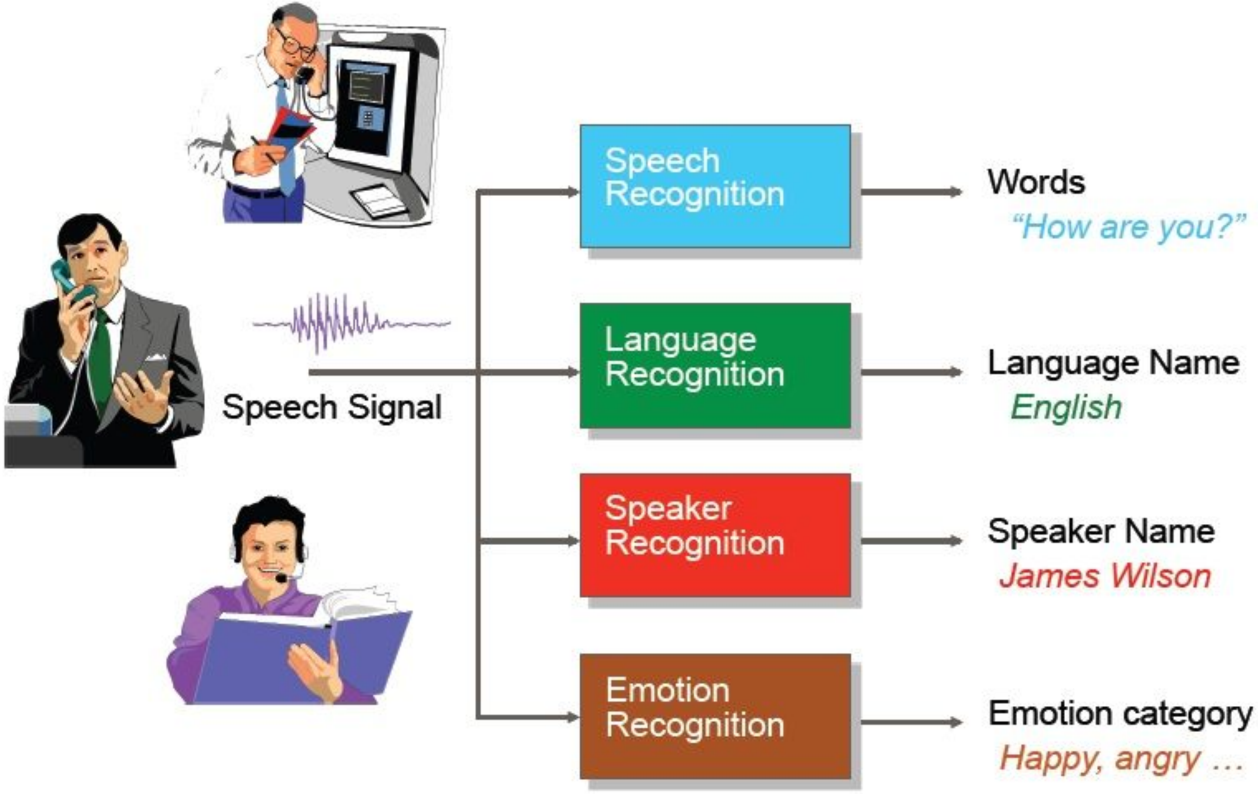
Neural Voice Conversion

Dipjyoti Paul
University of Crete, Greece

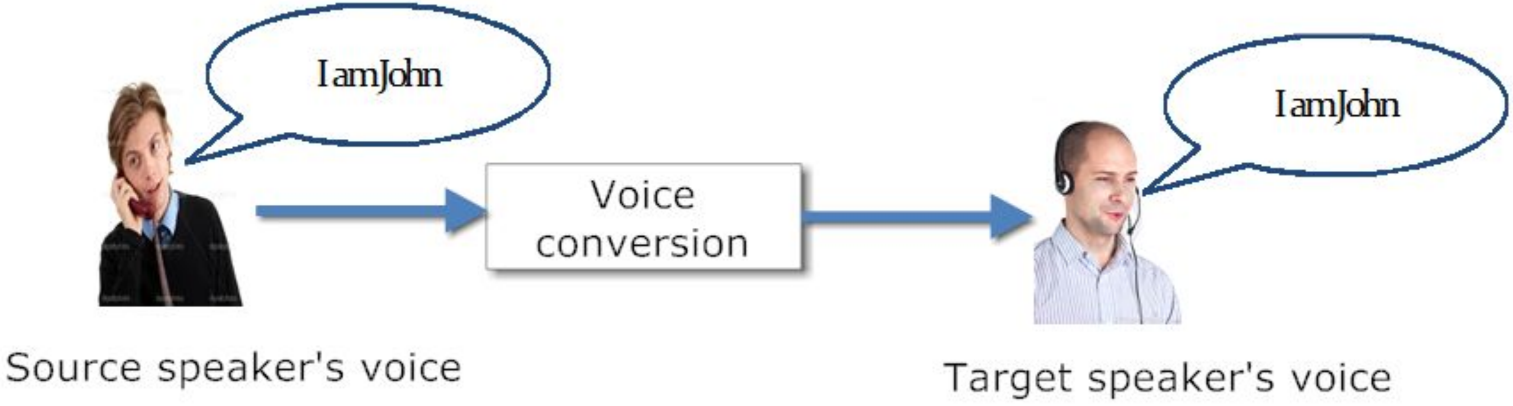
HY578: Digital Speech Signal Processing
22 November 2023

Information in Speech

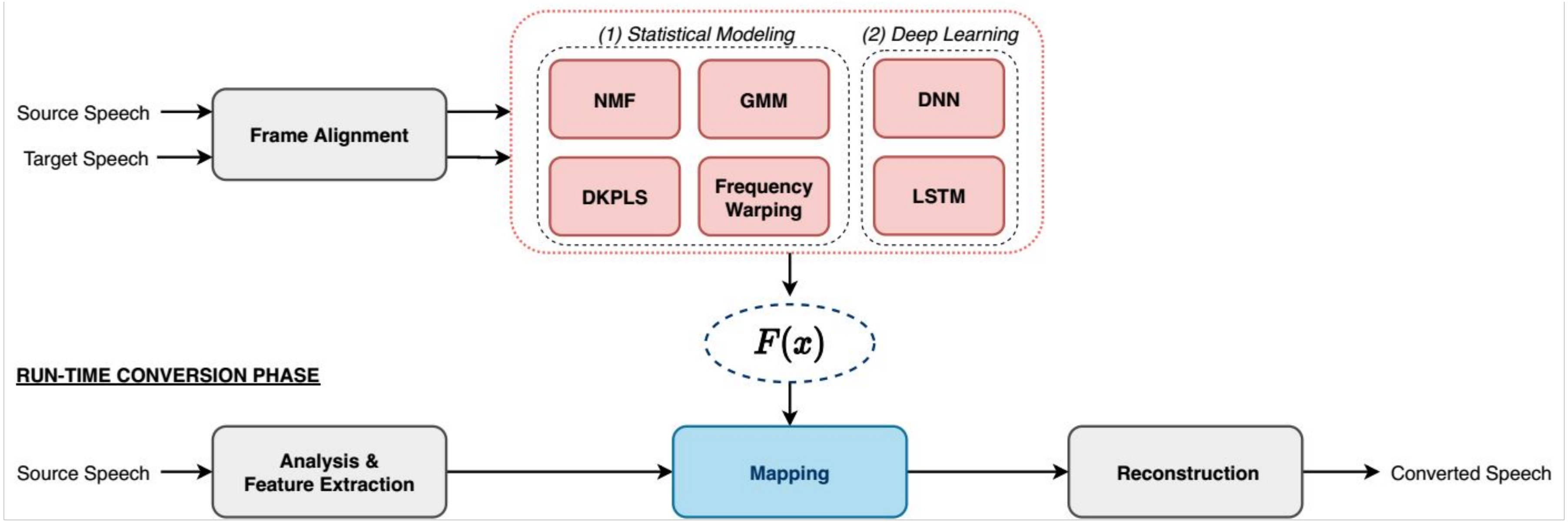


Voice Conversion (VC)

- Technique to convert the utterance of a source speaker to create the perception as if spoken by a specified target speaker.
- Only transform the speaker timbre (para-linguistic information) and keep the linguistic message in the utterance unchanged.



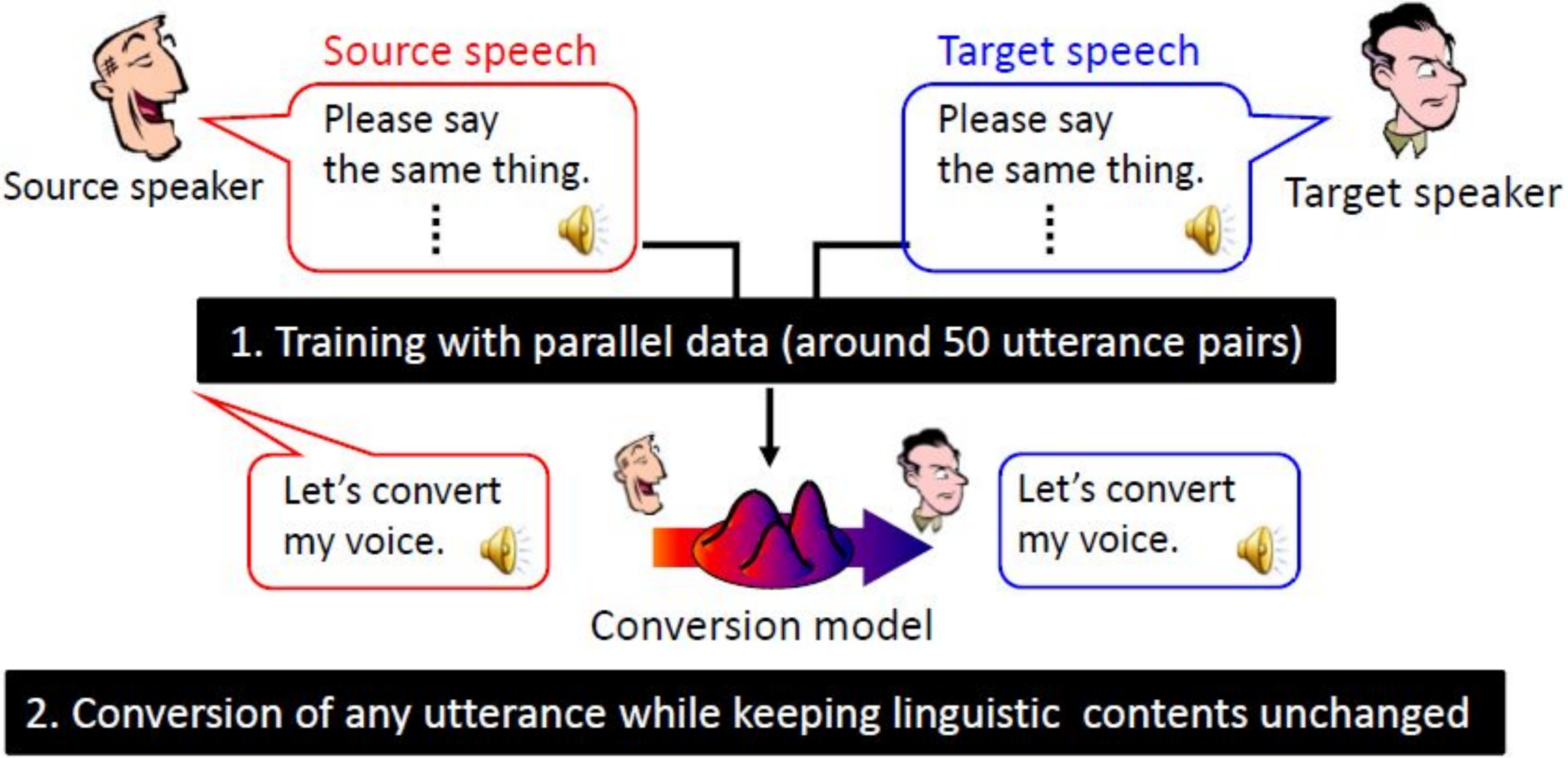
Voice Conversion



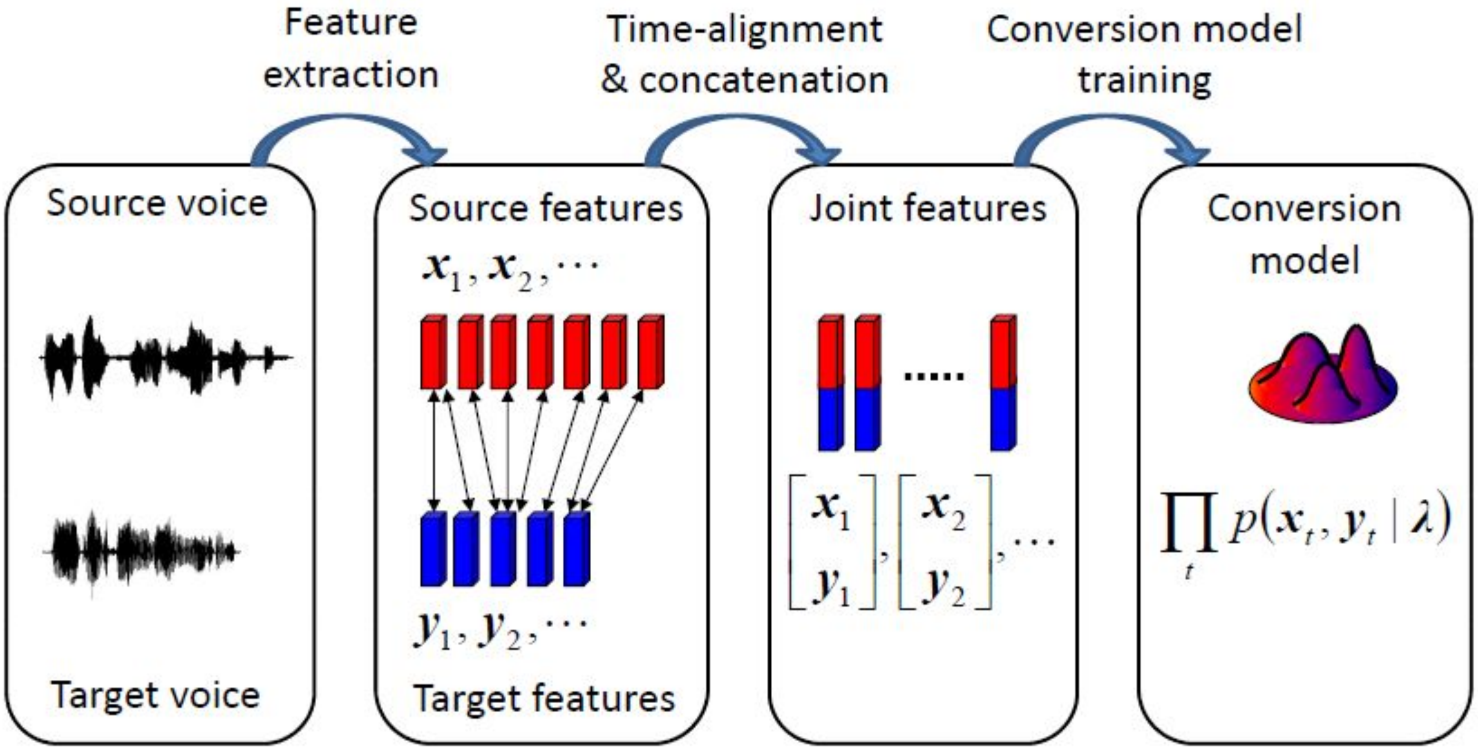
Applications

- Text-To-Speech (TTS) customization
- Film dubbing
- Design of speaking aids
- Education etc

