Chapter 8

Supplementary

8.1 Supplementary for Chapter 3

8.1.1 Network Expressiveness in Variational Inference

In Section 3.3.2, when analyze the properties of the vCLUB estimator, we claim a reasonable assumption that with high expressiveness of the neural network $q_\theta(\boldsymbol{y}|\boldsymbol{x})$, we can achieve $D_{\mathrm{KL}}(p(\boldsymbol{y}|\boldsymbol{x})\|q_\theta(\boldsymbol{y}|\boldsymbol{x})) < \varepsilon$. Here we provide a analysis under the scenario that the conditional distribution is a Gaussian distribution, $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}^*(\boldsymbol{x}), \mathbf{I})$. The variational approximation $q_\theta(\boldsymbol{y}|\boldsymbol{x})$ is parameterized by $q_\theta(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_\theta(\boldsymbol{x}), \mathbf{I})$.

Then training samples pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ can be treated as $(\boldsymbol{x}_i, \boldsymbol{\mu}^*(\boldsymbol{x}_i) + \boldsymbol{\xi}_i)$, where $\boldsymbol{\xi}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = \log \prod_{d=1}^{D} [\frac{1}{\sqrt{2\pi}} e^{(y^{(d)} - \mu^{*(d)}(\boldsymbol{x}))^2/2}] = -\frac{D}{2}\log(2\pi) - \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\mu}^*(\boldsymbol{x})\|^2,$$

$$\log q_\theta(\boldsymbol{y}|\boldsymbol{x}) = \log \prod_{d=1}^{D} [\frac{1}{\sqrt{2\pi}} e^{(y^{(d)} - \mu_\theta^{(d)}(\boldsymbol{x}))^2/2}] = -\frac{D}{2}\log(2\pi) - \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\mu}_\theta(\boldsymbol{x})\|^2.$$

The log-ratio between $p(\boldsymbol{y}_i|\boldsymbol{x}_i)$ and $q_\theta(\boldsymbol{y}_i|\boldsymbol{x}_i)$ is

$$\log \frac{p(\boldsymbol{y}_i|\boldsymbol{x}_i)}{q_\theta(\boldsymbol{y}_i|\boldsymbol{x}_i)} = \log p(\boldsymbol{y}_i|\boldsymbol{x}_i) - \log q_\theta(\boldsymbol{y}_i|\boldsymbol{x}_i) = [\boldsymbol{\mu}^*(\boldsymbol{x}_i) - \boldsymbol{\mu}_\theta(\boldsymbol{x}_i)]^T[\boldsymbol{y}_i - \boldsymbol{\mu}_\theta(\boldsymbol{x}_i) + \boldsymbol{\xi}_i].$$

We further assume $\|\boldsymbol{\mu}^*(\boldsymbol{x}) - \boldsymbol{\mu}_\theta(\boldsymbol{x})\| < A$ is bounded. Then

$$|\log p(\boldsymbol{y}_i|\boldsymbol{x}_i) - \log q_\theta(\boldsymbol{y}_i|\boldsymbol{x}_i)| < A\|\boldsymbol{y}_i - \boldsymbol{\mu}_\theta(\boldsymbol{x}_i) + \boldsymbol{\xi}_i\|.$$

Denote a loss function $l(\boldsymbol{\mu}_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i) = \|\boldsymbol{y}_i - \boldsymbol{\mu}_\theta(\boldsymbol{x}_i) + \boldsymbol{\xi}_i\|$. With all reasonable assumptions in [HLY19], and applying the Theorem 5.1 in [HLY19], we know that

when the number of samples $n \to \infty$, the expected error $\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[l(\boldsymbol{\mu}_\theta(\boldsymbol{x}), \boldsymbol{y})] \to \infty$ with probability $1 - \delta$.

$$D_{\mathrm{KL}}(p(\boldsymbol{y}|\boldsymbol{x})\|q_\theta(\boldsymbol{y}|\boldsymbol{x})) = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\log p(\boldsymbol{y}|\boldsymbol{x}) - \log q_\theta(\boldsymbol{y}|\boldsymbol{x})] < A \cdot \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[l(\boldsymbol{\mu}_\theta(\boldsymbol{x}), \boldsymbol{y})].$$

Therefore, when given a small number $\varepsilon > 0$, having the sample size n large enough, we can guarantee that $D_{\mathrm{KL}}(p(\boldsymbol{y}|\boldsymbol{x})\|q_\theta(\boldsymbol{y}|\boldsymbol{x}))$ is smaller than ε.

