G)$$

  - Discriminator loss

$$\mathcal{L}_{adv}(D)$$

  - Classifier loss

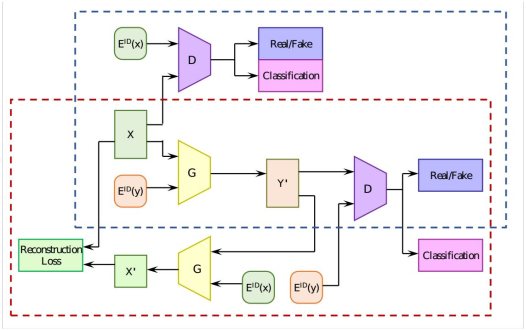
$$\mathcal{L}_{cls}(C)$$

ZERO-SHOT/FEW-SHOT VC

- The target speaker is unseen (zero-shot) during training or a very limited set of samples are available (few-shot).
- An universal embedding vector is used to represent speaker id.
- The idea is to represent any arbitrary unseen speaker ID with an embedding vector.
- Such embedding vector represents unseen speaker's timbre and would be a weighted combination of the timbres the speakers seen in the dataset. [Wang et. al. 2020]

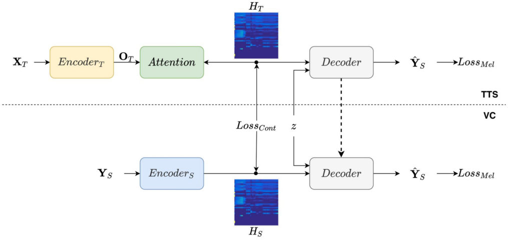
ZERO-SHOT STARGAN VC

- Illustration of our proposed StarGAN framework.
  - $E_{ID}(.)$ represents the embedding ID of a speaker.
  - X' and Y' refer to the features reconstructed through G with embedded ID $E_{ID}(X)$ and $E_{ID}(Y)$ respectively.



Text-to-Speech Synthesis to Voice Conversion

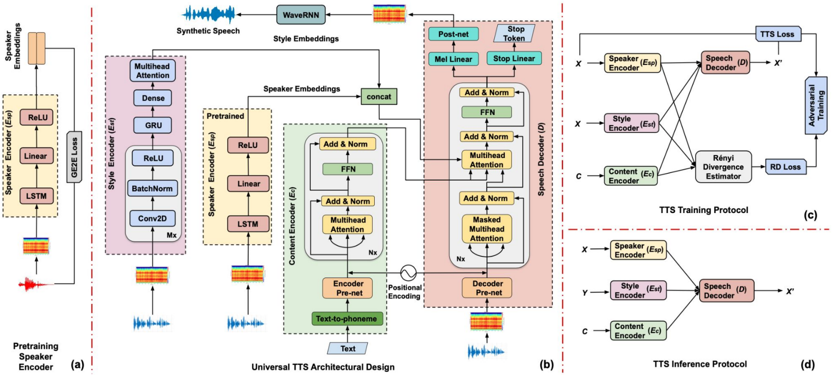
- VC framework by learning from a TTS synthesis system.
- The decoder is condition on a speaker embedding, becoming any-to-any VC.
- X_T denotes the input text, Y_S and $\hat{Y}_S$ are target melspecs and the melspecs generated by the pipelines; O_T denotes the text encoding, H_T denotes the context vectors from TTS pipeline, H_S denotes the context vectors equivalents from the VC pipeline. [Zhang et. al. 2021]



Multi-Speaker Multi-Style TTS

- To generalize the models with multiple speakers and multiple styles using just the reconstruction loss, performance unfortunately deteriorates.
- Leaking content information into the style embeddings ("content leakage") and leaking speaker information into style embeddings ("style leakage").
- Minimizes the Reyni Diverence Disentagle Representation (RDDR) between the joint distribution and the product of marginals for the content-style and style-speaker pairs.
- Training is conducted on non-parallel data and generates voices in an unsupervised manner, i.e., neither style annotation nor speaker label are required. [Paul et. al. 2021]

Multi-Speaker Multi-Style TTS



Multi-Speaker Multi-Style TTS

Algorithm: Pseudo-code for proposed RDDR training

Input: Speech and text pairs $\langle \mathbf{x}_i, \mathbf{c}_i \rangle$.
Pre-training: Optimize E_c, D on LJSpeech using
$\min_{E_c, E_{st}, D} \sum_i \| D(E_c(\mathbf{c}_i), E_{st}(\mathbf{x}_i), E_{sp}(\mathbf{x}_i)) - \mathbf{x}_i \|_1$
$E_{sp} \leftarrow$ GE2E training
$E_{st}, T_\theta, T'_{\theta'} \leftarrow$ initialization with random weights
while $E_{st}, D, T_\theta, T'_{\theta'}$ *not converged* **do**