Statistical VC

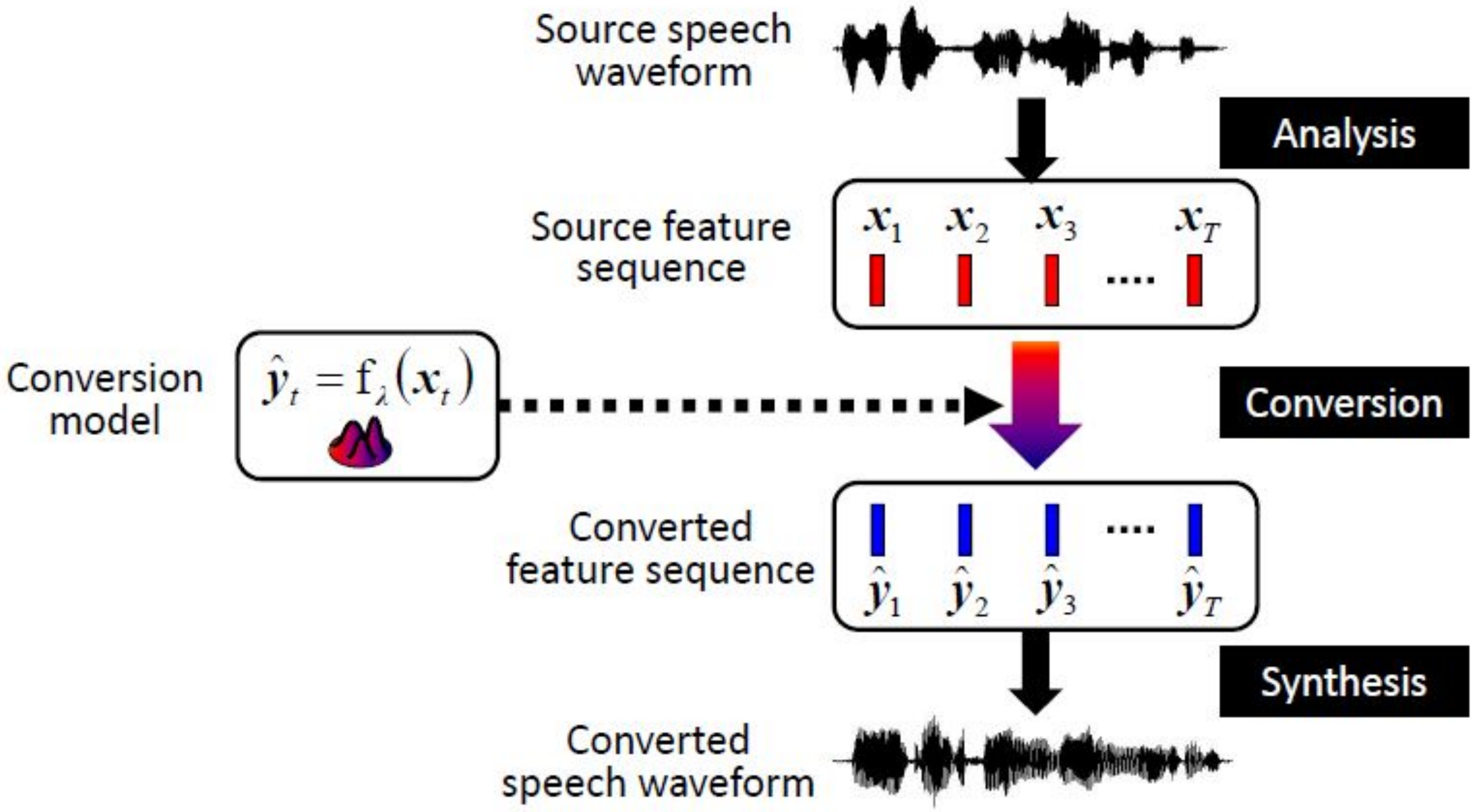


VC Training

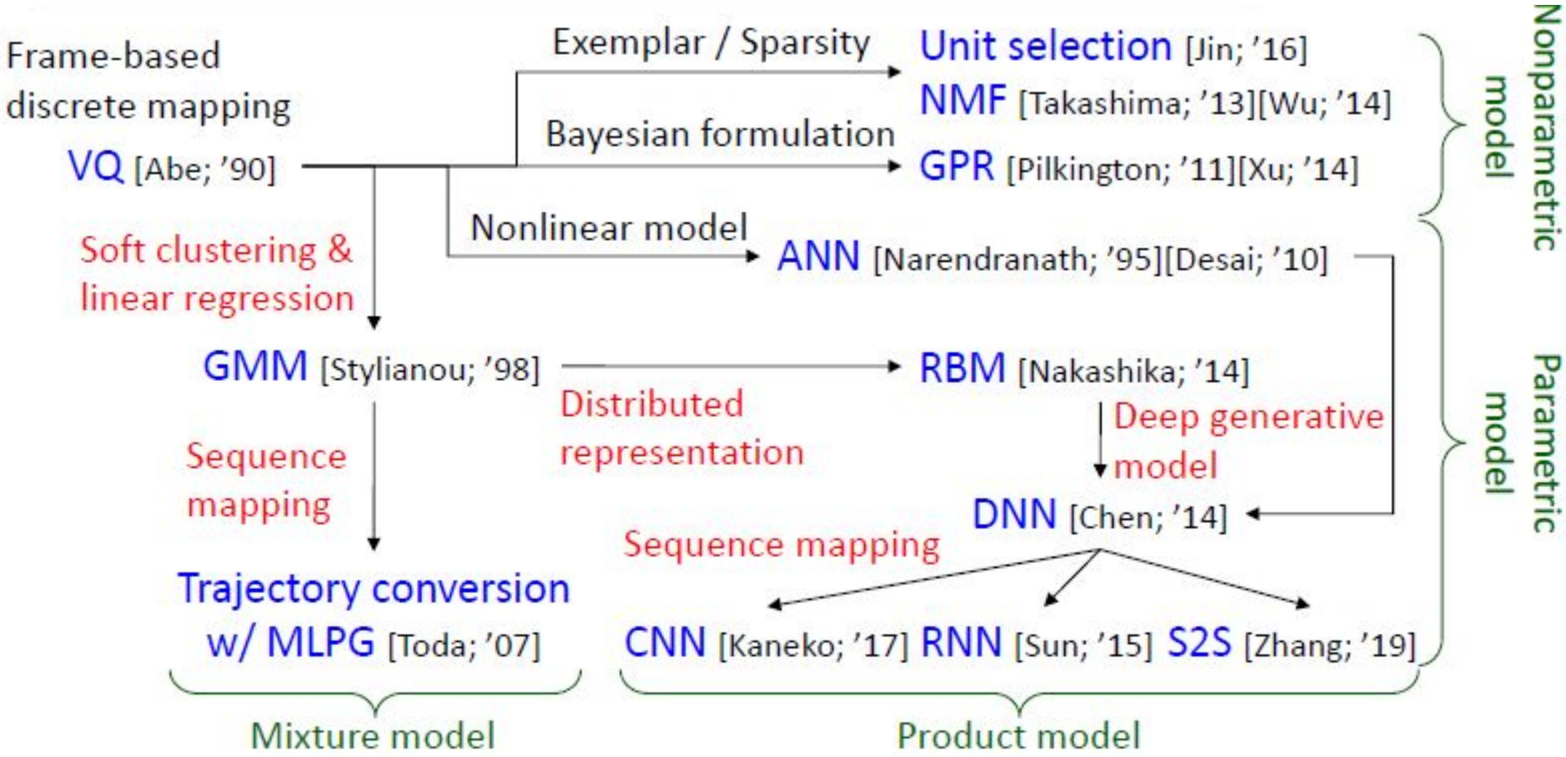


Feature extraction
Time-alignment & concatenation
Conversion model training
Source voice
Target voice
Source features
$x_1, x_2, \cdots$
$y_1, y_2, \cdots$
Target features
Joint features
$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \cdots$
Conversion model
$\prod_t p(x_t, y_t \mid \lambda)$

VC Inference



Timeline of VC Research

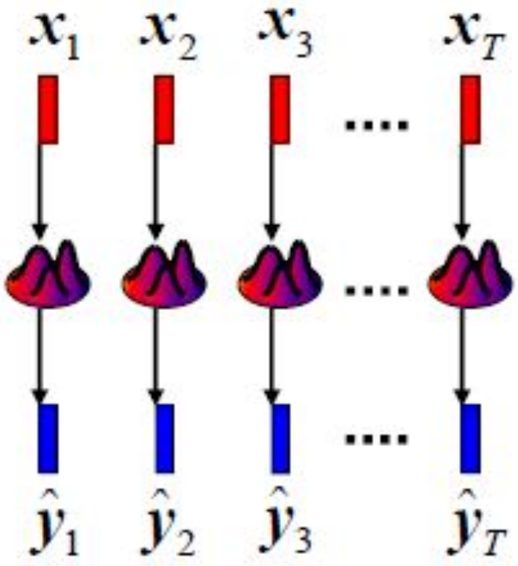


Frame-based VC

- Source feature: x
- Target feature: y
- Converted feature: ŷ

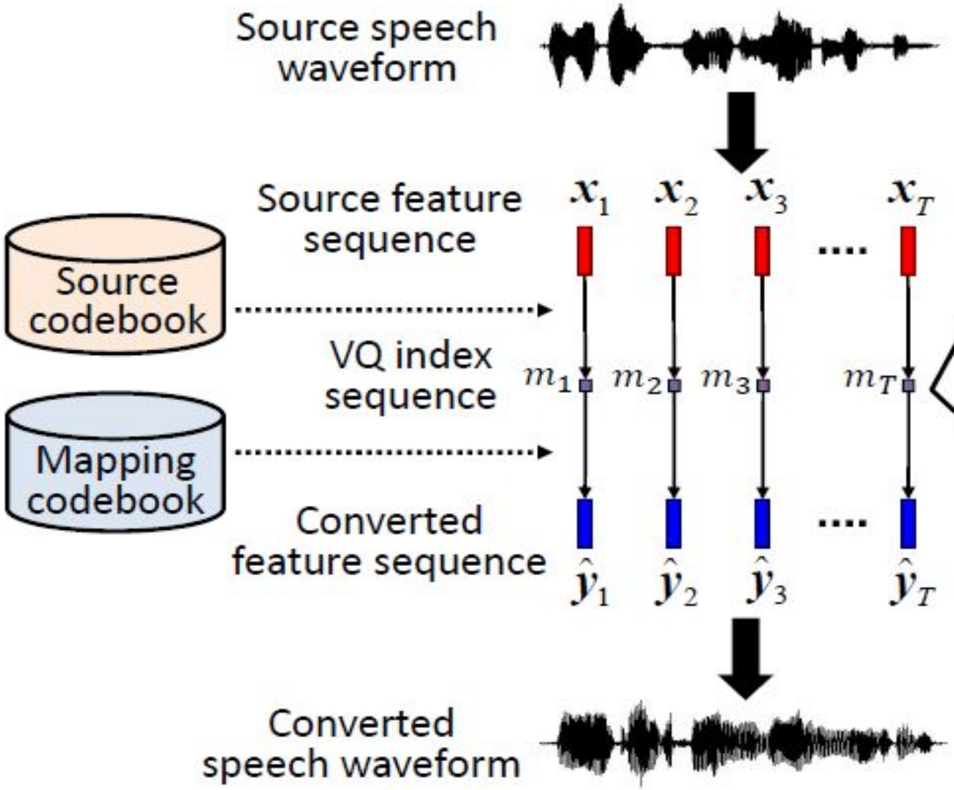
Frame-based conversion function

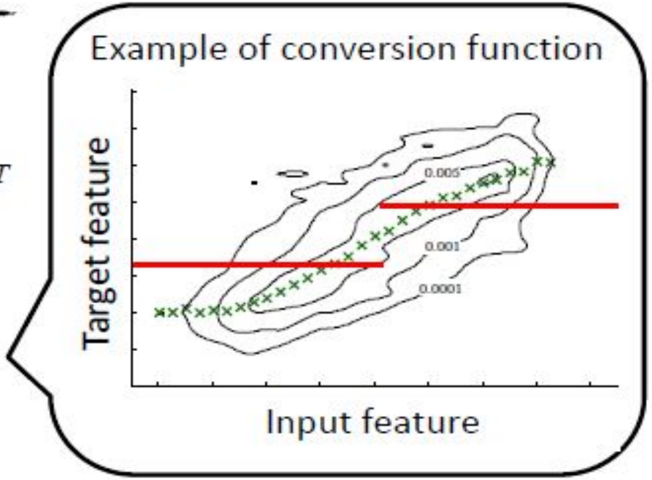
$$\hat{y}_t = f_\lambda(x_t)$$



Vector Quantization-based VC

[Abe et. al. 1990]



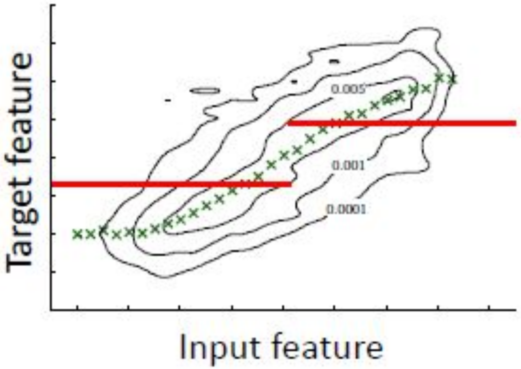


NOTE: only spectral parameter is converted w/ VQ. On the other hand, F_0 is converted with time-invariant linear transformation based on means & variances of source & target F_0s.

Discontinuous to Continuous Conversion

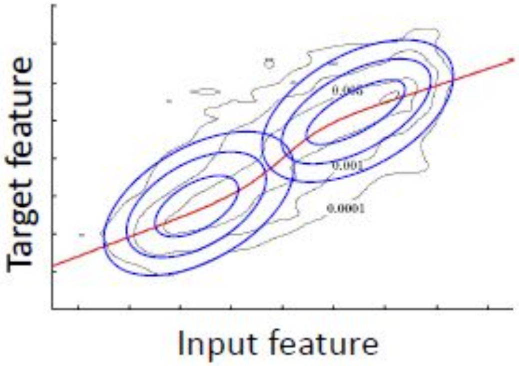
VQ-based conversion

- Discrete function w/ hard clustering
- Ignore feature correlation w/ discrete mapping



GMM-based conversion

- Continuous function w/ soft clustering
- Directly model feature correlation w/ linear regression



GMM based Conversion

Joint feature vector: $\boldsymbol{z}_t = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{y}_t \end{bmatrix}$

GMM:

Joint $p.d.f.$: $P(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\lambda}) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(\boldsymbol{z}_t; \boldsymbol{\mu}_m^{(z)}, \boldsymbol{\Sigma}_m^{(zz)})$

Maximum likelihood training

$$\underline{\hat{\boldsymbol{\lambda}}} = \arg\max \underline{\prod_t P(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\lambda})}$$

Updated model parameters

Likelihood for all joint vectors

GMM based Conversion

[Stylianou et. al. 1998]

Training of joint *p.d.f.* (modeled by a GMM) [Kain; '98]

Parameter set

Component weight α_m

$$p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\lambda}) = \sum_{m=1}^{M} \alpha_m \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{y}_t \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_m^{(x)} \\ \boldsymbol{\mu}_m^{(y)} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_m^{(xx)} & \boldsymbol{\Sigma}_m^{(xy)} \\ \boldsymbol{\Sigma}_m^{(yx)} & \boldsymbol{\Sigma}_m^{(yy)} \end{bmatrix} \right)$$

Joint feature vector $\boldsymbol{x}_t$, $\boldsymbol{y}_t$

Mean vector $\boldsymbol{\mu}_m^{(x)}$, $\boldsymbol{\mu}_m^{(y)}$

Covariance matrix $\boldsymbol{\Sigma}_m^{(xx)}$, $\boldsymbol{\Sigma}_m^{(xy)}$, $\boldsymbol{\Sigma}_m^{(yx)}$, $\boldsymbol{\Sigma}_m^{(yy)}$

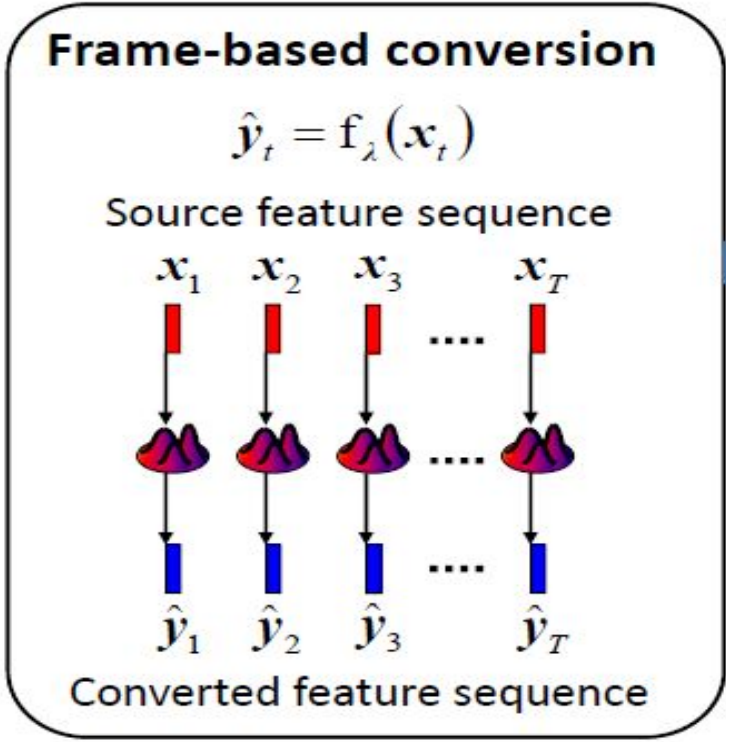
Conversion w/ conditional *p.d.f.* (also modeled by a GMM)