8.1.2 Properties of Variational Upper Bounds

In the Section 3.2, we introduce two variational MI upper bounds with neural network approximation $q_\theta(\boldsymbol{y}|\boldsymbol{x})$ to $p(\boldsymbol{y}|\boldsymbol{x})$:

$$\mathcal{I}_{\mathrm{vVUB}}(\boldsymbol{x};\boldsymbol{y}) = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\left[\log \frac{q_\theta(\boldsymbol{y}|\boldsymbol{x})}{r(\boldsymbol{y})}\right],$$

$$\mathcal{I}_{\mathrm{vL1Out}}(\boldsymbol{x};\boldsymbol{y}) = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\left[\log \frac{q_\theta(\boldsymbol{y}_i|\boldsymbol{x}_i)}{\frac{1}{N-1}\sum_{j\neq i} q_\theta(\boldsymbol{y}_i|\boldsymbol{x}_j)}\right]\right].$$

With the neural approximation $q_\theta(\boldsymbol{y}|\boldsymbol{x})$, $\mathcal{I}_{\mathrm{vVUB}}$ and $\mathcal{I}_{\mathrm{vL1Out}}$ no longer guarantee to be the MI upper bounds. However, both of the two estimators have good properties with a good approximation $q_\theta(\boldsymbol{y}|\boldsymbol{x})$.

Theorem 8.1.1. *If $q_\theta(\boldsymbol{y}|\boldsymbol{x})$ satisfies $KL(p(\boldsymbol{y}|\boldsymbol{x})\|q_\theta(\boldsymbol{y}|\boldsymbol{x})) \leq KL(p(\boldsymbol{y})\|r(\boldsymbol{y}))$, then $\mathcal{I}(\boldsymbol{x};\boldsymbol{y}) \leq \mathcal{I}_{vVUB}(\boldsymbol{x};\boldsymbol{y})$.*

Proof of Theorem 8.1.1. With the conditional $\mathrm{KL}(p(\boldsymbol{y}|\boldsymbol{x})\|q_\theta(\boldsymbol{y}|\boldsymbol{x})) \leq \mathrm{KL}(p(\boldsymbol{y})\|r(\boldsymbol{y}))$,

$$\begin{aligned}\mathcal{I}(\boldsymbol{x};\boldsymbol{y}) =& \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\left[\log \frac{p(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y})}\right] = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\left[\log\left(\frac{p(\boldsymbol{y}|\boldsymbol{x})}{q_\theta(\boldsymbol{y}|\boldsymbol{x})}\cdot\frac{q_\theta(\boldsymbol{y}|\boldsymbol{x})}{r(\boldsymbol{y})}\cdot\frac{r(\boldsymbol{y})}{p(\boldsymbol{y})}\right)\right] \\ =& \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\left[\log \frac{q_\theta(\boldsymbol{y}|\boldsymbol{x})}{r(\boldsymbol{y})}\right] + \mathrm{KL}(p(\boldsymbol{y}|\boldsymbol{x})\|q_\theta(\boldsymbol{y}|\boldsymbol{x})) \\ & - \mathrm{KL}(p(\boldsymbol{y})\|r(\boldsymbol{y})) \leq \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\left[\log \frac{q_\theta(\boldsymbol{y}|\boldsymbol{x})}{r(\boldsymbol{y})}\right].\end{aligned}$$

□

Theorem 8.1.2. *Given $N-1$ samples $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N-1}$ from the marginal $p(\boldsymbol{x})$, If*

$$KL(p(\boldsymbol{y}|\boldsymbol{x})\|q_\theta(\boldsymbol{y}|\boldsymbol{x})) \leq \mathbb{E}_{\boldsymbol{x}_i\sim p(\boldsymbol{x})}\left[KL\left(p(\boldsymbol{y})\|\frac{1}{N-1}\sum_{i=1}^{N-1} q_\theta(\boldsymbol{y}|\boldsymbol{x}_i)\right)\right],$$

then $\mathcal{I}(\boldsymbol{x};\boldsymbol{y}) \leq \mathcal{I}_{vL1Out}(\boldsymbol{x};\boldsymbol{y})$.