Sample mini-batch from $\langle \mathbf{x}_i, \mathbf{c}_i \rangle$; i = {1, 2, . . . , b}
$\{\mathbf{p}_i\} \leftarrow \{E_c(\mathbf{c}_i) | i = 1, 2, \ldots, b\}$
$\{\mathbf{q}_i\} \leftarrow \{E_{st}(\mathbf{x}_i) | i = 1, 2, \ldots, b\}$
$\{\mathbf{r}_i\} \leftarrow \{E_{sp}(\mathbf{x}_i) | i = 1, 2, \ldots, b\}$
$\{\hat{\mathbf{p}}_i\}, \{\tilde{\mathbf{r}}_i\} \leftarrow$ random permutation of $\{\mathbf{p}_i\}$, $\{\mathbf{r}_i\}$
$\mathcal{L}_{RD^1} = \sum_k [-\frac{1}{\beta_k} \log \frac{1}{b} \sum_{i=1}^{b} e^{-\beta_k T_\theta(\mathbf{p}_i, \mathbf{q}_i)}$
$- \frac{1}{\gamma_k} \log \frac{1}{b} \sum_{i=1}^{b} e^{\gamma_k T_\theta(\hat{\mathbf{p}}_i, \mathbf{q}_i)}]$
$\mathcal{L}_{RD^2} = \sum_k [-\frac{1}{\beta_k} \log \frac{1}{b} \sum_{i=1}^{b} e^{-\beta_k T'_{\theta'}(\mathbf{r}_i, \mathbf{q}_i)}$
$- \frac{1}{\gamma_k} \log \frac{1}{b} \sum_{i=1}^{b} e^{\gamma_k T'_{\theta'}(\tilde{\mathbf{r}}_i, \mathbf{q}_i)}]$
The overall objective function:
$\mathcal{L} = \frac{1}{b} \sum_{i=1}^{b} \| D(\mathbf{p}_i, \mathbf{q}_i, \mathbf{r}_i) - \mathbf{x}_i \|_1$
$+ \lambda \max(0, \mathcal{L}_{RD^1}) + \lambda \max(0, \mathcal{L}_{RD^2})$
$D = D - \epsilon \nabla_D \mathcal{L}; \quad E_{st} = E_{st} - \epsilon \nabla_{E_{st}} \mathcal{L}$
$T_\theta = T_\theta + \epsilon \nabla_D \mathcal{L}_{RD^1}; \quad T'_{\theta'} = T'_{\theta'} + \epsilon \nabla_D \mathcal{L}_{RD^2}$

end

References

1. Kaneko, Takuhiro, Hirokazu Kameoka, Kaoru Hiramatsu, and Kunio Kashino. "Sequence-to-Sequence Voice Conversion with Similarity Metric Learned Using Generative Adversarial Networks." In Interspeech, vol. 2017, pp. 1283-1287. 2017.
2. Sun, Lifa, Shiyin Kang, Kun Li, and Helen Meng. "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks." In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 4869-4873. IEEE, 2015.
3. Hsu, Chin-Cheng, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. "Voice conversion from non-parallel corpora using variational auto-encoder." In 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp. 1-6. IEEE, 2016.
4. Van den Oord, Aaron, Oriol Vinyals, and Koray Kavukcuoglu. "Neural discrete representation learning." In Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6309-6318. 2017.
5. Sun, Lifa, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng. "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training." In 2016 IEEE International Conference on Multimedia and Expo (ICME), pp. 1-6. IEEE, 2016.
6. Huang, Wen-Chin, Yi-Chiao Wu, Hsin-Te Hwang, Patrick Lumban Tobing, Tomoki Hayashi, Kazuhiro Kobayashi, Tomoki Toda, Yu Tsao, and Hsin-Min Wang. "Refined wavenet vocoder for variational autoencoder based voice conversion." In 2019 27th European Signal Processing Conference (EUSIPCO), pp. 1-5. IEEE, 2019.
7. Liu, Songxiang, Yuewen Cao, Xixin Wu, Lifa Sun, Xunying Liu, and Helen Meng. "Jointly Trained Conversion Model and WaveNet Vocoder for Non-Parallel Voice Conversion Using Mel-Spectrograms and Phonetic Posteriorgrams." In INTERSPEECH, pp. 714-718. 2019.
8. Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks." Communications of the ACM 63, no. 11 (2020): 139-144.

References

9. Saito, Yuki, Shinnosuke Takamichi, and Hiroshi Saruwatari. "Statistical parametric speech synthesis incorporating generative adversarial networks." IEEE/ACM Transactions on Audio, Speech, and Language Processing 26, no. 1 (2017): 84-96.
10. Kaneko, Takuhiro, and Hirokazu Kameoka. "Parallel-data-free voice conversion using cycle-consistent adversarial networks." arXiv preprint arXiv:1711.11293 (2017).
11. Kaneko, Takuhiro, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. "Maskcyclegan-vc: Learning non-parallel voice conversion with filling in frames." In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5919-5923. IEEE, 2021.
12. Kameoka, Hirokazu, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo. "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks." In 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 266-273. IEEE, 2018.
13. Wang, Ruobai, Yu Ding, Lincheng Li, and Changjie Fan. "One-shot voice conversion using star-GAN." In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7729-7733. IEEE, 2020.
14. Zhang, Mingyang, Yi Zhou, Li Zhao, and Haizhou Li. "Transfer learning from speech synthesis to voice conversion with non-parallel training data." IEEE/ACM Transactions on Audio, Speech, and Language Processing 29 (2021): 1290-1302.
15. Paul, Dipjyoti, Sankar Mukherjee, Yannis Pantazis, and Yannis Stylianou. "A Universal Multi-Speaker Multi-Style Text-to-Speech via Disentangled Representation Learning Based on Rényi Divergence Minimization." In Interspeech, pp. 3625-3629. 2021.
16. Tomoki Toda: Advanced Voice Conversion, Speech Processing Courses in Crete (SPCC2018).

THANK YOU
for your attention