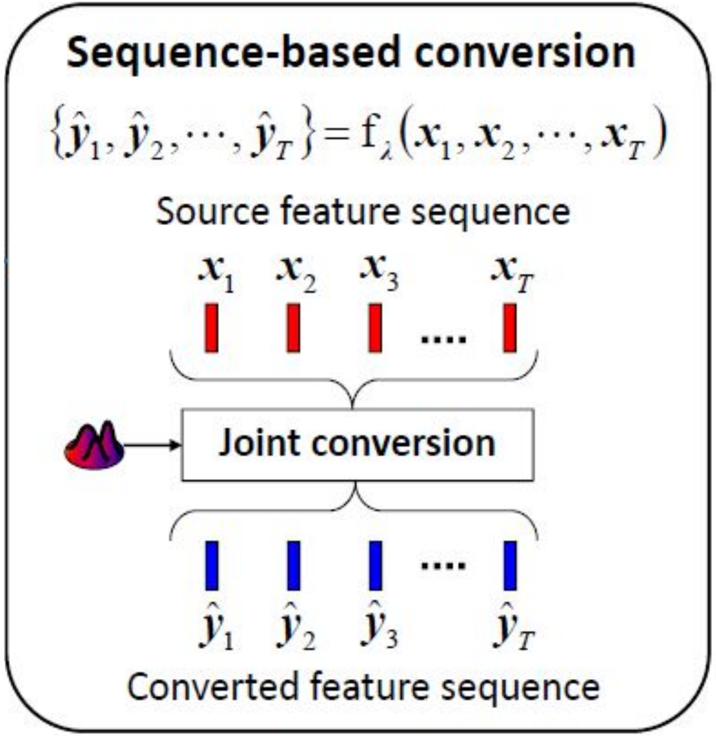
$$p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{\lambda}) = \frac{p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\lambda})}{\int p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\lambda}) \mathrm{d}\boldsymbol{y}_t} = \sum_{m=1}^{M} p(m \mid \boldsymbol{x}_t, \boldsymbol{\lambda}) \mathcal{N}\left(\boldsymbol{y}_t; \boldsymbol{\mu}_{m,t}^{(y|x)}, \boldsymbol{\Sigma}_m^{(y|x)} \right)$$

MMSE estimate: $\hat{\boldsymbol{y}}_t = \int \boldsymbol{y}_t p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{\lambda}) \mathrm{d}\boldsymbol{y}_t = \sum_{m=1}^{M} p(m \mid \boldsymbol{x}_t, \boldsymbol{\lambda}) \boldsymbol{\mu}_{m,t}^{(y|x)}$

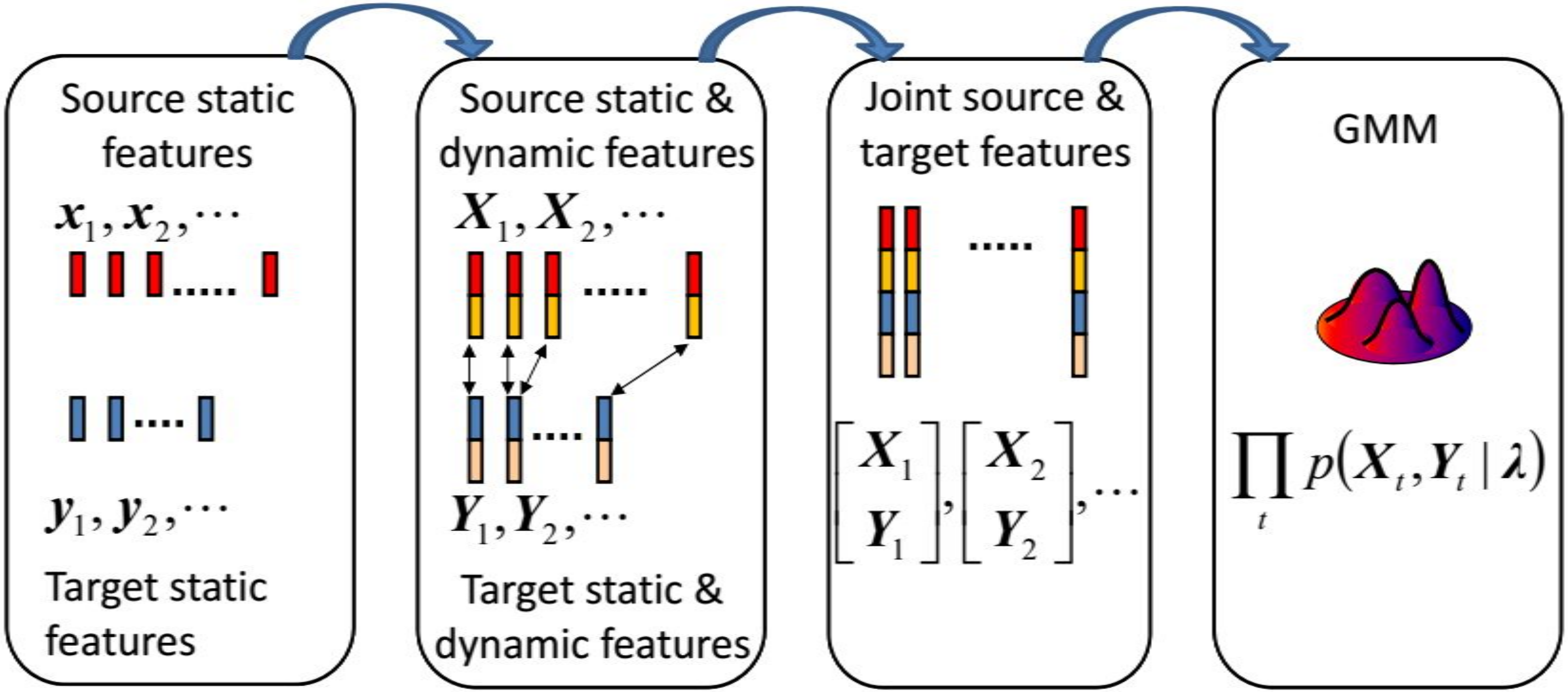
Sequence-based VC

[Toda et. al. 2007]

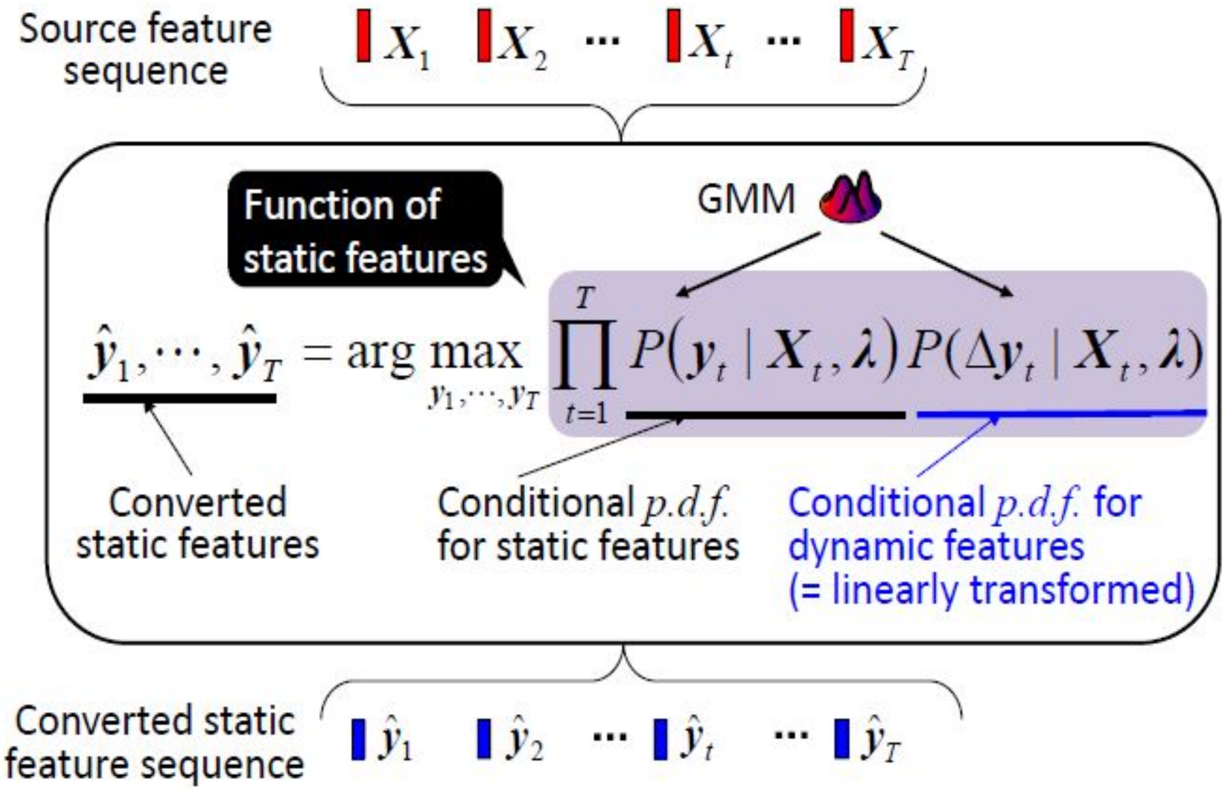




Sequence-based VC



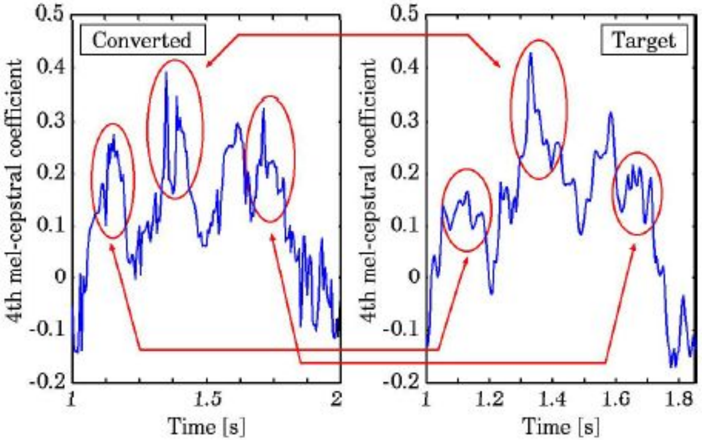
Sequence-based VC



Limitations of JD-GMM

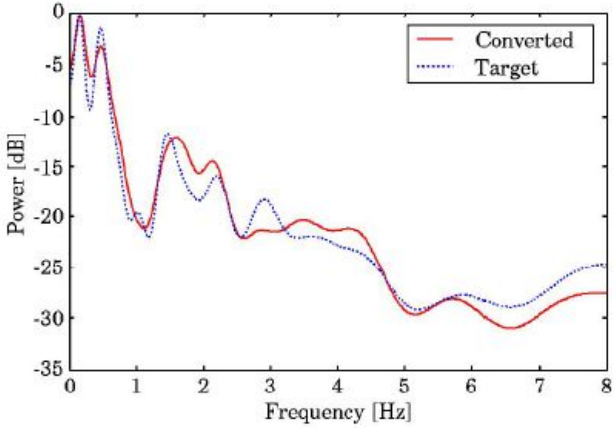
Discontinuous transitions

- Ignoring inter-frame correlation...



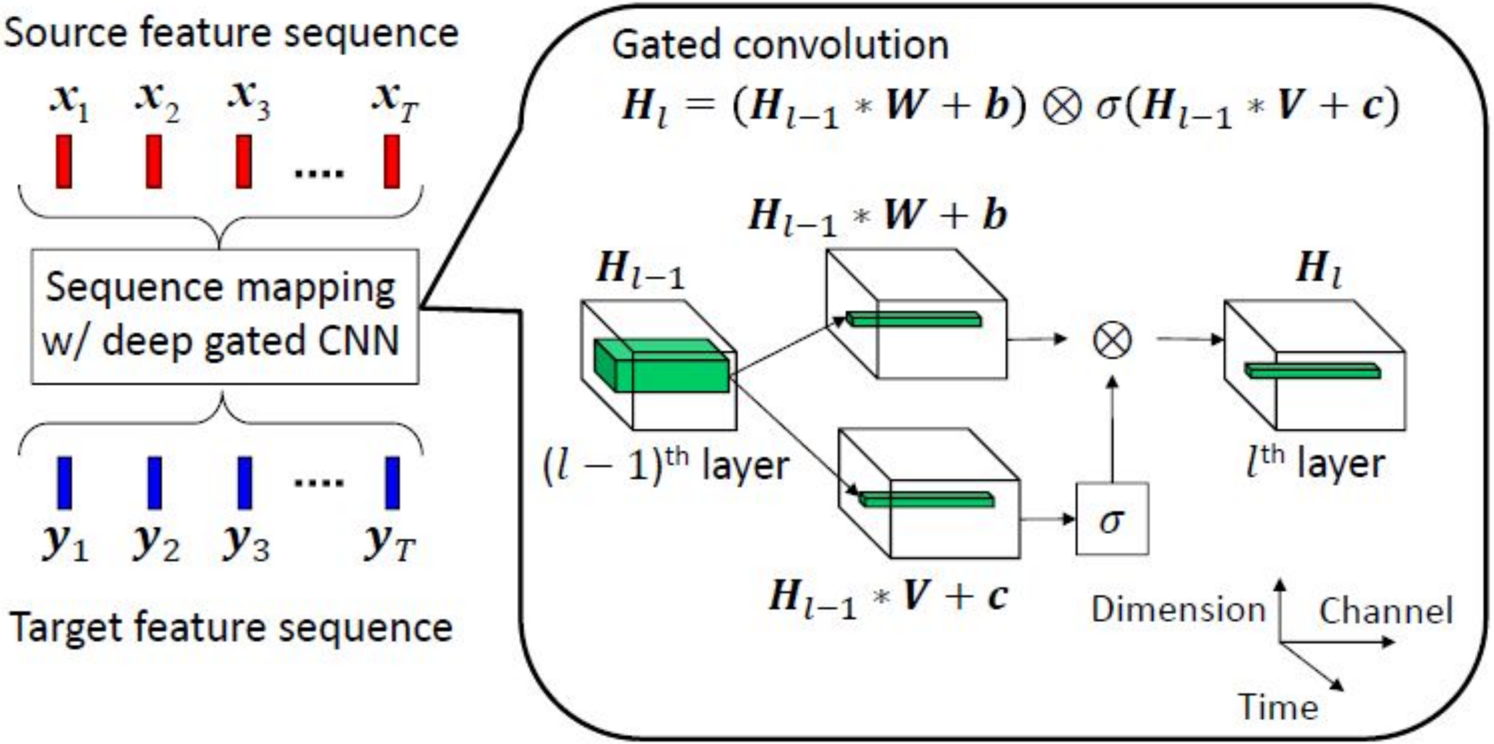
Over-smoothing effect

- Missing some characteristics not well modeled by GMM...