Proof. Assume we have N sample pairs $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ drawn from $p(\boldsymbol{x}, \boldsymbol{y})$, then

$$\mathcal{I}(\boldsymbol{x};\boldsymbol{y}) = \mathbb{E}_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\sim p(\boldsymbol{x},\boldsymbol{y})}\left[\frac{1}{N}\sum_{i=1}^{N}\left[\log\frac{p(\boldsymbol{y}_i|\boldsymbol{x}_i)}{p(\boldsymbol{y}_i)}\right]\right]$$

$$=\mathbb{E}_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\sim p(\boldsymbol{x},\boldsymbol{y})}\left[\frac{1}{N}\sum_{i=1}^{N}\left[\log\left(\frac{p(\boldsymbol{y}_i|\boldsymbol{x}_i)}{q_\theta(\boldsymbol{y}_i|\boldsymbol{x}_i)}\cdot\frac{q_\theta(\boldsymbol{y}_i|\boldsymbol{x}_i)}{\frac{1}{N-1}\sum_{j\neq i} q_\theta(\boldsymbol{y}_i|\boldsymbol{x}_j)}\cdot\frac{\frac{1}{N-1}\sum_{j\neq i} q_\theta(\boldsymbol{y}_i|\boldsymbol{x}_j)}{p(\boldsymbol{y}_i)}\right)\right]\right]$$

$$=D_{\mathrm{KL}}(p(\boldsymbol{y}|\boldsymbol{x})\|q_\theta(\boldsymbol{y}|\boldsymbol{x})) + \mathcal{I}_{\mathrm{vVUB}}(\boldsymbol{x};\boldsymbol{y}) - \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}\left(p(\boldsymbol{y})\|\frac{1}{N-1}\sum_{j\neq i} q_\theta(\boldsymbol{y}|\boldsymbol{x}_j)\right)\right].$$

Apply the condition in Theorem 8.1.2 to each $N-1$ combination of $\{\boldsymbol{x}_j\}_{j\neq i}$, we conclude $\mathcal{I}(\boldsymbol{x};\boldsymbol{y}) \leq \mathcal{I}_{\mathrm{vL1Out}}(\boldsymbol{x};\boldsymbol{y})$. □

Theorem 8.1.1 and Theorem 8.1.2 indicate that if the approximation $q_\theta(\boldsymbol{y}|\boldsymbol{x})$ is good enough, the estimators $\mathcal{I}_{\mathrm{vVUB}}$ and $\mathcal{I}_{\mathrm{vL1Out}}$ can remain as MI upper bounds. Based on the analysis in Section 8.1.1, when implemented with neural networks, the approximation can be far more accurate to preserve the variational estimators as MI upper bounds.

8.1.3 Detailed Experiment Setups

MI estimation for Gaussian distribution: All the MI lower bounds require learning of a value function $f(\boldsymbol{x}, \boldsymbol{y})$; all the upper bounds require learning of a network approximation $q_\theta(\boldsymbol{y}|\boldsymbol{x})$. To make fair comparison, we set the value function and the neural approximation with one hidden layer and the same hidden units. For Gaussian

setup, the number of hidden units is 20; for Cubic setup, the number of hidden units is 40. On the top of hidden layer outputs, we add the ReLU activation function. The learning rate for all estimators is set to 1×10^{-4}.

MI estimation for Bernoulli distribution: A two-dimensional joint Bernoulli distribution is parameterized by $\begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}$, with $\sum_{i,j} p_{ij} = 1$ and $0 \leq p_{ij} \leq 1$. the mutual information between the two dimensions can be calculated by $\sum_{i,j} p_{ij} \log \frac{p_{ij}}{(p_{i0}+p_{i1})(p_{0j}+p_{1j})}$. We obtain the true parameter value for a target MI value by using gradient descent. The p_{ij} values are implemented by the output of a Softmax function of 4 logits. The neural approximation is implemented by two-layer MLP with hidden size 50, The learning rate is 5×10^{-5}. The batch size is 10.

Information Bottleneck: For the experiment on information bottleneck, we follow the setup from [AFDM17]. The parameters $\mu_\sigma(\boldsymbol{x})$ and $\Sigma_\sigma(\boldsymbol{x})$ are the output from a MLP with layers $784 \to 1024 \to 1024 \to 2K$, where K is the size of the bottleneck. We set $K = 256$. For the variational classifier to implement the Barber-Agakov MI lower bound, the structure is set to a one-layer MLP. The batch size is 100. We set our learning rate to 10^{-4}, with an exponential decay rate of 0.97 and a decay step of 1200.

Domain Adaptation:

The network is constructed as follows. Both feature extractors (*i.e.*, E_c and E_d) are nine-layer convolutional neural network with leaky ReLU non-linearities. The content classifier C and the domain discriminator D are a one-layer and a two-layer MLPs, respectively. Images from each domain are normalized using Gaussian normalization.