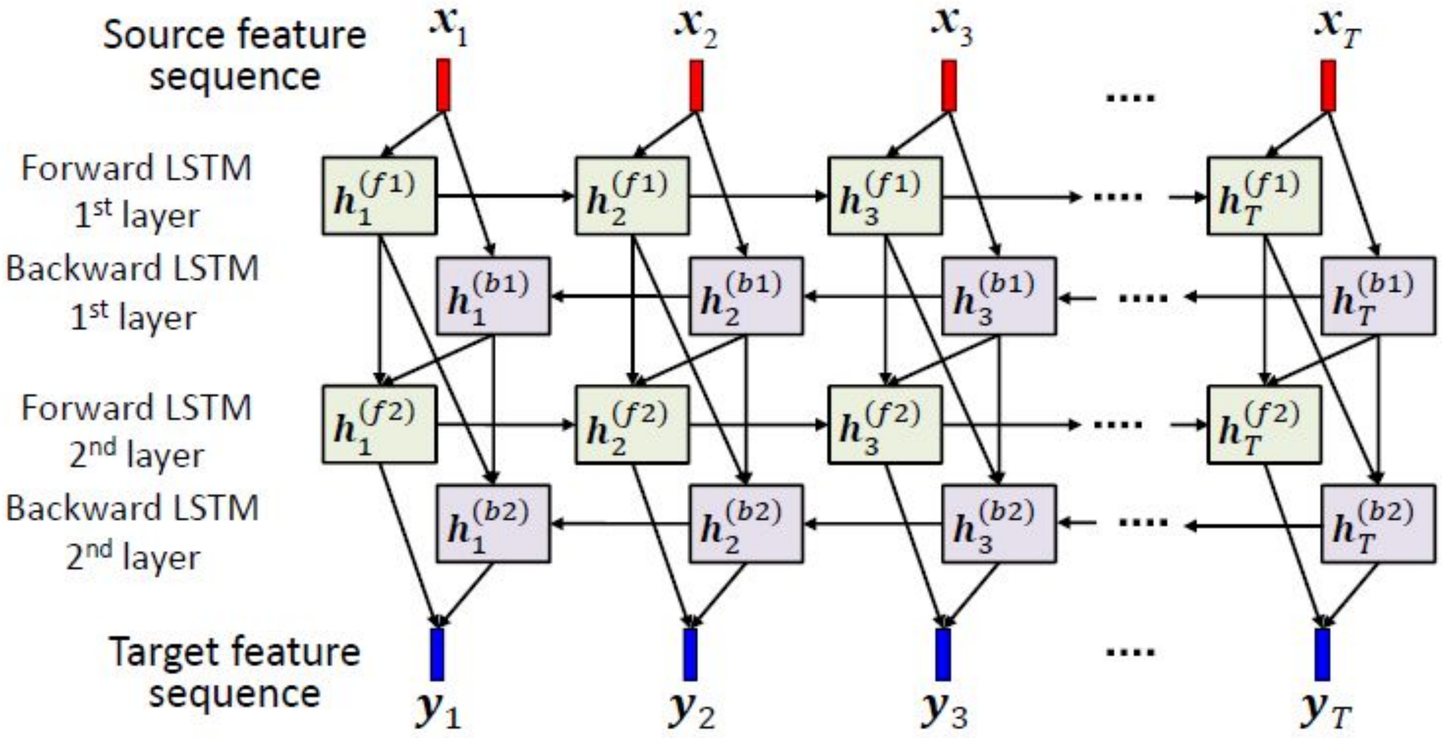
VC based on Deep Neural Networks

Sequence-based VC



[Kaneko et. al. 2017]

Sequence-based VC



[Sun et. al. 2015]

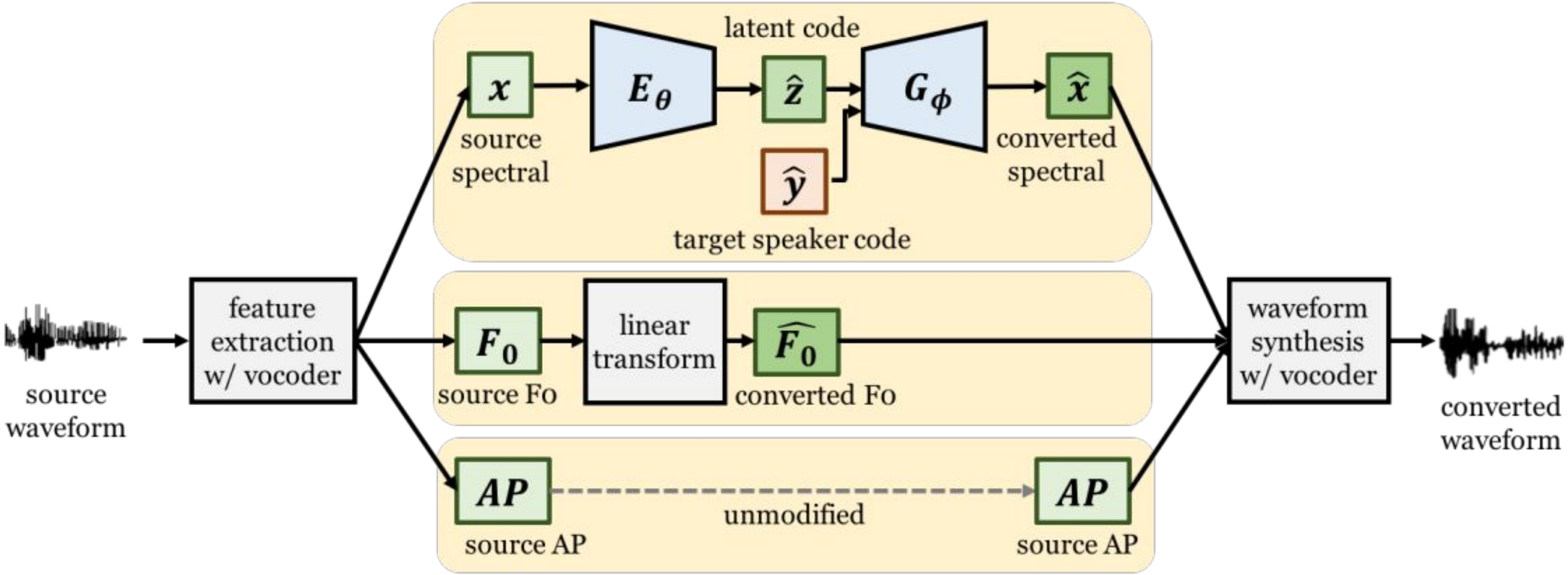
Variational Autoencoder (VAE)-VC

- The core of VAE-VC is an encoder-decoder network.
- During training, given an observed (source or target) spectral frame $\boldsymbol{x}$, a speaker-independent encoder $\boldsymbol{E_\theta}$ with parameter set $\boldsymbol{\theta}$ encodes $\boldsymbol{x}$ into a latent code: $\bar{\boldsymbol{z}} = E_\theta(\boldsymbol{x})$.
- The speaker code $\boldsymbol{y}$ of the input frame is then concatenated with the latent code, and passed to a conditional decoder $\boldsymbol{G_\phi}$ with parameter set $\boldsymbol{\phi}$ to reconstruct the input.

$$\bar{\boldsymbol{x}} = G_\phi(\bar{\boldsymbol{z}}, \boldsymbol{y}) = G_\phi(E_\theta(\boldsymbol{x}), \boldsymbol{y})$$

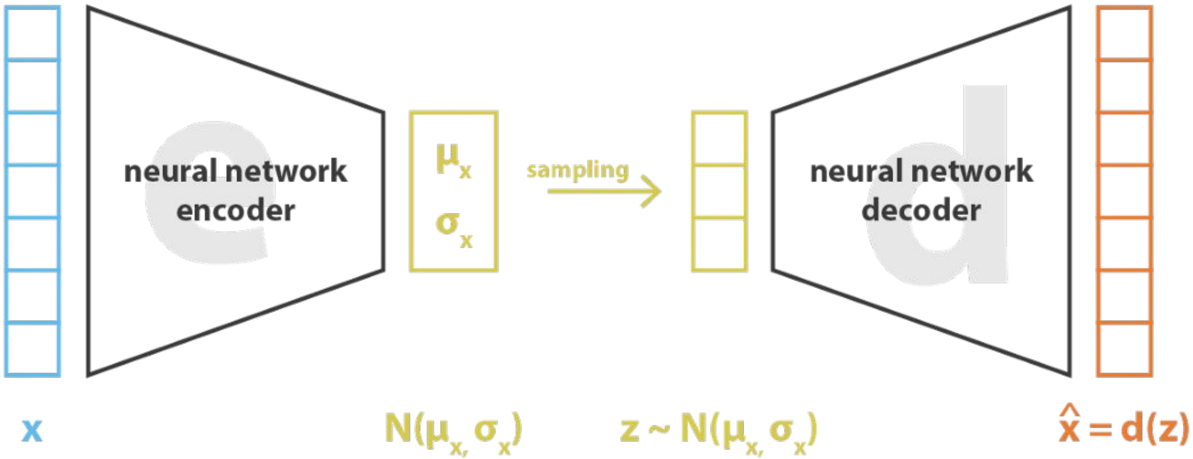
[Hsu et al, 16]

VAE-VC



latent code
x
E_θ
$\hat{z}$
G_ϕ
$\hat{x}$
source spectral
converted spectral
$\hat{y}$
target speaker code
source waveform
feature extraction w/ vocoder
F_0
linear transform
$\widehat{F_0}$
source Fo
converted Fo
waveform synthesis w/ vocoder
converted waveform
AP
unmodified
AP
source AP
source AP

VAE



$$\text{loss} = \| x - \hat{x} \|^2 + \mathrm{KL}[\, N(\mu_x, \sigma_x), N(0, I) \,] = \| x - d(z) \|^2 + \mathrm{KL}[\, N(\mu_x, \sigma_x), N(0, I) \,]$$

VAE

- The model parameters can be obtained by maximizing the variational lower bound:

$$\mathcal{L}_{vae}(\theta, \phi; \boldsymbol{x}, \boldsymbol{y}) = \mathcal{L}_{recon}(\boldsymbol{x}, \boldsymbol{y}) + \mathcal{L}_{lat}(\boldsymbol{x}),$$
$$\mathcal{L}_{recon}(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{\boldsymbol{z} \sim q_\theta(\bar{\boldsymbol{z}}|\boldsymbol{x})}\left[\log p_\phi(\bar{\boldsymbol{x}}|\boldsymbol{z}, \boldsymbol{y})\right],$$
$$\mathcal{L}_{lat}(\boldsymbol{x}) = -D_{KL}(q_\theta(\bar{\boldsymbol{z}}|\boldsymbol{x}) \| p(\boldsymbol{z})),$$

$q_\theta(\bar{\boldsymbol{z}}|\boldsymbol{x})$: approximate posterior.

$p_\phi(\bar{\boldsymbol{x}}|\boldsymbol{z}, \boldsymbol{y})$: data likelihood.

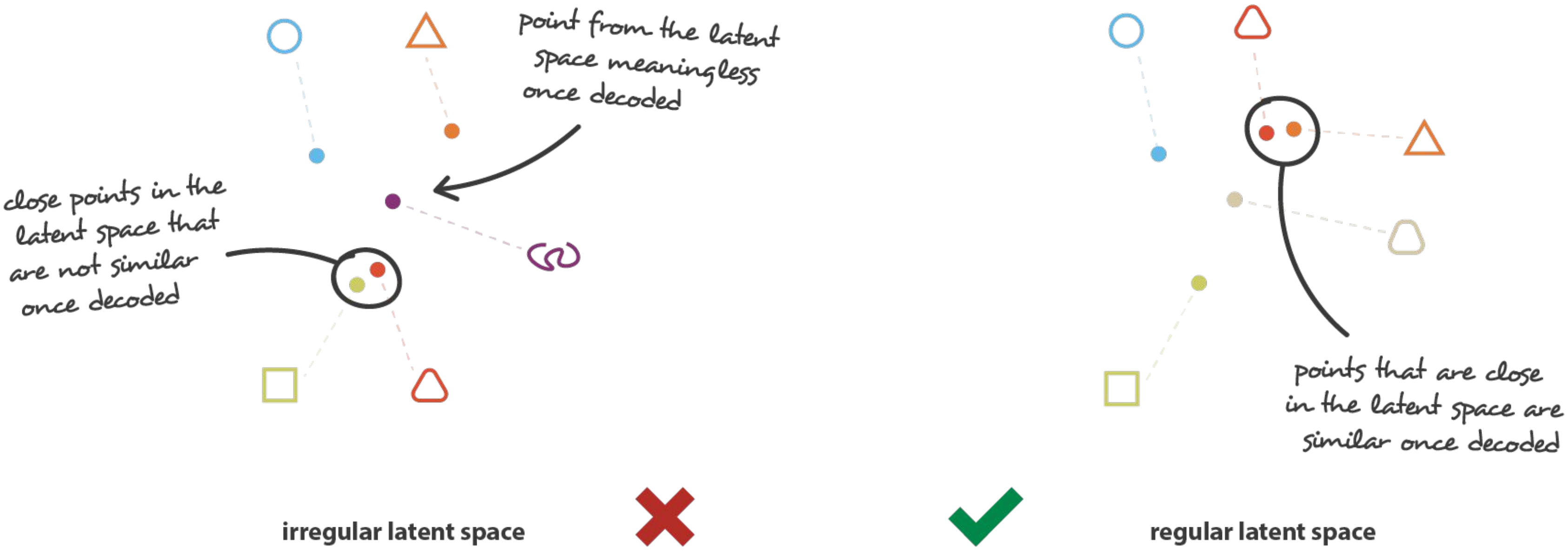
$p(\boldsymbol{z})$: prior distribution of the latent space.

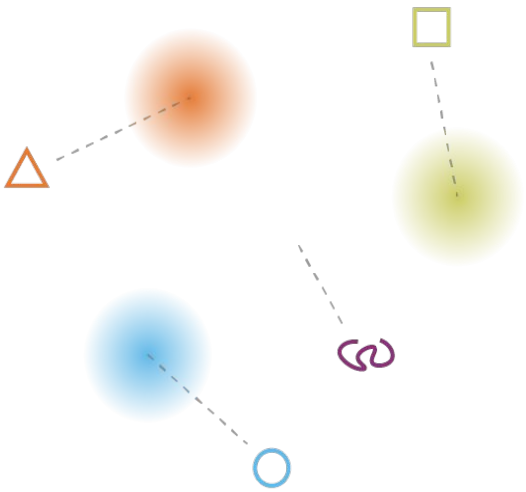
- Conversion phase:

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

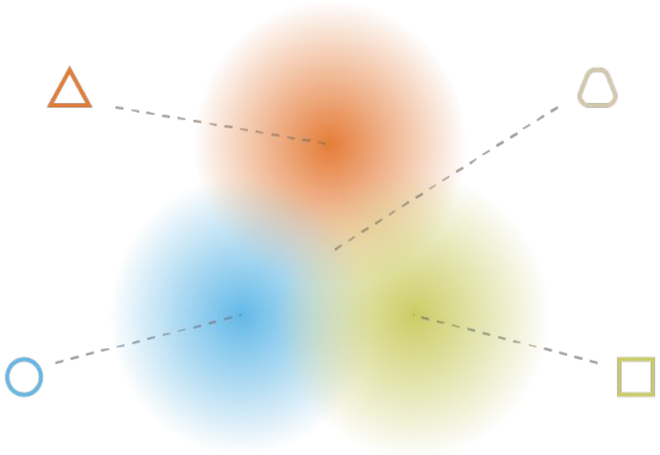
[Hsu et al, 16]

Intuitions about Regularization





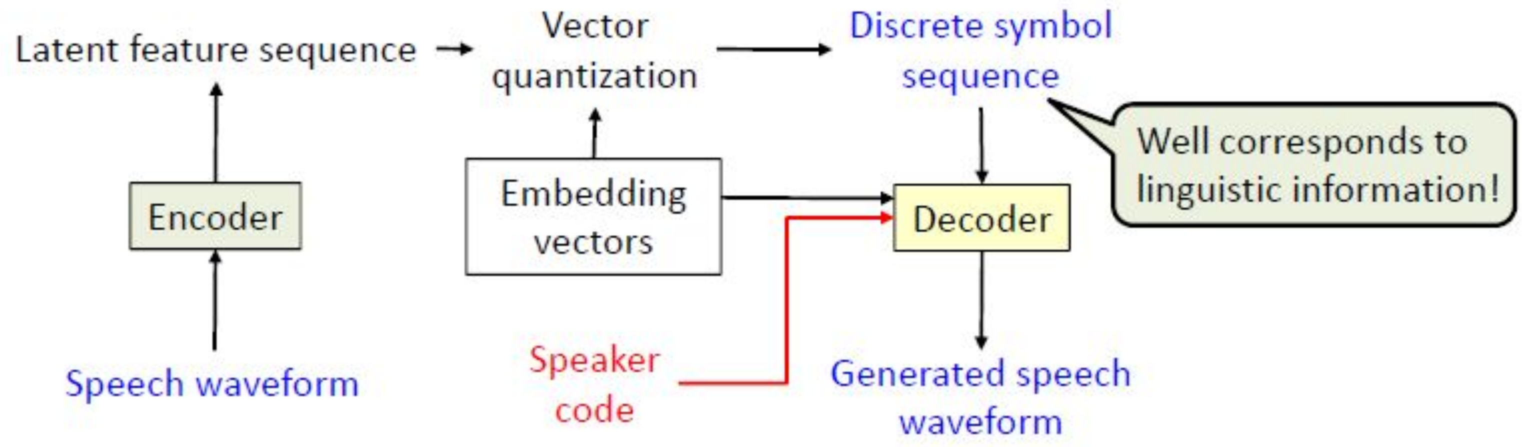
what can happen without regularisation



what we want to obtain with regularisation

Vector Quantization VAE (VQ-VAE)

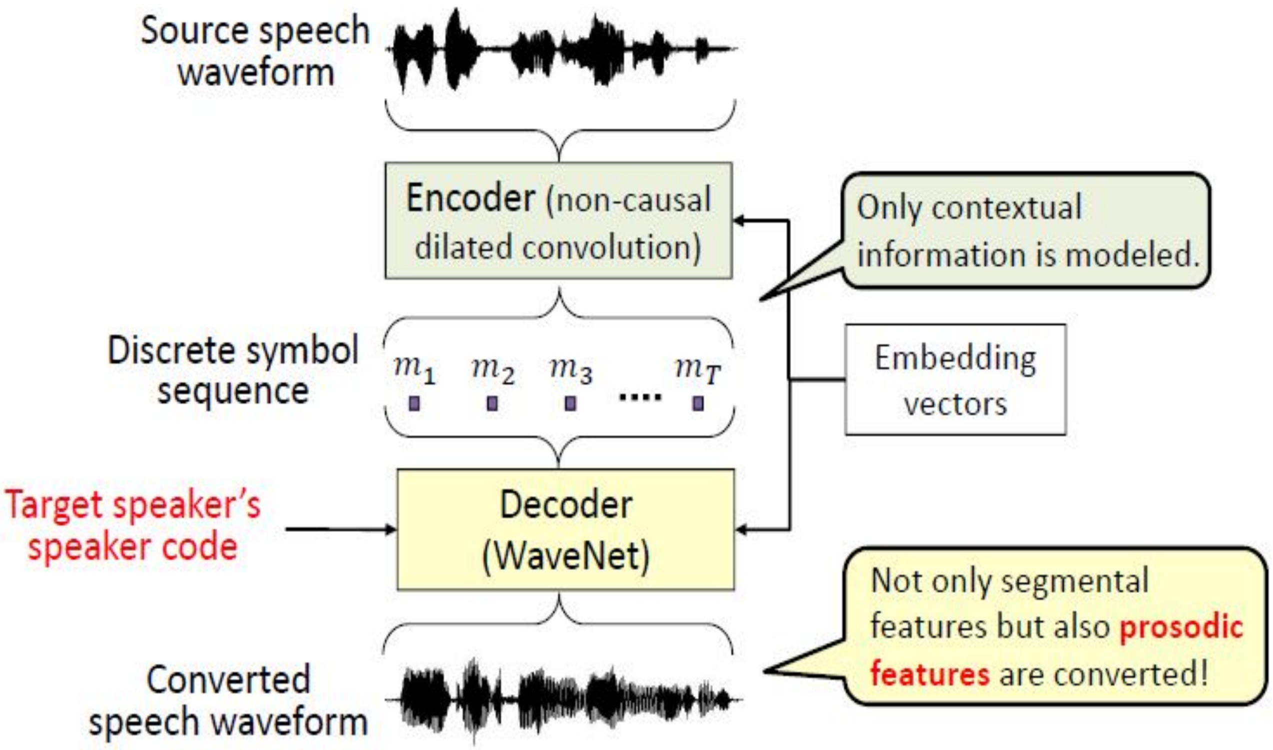
- Directly encode speech waveform into a discrete symbol sequence capturing long-term dependencies (including prosodic features!) by using a dilated convolution network



[van den Oord; ’17]

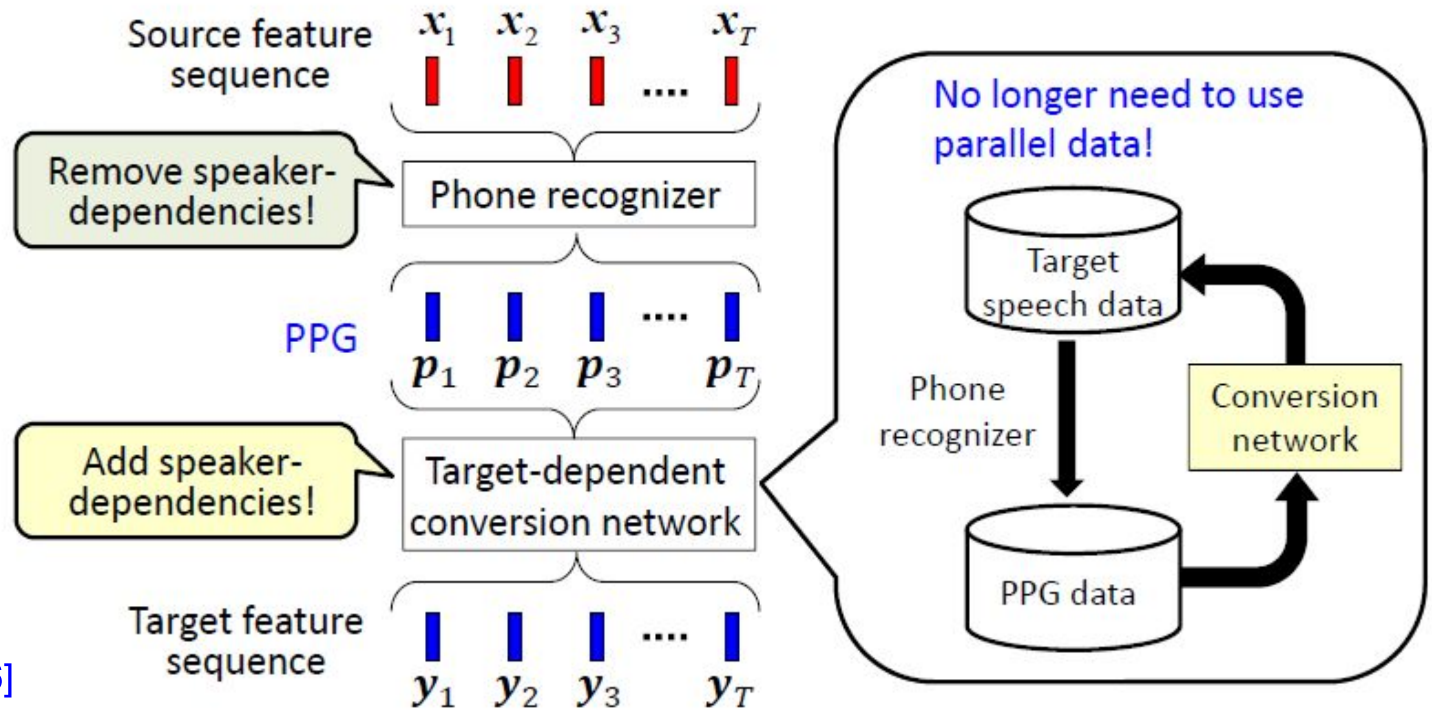
VC based on VQ VAE

- Extract phoneme posteriorgram (PPG) as speaker-independent contextual features.



Phoneme Posteriogram VC

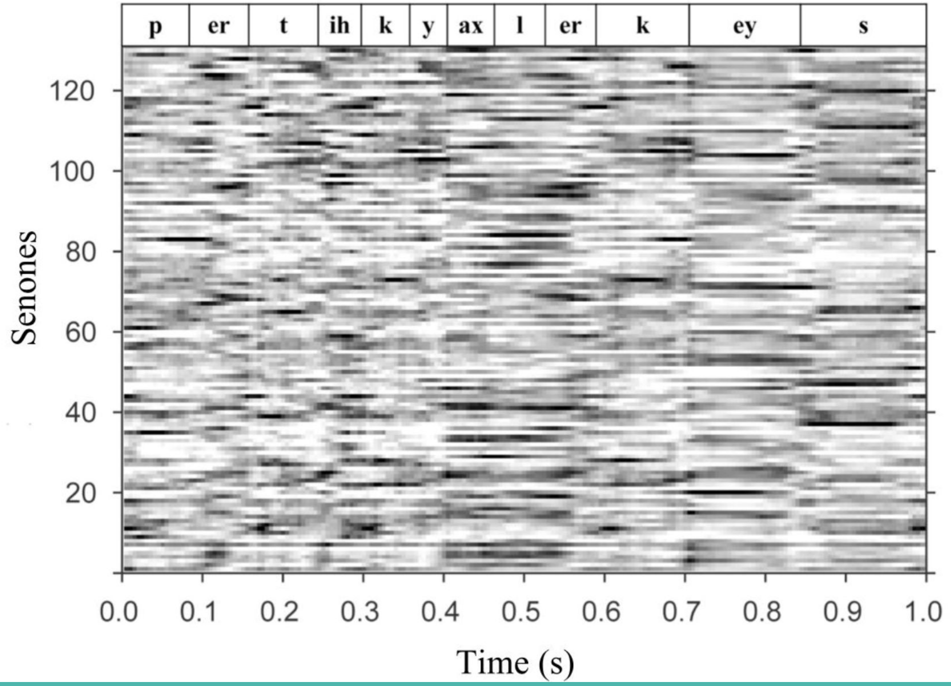
- Extract phoneme posteriorgram (PPG) as speaker-independent contextual features and use them as input of the conversion network.



[Sun et. al. 16]

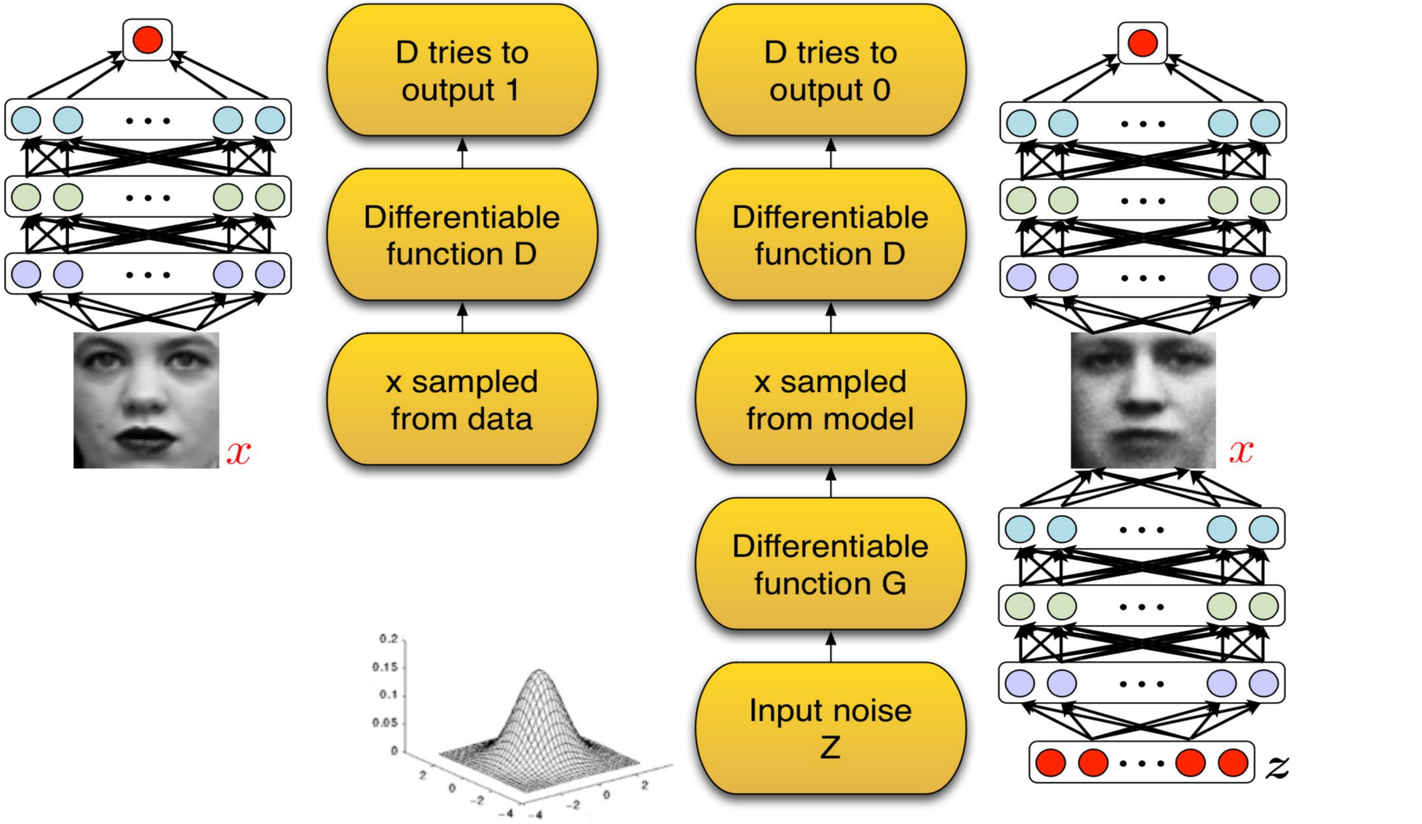
Phoneme Posteriogram VC

- 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

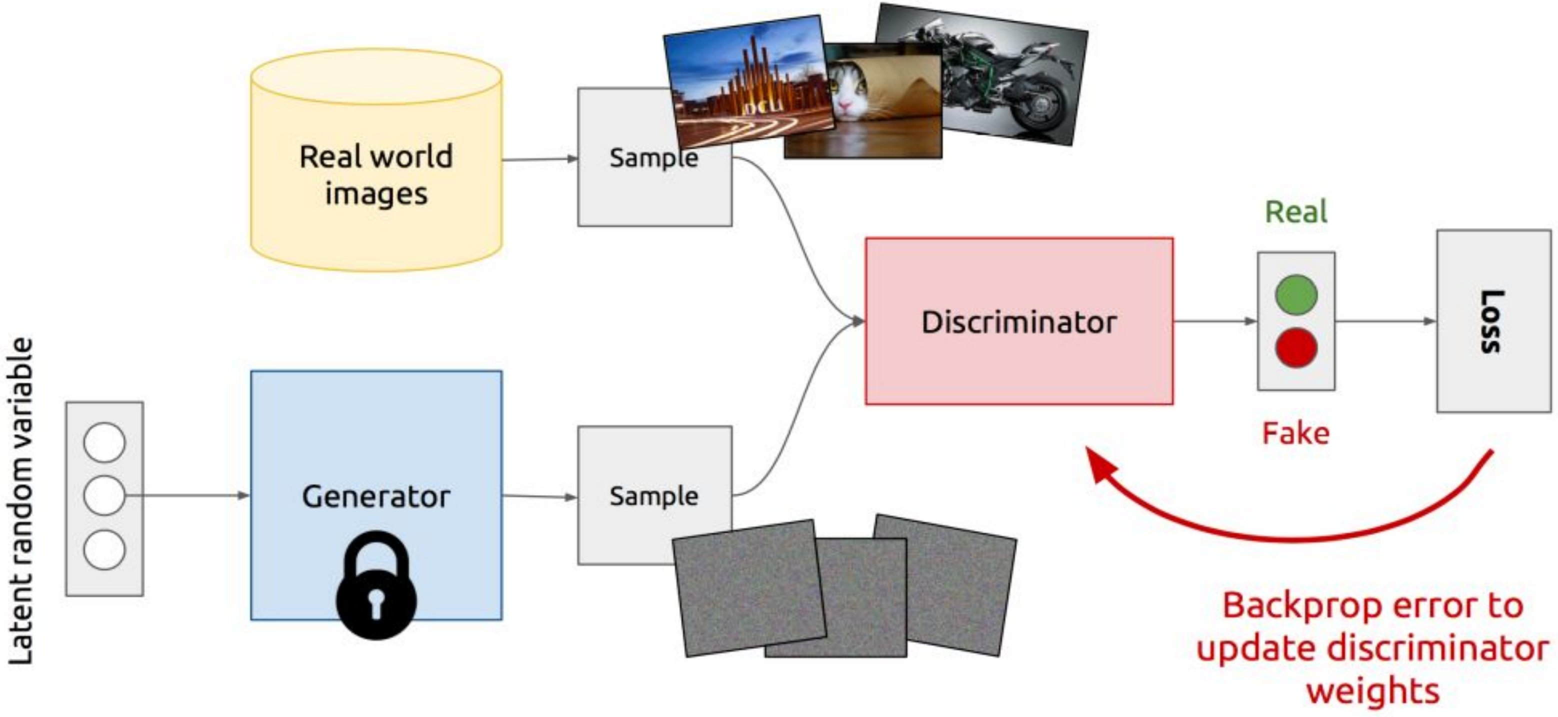


VC based on Generative Adversarial Networks

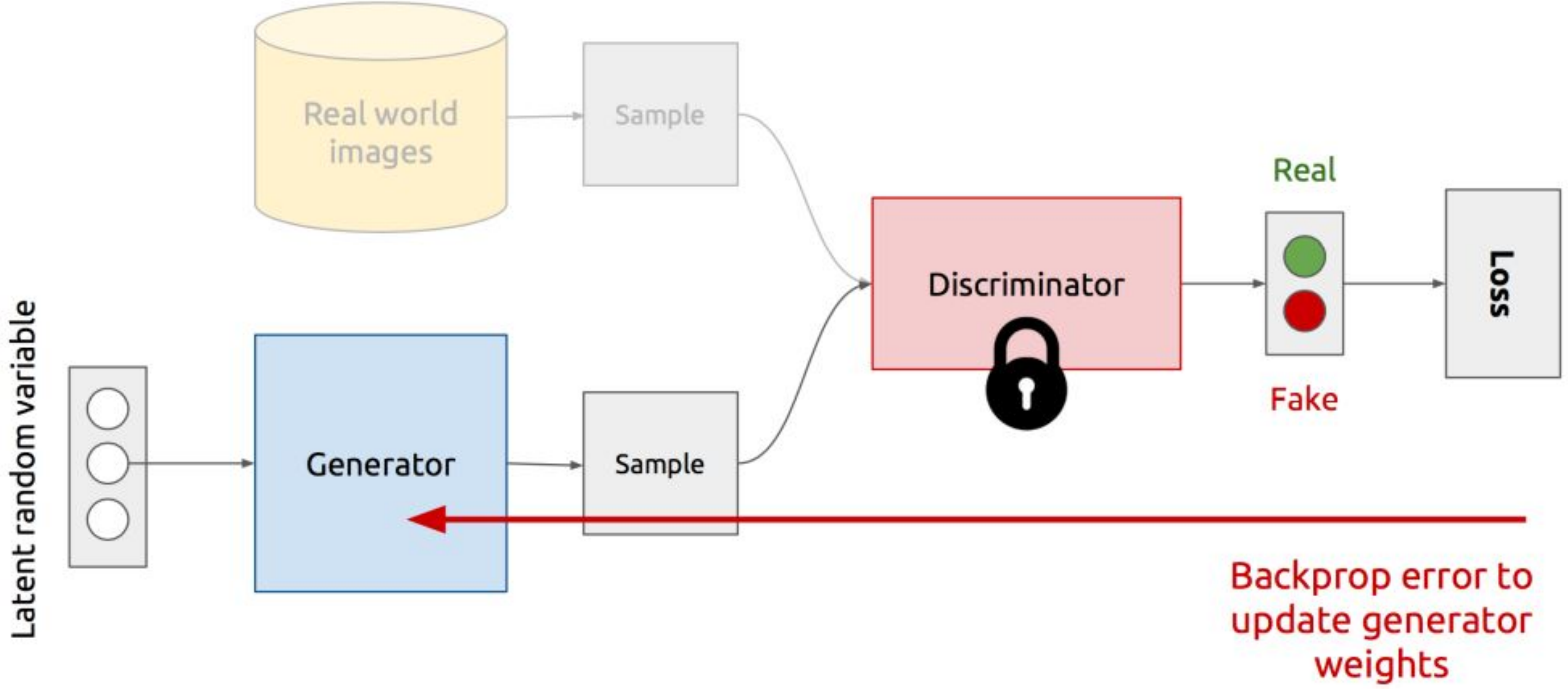
GAN Formulation



Discriminator Training



Generator 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

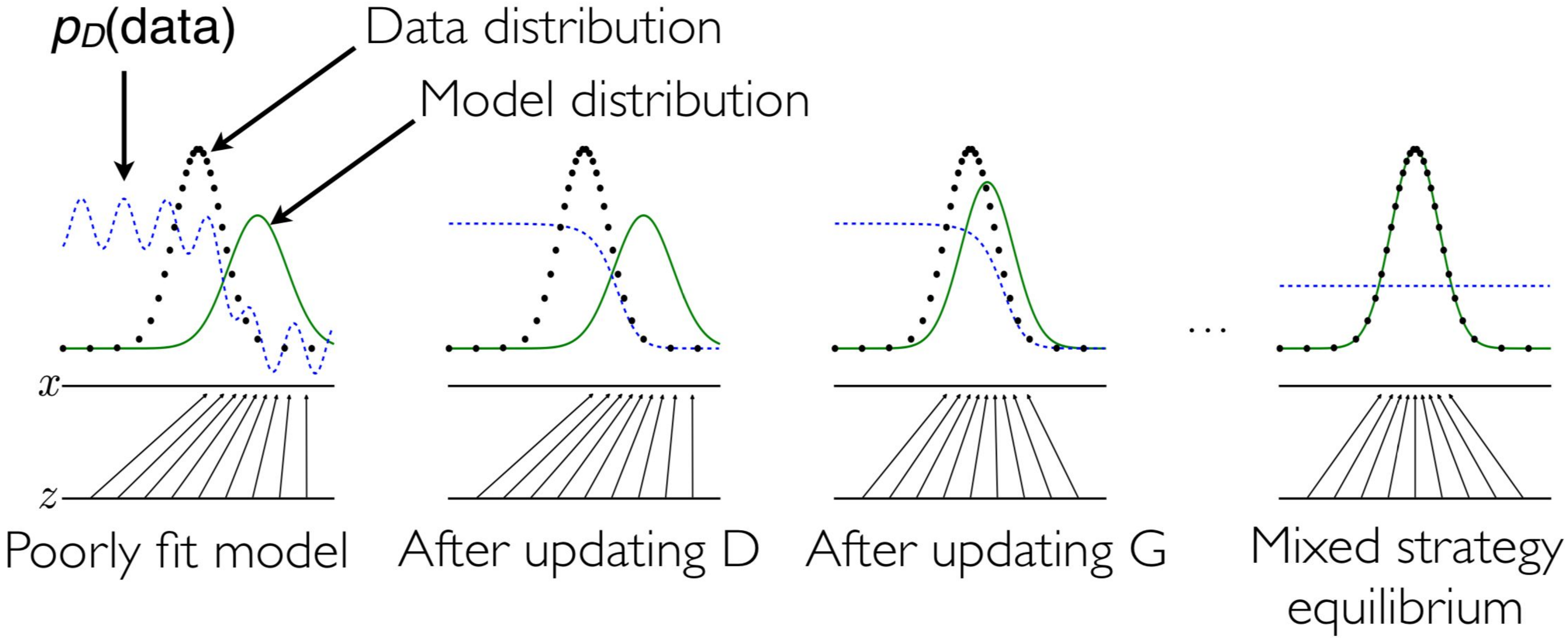
Maximize D

x is sampled from real data

z is sampled from N(0, I)

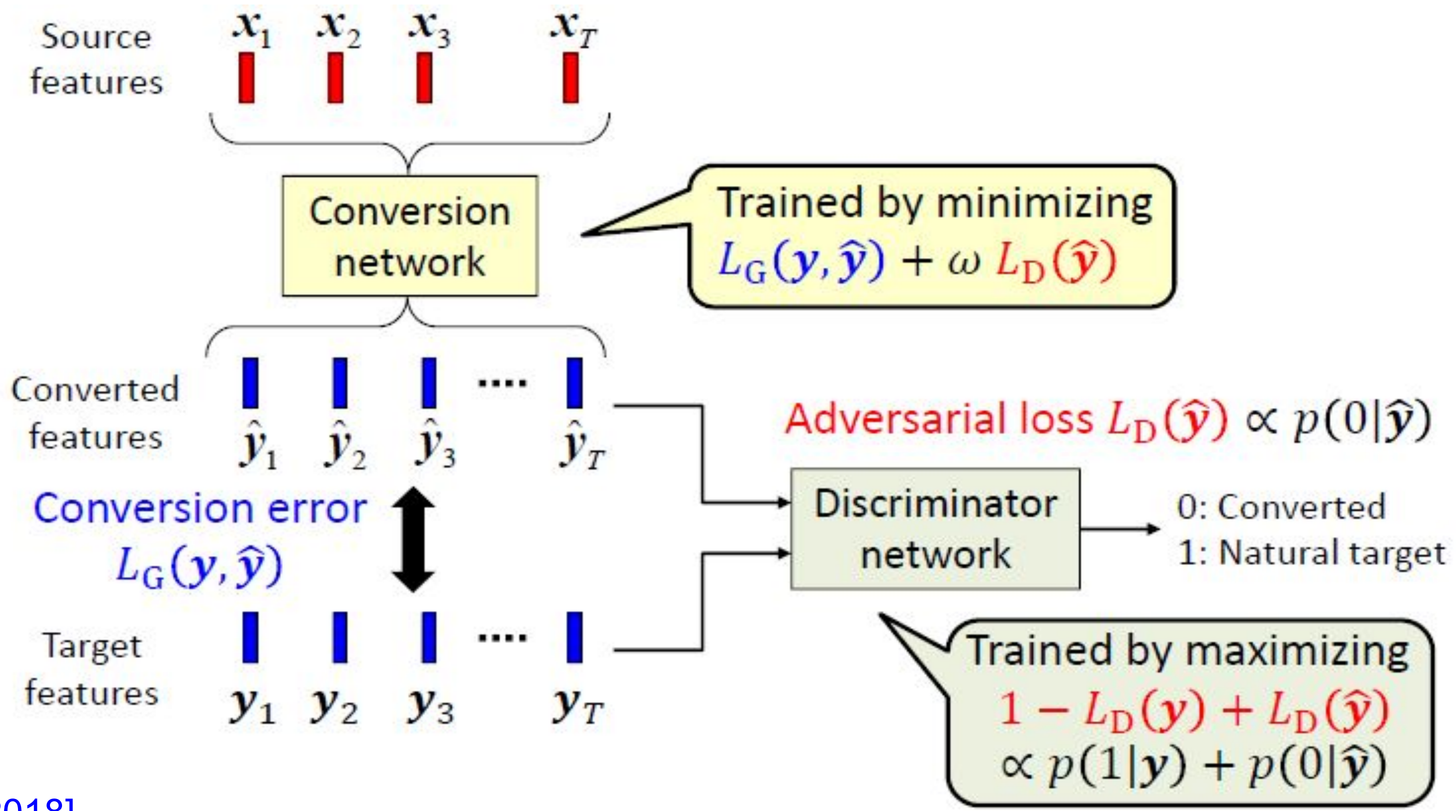
fake

Learning GANs



[Goodfellow et al., 2017]

GAN-based VC



[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.

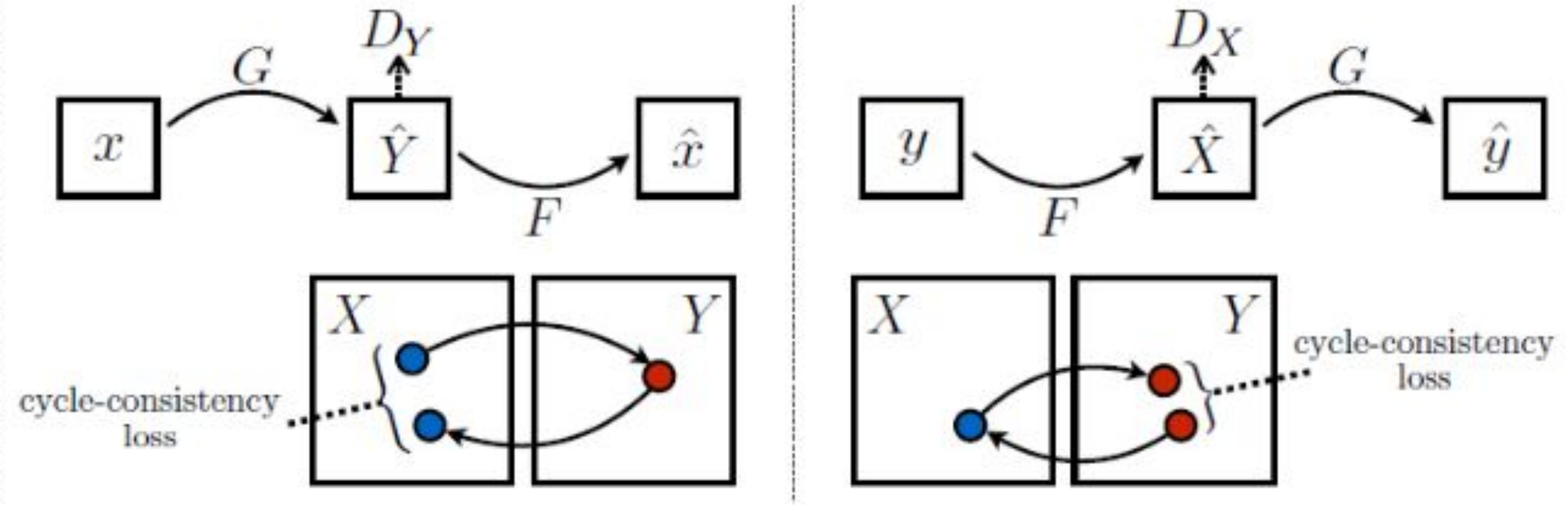
- In a CycleGAN, forward and inverse mappings are simultaneously learned using an adversarial loss and cycle-consistency loss.

- Two important losses are introduced:
  - Adversarial loss
  - cycle-consistency loss
  - identity-mapping loss

CycleGAN losses

Adversarial loss

Adversarial loss



[Kaneko et. al. 2018]

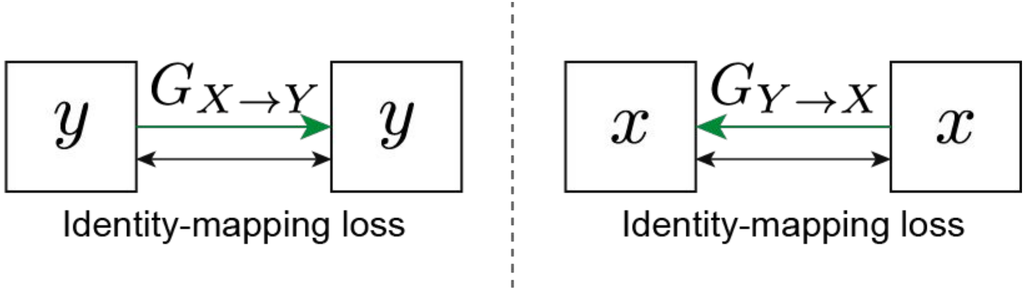
CycleGAN losses

- Two mapping function (Adversarial loss): *G* and *F*. $G : X \rightarrow Y$ and $F : Y \rightarrow X$
- Cycle-consistency loss:
  - Forward: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$
  - Backward: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$
- Adversarial loss + cycle-consistency loss:

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

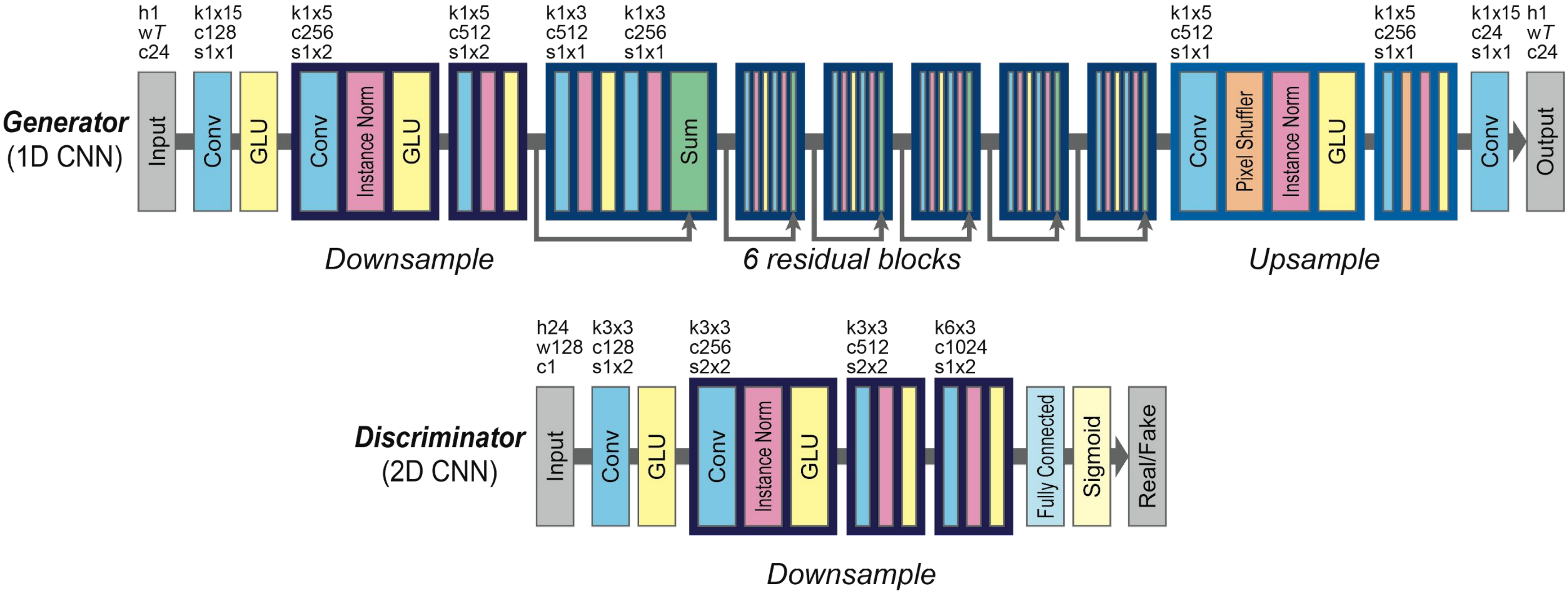
Identity-mapping loss

- 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.



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

CycleGAN Architecture



Sound Samples

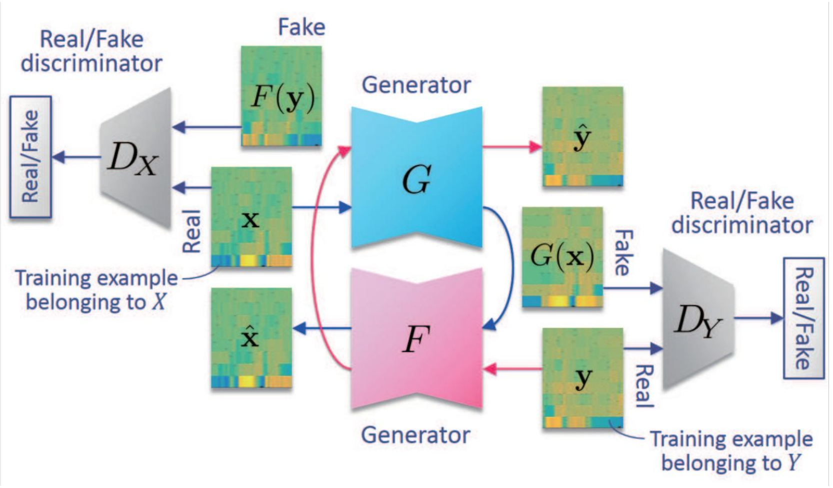
http://www.kecl.ntt.co.jp/people/kaneko.takuhiro/projects/cyclegan-vc/

StarGAN Voice Conversion

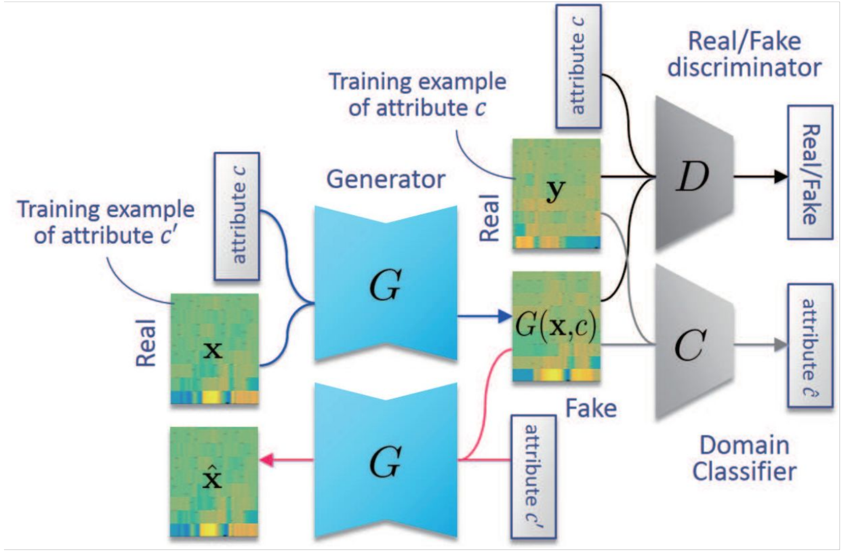
- A non-parallel many-to-many voice conversion (VC) by using a variant of a genitive adversarial network called StarGAN.
- Generator (G) takes an acoustic feature with an attribute c as the inputs and generates an acoustic feature sequence $y' = G(x, c)$.
- Discriminator (D) is designed to produce a probability $D(y, c)$ that an input y is a real speech feature.
- A domain classifier (C) predicts classes of the input.

StarGAN training

CycleGAN



StarGAN



[Kameoka et. al. 2018]

StarGAN training losses

Adversarial loss:

- Adversarial losses for discriminator *D* and generator *G*, respectively, where y denotes a training example of an acoustic feature sequence of real speech with attribute *c* and *x* denotes that with an arbitrary attribute.

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

StarGAN training losses

Domain Classification loss:

- Domain classification losses for classifier C and generator G is described.

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

StarGAN training losses

Cycle Consistency Loss:

- To encourage *G(x, c)* to be a bijection, a cycle consistency loss is implemented, where x denotes an acoustic feature sequence of real speech with attribute *c'*.

$$\mathcal{L}_{\mathrm{cyc}}(G) = \mathbb{E}_{c'\sim p(c),\mathbf{x}\sim p(\mathbf{x}|c'),c\sim p(c)}[\|G(G(\mathbf{x},c),c') - \mathbf{x}\|_{\rho}],$$

StarGAN training losses

Identity mapping loss:

- Ensure that an input into *G* will remain unchanged when the input already belongs to the target attribute *c'*.

$$\mathcal{L}_{\mathrm{id}}(G) = \mathbb{E}_{c' \sim p(c), \mathbf{x} \sim p(\mathbf{x}|c')}[||G(\mathbf{x}, c') - \mathbf{x}||_{\rho}],$$

StarGAN Objective Function

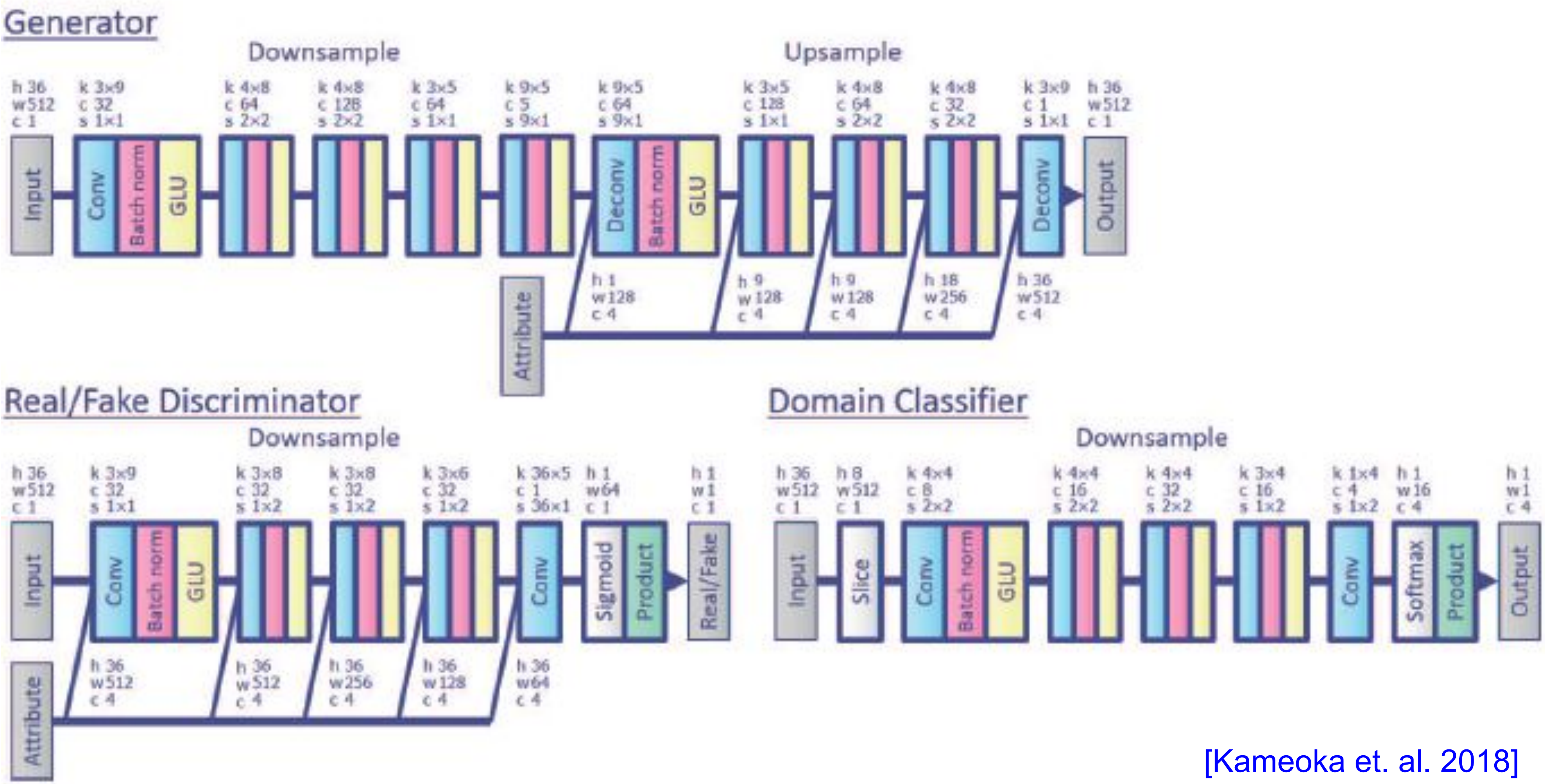
Objective function :

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

$$\mathcal{I}_G(G) = \mathcal{L}_{\text{adv}}^{G}(G) + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}}^{G}(G) + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}}(G) + \lambda_{\text{id}}\mathcal{L}_{\text{id}}(G)$$

$$\mathcal{I}_D(D) = \mathcal{L}_{\text{adv}}^{D}(D),$$

$$\mathcal{I}_C(C) = \mathcal{L}_{\text{cls}}^{C}(C),$$



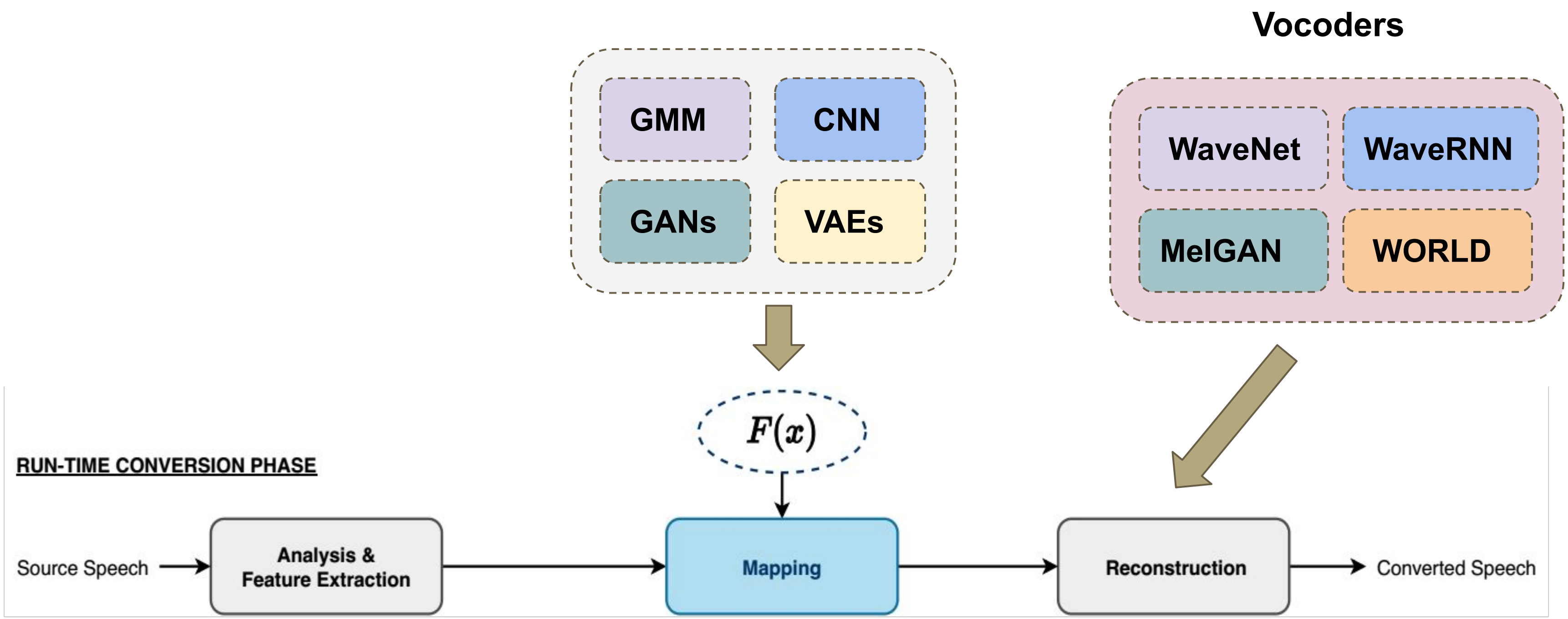
[Kameoka et. al. 2018]

Sound Samples

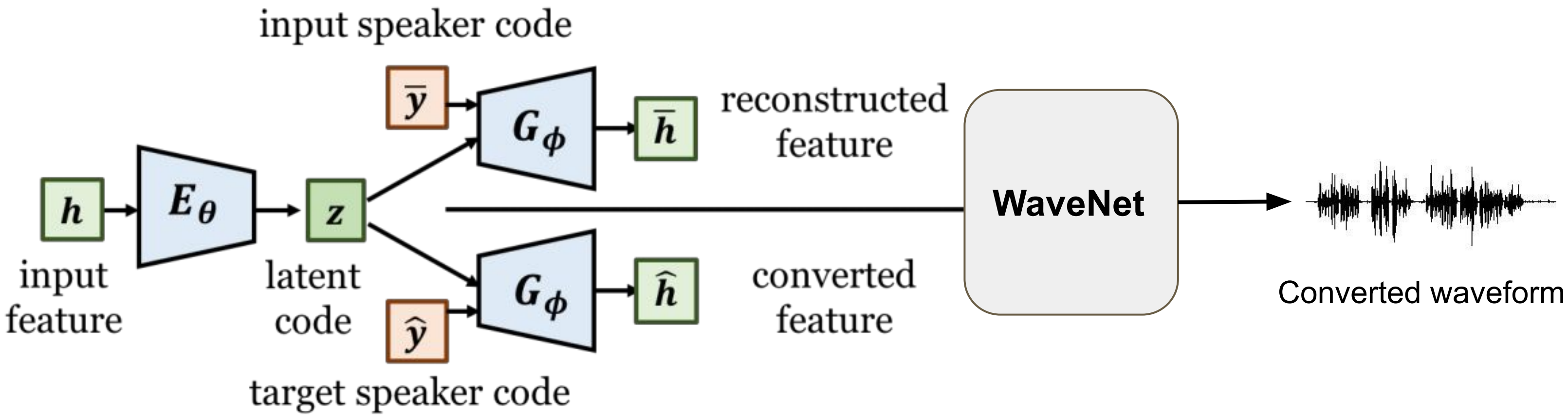
http://www.kecl.ntt.co.jp/people/kameoka.hirokazu/Demos/stargan-vc/

Various Vocoders in VC

General Framework



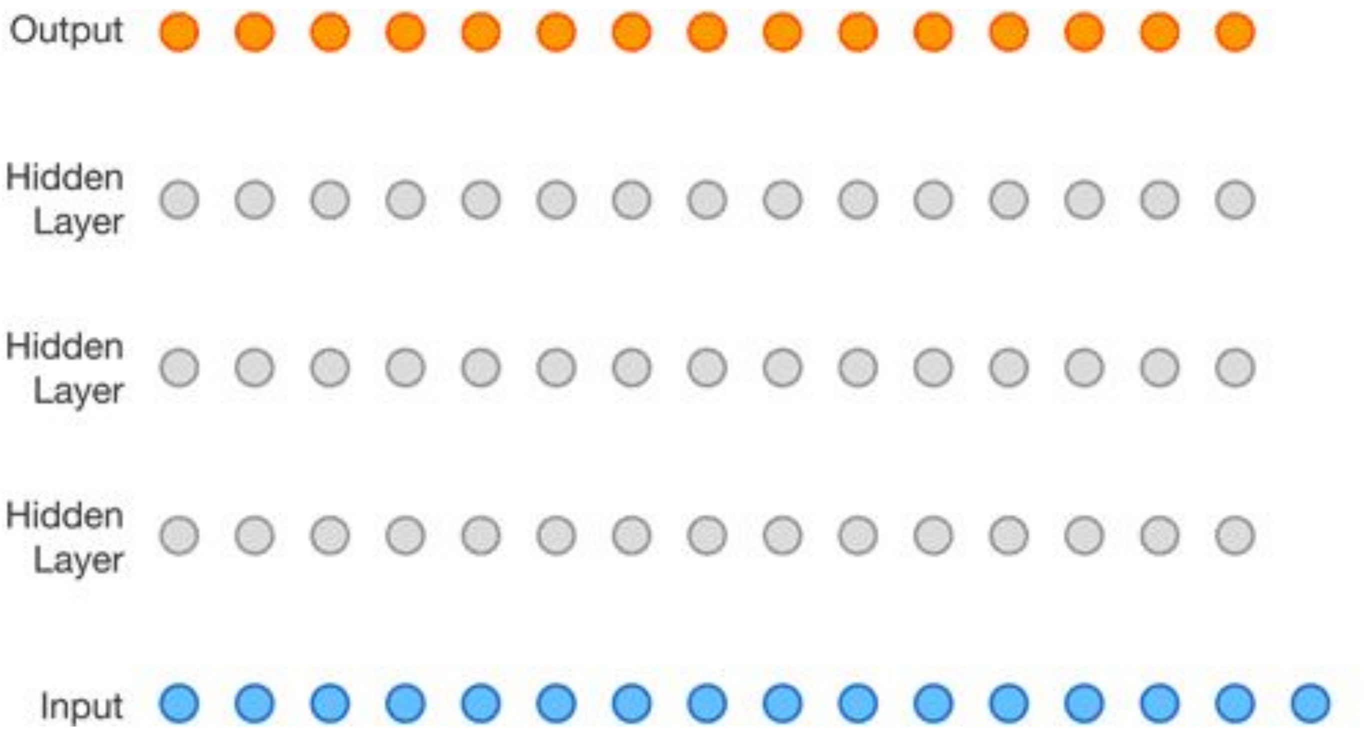
WaveNet Vocoder in VAE-VC



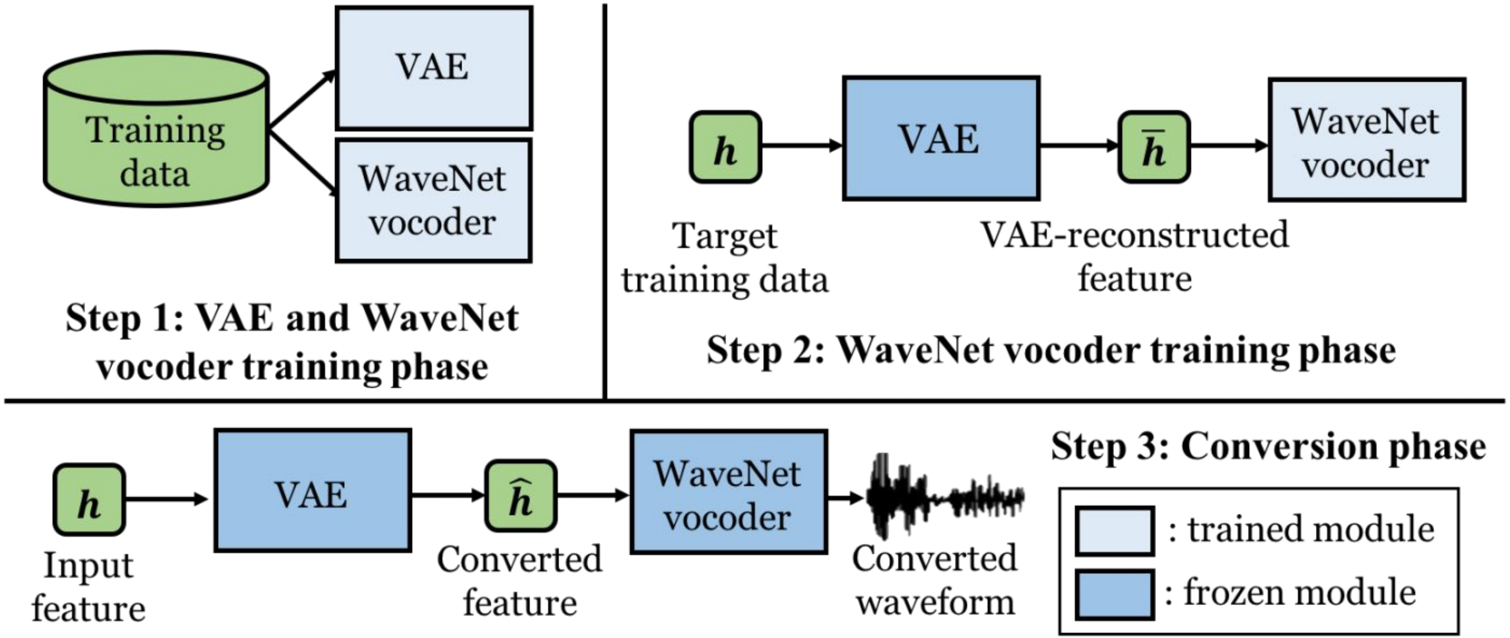
A general framework of WaveNet vocoder in voice conversion.

[Huang et. al. 2019]

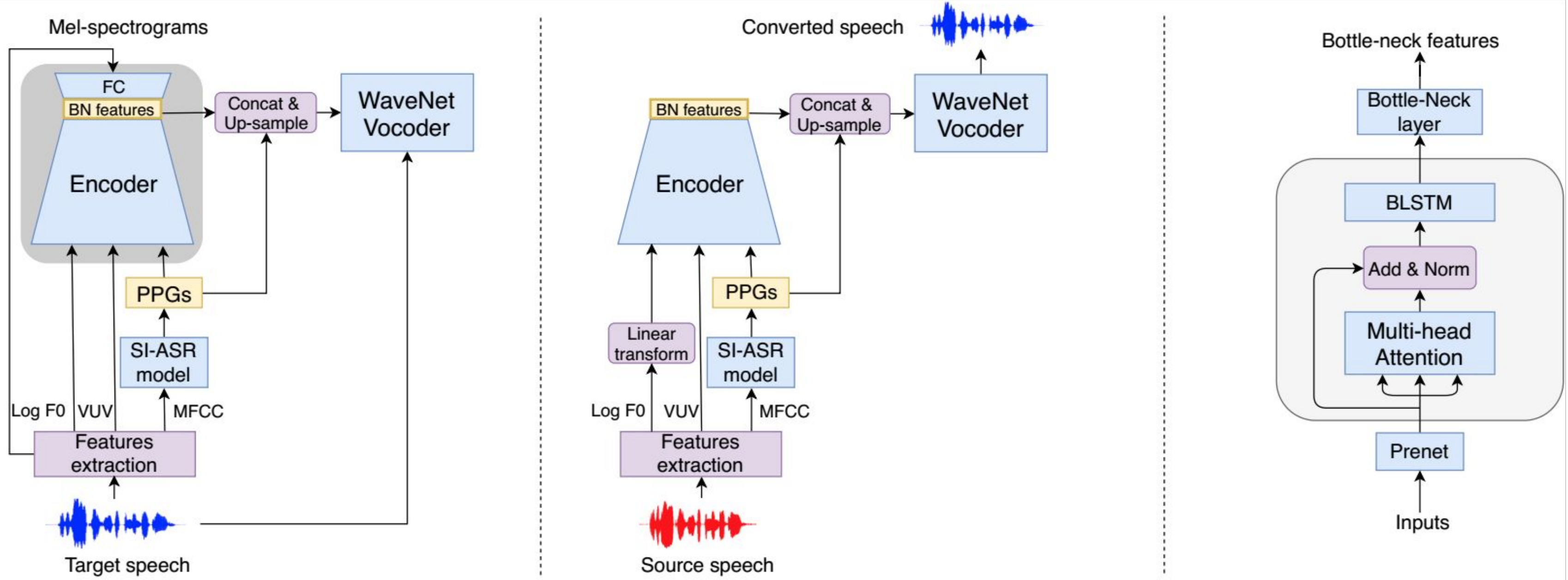
WaveNet



Training Protocol



Jointly Trained Conversion Model and Vocoder



(a) Training stage (b) Conversion stage (c) Encoder module

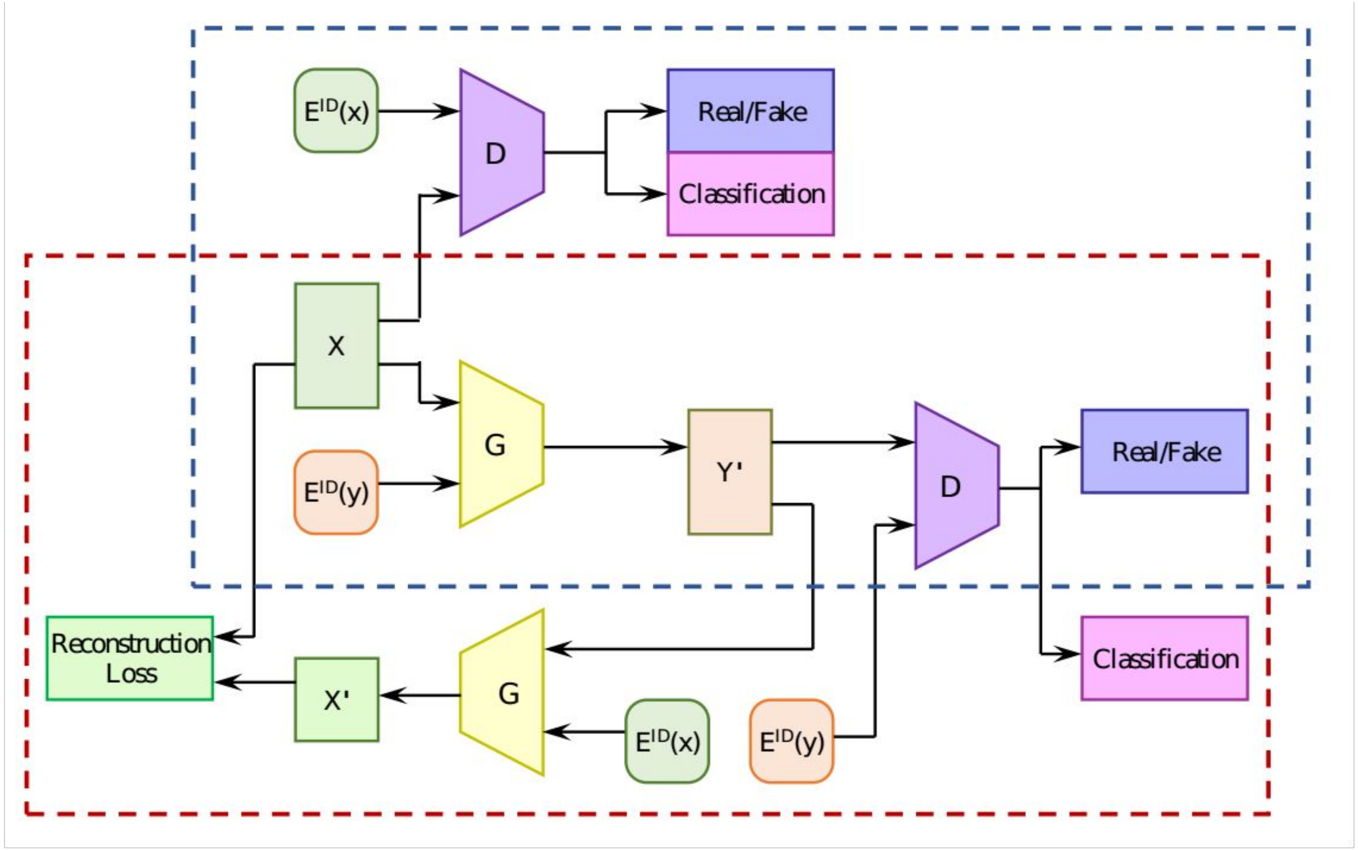
[Liu et. al. 2019]

Zero-shot/Few-shot VC

Zero-shot/Few-shot VC

- The target speaker is unseen in training dataset or both source and target speakers are unseen in the training dataset.
- 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 would be a weighted combination of the timbres the speakers seen in the dataset.

Zero-shot StarGAN VC



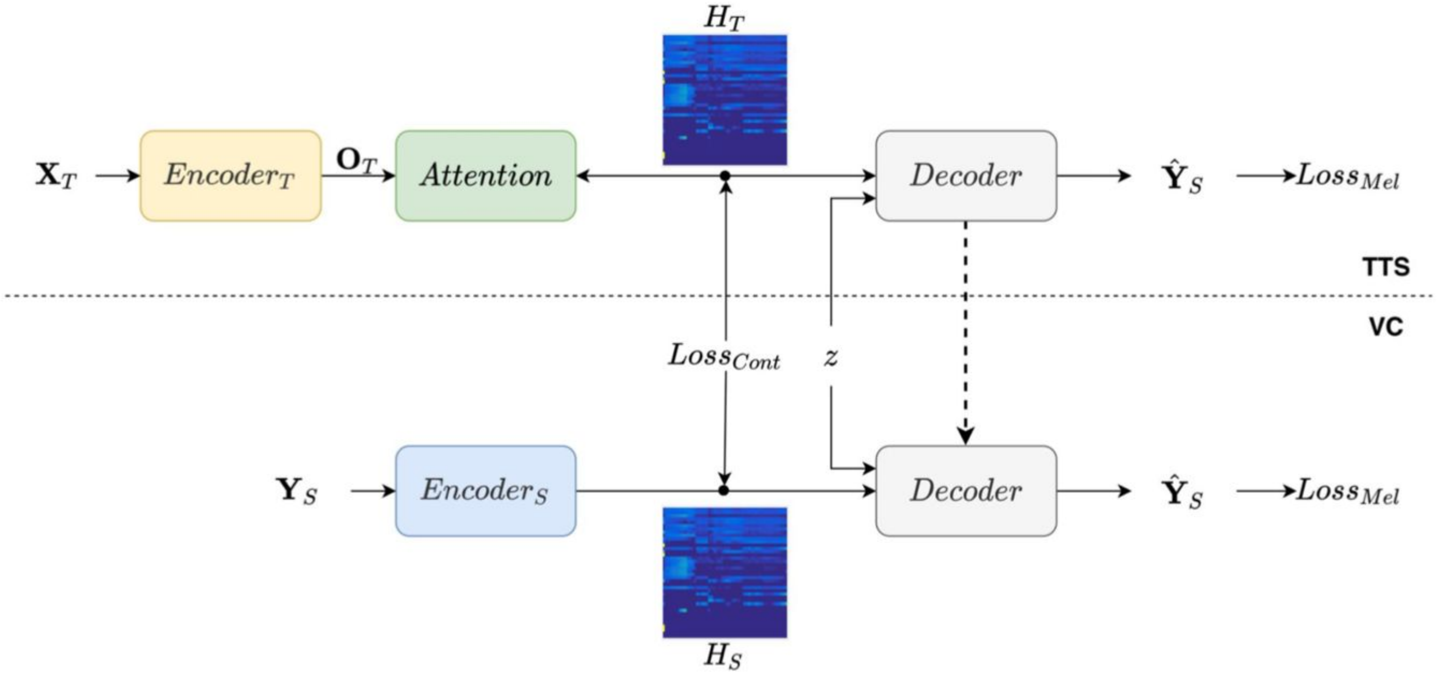
[Wang et. al. 2020]

Text-to-Speech Synthesis to Voice Conversion

TTS to VC

- 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]

TTS to VC



[Zhang et. al. 2021]

Bibliography

- I. Goodfellow et al. “Generative adversarial nets,” in Proc. NIPS, 2014.
- Arjovsky et al. “Wasserstein GAN,” arXiv preprint arXiv:1701.07875, 2017.
- I. Gulrajani et al., “Improved training of Wasserstein GANs,” in Advances in Neural Information Processing Systems, 2017, pp. 5767–5777.
- Hung-yi Lee, and Yu Tsao. ”Generative Adversarial Network and its Applications to Speech Signal and Natural Language Processing.” tutorial in ICASSP, 2018.
- 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.
- A. Kain and M. W. Macon. Spectral voice conversion for text-to-speech synthesis. Proc. IEEE ICASSP, pp. 285–288, 1998.
- T. Kaneko, H. Kameoka. CycleGAN-VC: non-parallel voice conversion using cycle-consistent adversarial networks. Proc. EUSIPCO, pp. 2114–2118, 2018.
- H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo. StarGAN-VC: non-parallel many-to-many voice conversion using star generative adversarial networks. Proc. IEEE SLT, pp. 266–272, 2018.
- Kaneko, T., Kameoka, H., Tanaka, K. and Hojo, N., 2019. StarGAN-VC2: Rethinking Conditional Methods for StarGAN-Based Voice Conversion}}. Proc. Interspeech 2019, pp. 679-683.

Bibliography

- Gao, Yang, Rita Singh, and Bhiksha Raj. "Voice impersonation using generative adversarial networks." In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2506-2510. IEEE, 2018.
- 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.
- 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.
- Pasini, Marco. "Melgan-vc: Voice conversion and audio style transfer on arbitrarily long samples using spectrograms." *arXiv preprint arXiv:1910.03713* (2020).
- Chen, Mingjie, and Thomas Hain. "Unsupervised acoustic unit representation learning for voice conversion using wavenet auto-encoders." *arXiv preprint arXiv:2008.06892* (2020).
- 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.
- Chou, Ju-chieh, and Hung-Yi Lee. "One-Shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization}}." *Proc. Interspeech 2019* (2019): 664-668.

Bibliography

- M. Abe, S. Nakamura, K. Shikano, H. Kuwabara. Voice conversion through vector quantization. J. Acoust. Soc. Jpn (E), Vol. 11, No. 2, pp. 71–76, 1990.
- Y. Stylianou, O. Cappe, E. Moulines. Continuous probabilistic transform for voice conversion. IEEE Trans. Speech & Audio Process., Vol. 6, No. 2, pp. 131–142, 1998.
- L. Sun, K. Li, H. Wang, S. Kang, H.M. Meng. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. Proc. IEEE ICME, 6 pages, 2016.
- T. Toda, A.W. Black, K. Tokuda. Voice conversion based on maximum likelihood estimation of spectral parameter trajectory. IEEE Trans. Audio, Speech & Lang. Process., Vol. 15, No. 8, pp. 2222–2235, 2007.
- http://spcc.csd.uoc.gr/SPCC2019/Lectures/SPCC2019_VC_Lecture_TomokiTODA.pdf
- Pantazis et al. , ” https://www.csd.uoc.gr/~spcc/ ”


Спасибо
Gracias
شكرا
Obrigado
Спасибо
Dank U
Grazie
Ευχαριστώ
Danke
Ευχαριστώ
شكرا
Merci
Ngiyabonga
Dank U
Diolch
Obrigado
Thank You
Ngiyabonga
Tack
Merci
Ευχαριστώ
Dzięki
Danke
Grazie
תודה
Dzięki
謝謝
Dank U
Terima Kasih
Diolch
Grazie
Merci
Tack
Tack
감사합니다
Diolch