8.1.4 Numerical Results of MI estimation

Numerical Results of MI estimation are shown in table 8.2.

Table 8.1: Model architecture for the unsupervised domain adaptation experiments.

Classifier C	Discriminator D	Extractor (both E_c and E_d)
Content feature $\boldsymbol{z}_c^s$	Domain feature $\boldsymbol{z}_d$	Input data $\boldsymbol{x}$
MLP output $C(\boldsymbol{z}_c^s)$ with shape 10	MLP, 64 ReLU MLP output $D(\boldsymbol{z}_d)$ with shape 2	3×3 conv. 64 lReLU, stride 1 3×3 conv. 64 lReLU, stride 1 3×3 conv. 64 lReLU, stride 1 2×2 max pool, stride 2, dropout, $p = 0.5$, Gaussian noise, $\sigma = 1$ 3×3 conv. 64 lReLU, stride 1 3×3 conv. 64 lReLU, stride 1 3×3 conv. 64 lReLU, stride 1 2×2 max pool, stride 2, dropout, $p = 0.5$, Gaussian noise, $\sigma = 1$ 3×3 conv. 64 lReLU, stride 1 3×3 conv. 64 lReLU, stride 1 3×3 conv. 64 lReLU, stride 1 global average pool, output feature with shape 64

Table 8.2: MSE of MI estimation

	Gaussian					Cubic				
MI	2	4	6	8	10	2	4	6	8	10
VUB	3.85	15.33	34.37	61.25	95.70	2.09	10.38	25.56	47.84	77.59
NWJ	1.67	7.20	17.46	33.26	55.34	1.10	5.54	14.68	30.25	51.07
MINE	1.61	6.66	16.01	29.60	49.87	1.53	6.58	17.4	34.20	59.46
NCE	0.59	2.85	8.56	19.66	37.79	0.45	1.89	6.70	17.48	35.86
L1Out	**0.13**	**0.11**	0.75	4.65	17.08	2.30	5.58	8.92	8.27	7.19
CLUB	0.15	0.12	**0.70**	**4.53**	**16.57**	**2.22**	**5.89**	8.25	**8.23**	**6.93**
CLUBSample	0.38	0.44	1.31	5.30	17.63	2.37	**5.89**	**8.07**	8.87	7.54

8.2 Supplementary for Chapter 4

8.2.1 Proofs of Theorems

Proof of Theorem 5.3.3. First, we show that

$$\mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c})] - \mathbb{E}_{p(\boldsymbol{s})p(\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c})] \geq \mathcal{I}(\boldsymbol{s};\boldsymbol{c}). \tag{8.1}$$

Calculate the gap Δ between the left-hand side and right-hand side of Eq. (8.1):

$$\begin{aligned}
\Delta =& \mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c})] - \mathbb{E}_{p(\boldsymbol{s})p(\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c})] - \mathcal{I}(\boldsymbol{s};\boldsymbol{c}) \\
=& \mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c})] - \mathbb{E}_{p(\boldsymbol{s})p(\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c})] - \mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}\left[\log p(\boldsymbol{s}|\boldsymbol{c}) - \log p(\boldsymbol{s})\right] \\
=& \mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log p(\boldsymbol{s})] - \mathbb{E}_{p(\boldsymbol{s})}\mathbb{E}_{p(\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c})] \\
=& \mathbb{E}_{p(\boldsymbol{s})}\left[\log p(\boldsymbol{s}) - \mathbb{E}_{p(\boldsymbol{c})}[\log(p(\boldsymbol{s}|\boldsymbol{c})]\right] \\
=& \mathbb{E}_{p(\boldsymbol{s})}\left[\log\left(\mathbb{E}_{p(\boldsymbol{c})}[p(\boldsymbol{s}|\boldsymbol{c})]\right) - \mathbb{E}_{p(\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c})]\right] \geq 0. \qquad \text{(Jensen's Inequality)}
\end{aligned}$$

Therefore, the inequality in Eq. (8.1) holds.

Given sample pairs $\{(\boldsymbol{s}_j, \boldsymbol{c}_j)\}_{j=1}^M \sim p(\boldsymbol{s}, \boldsymbol{c})$, the left-hand side of Eq. (8.1) has an unbiased estimation:

$$\begin{aligned}
&\frac{1}{M}\sum_{j=1}^{M} \mathbb{E}_{(\boldsymbol{s}_j,\boldsymbol{c}_j)\sim p(\boldsymbol{s},\boldsymbol{c})}\left[\log p(\boldsymbol{s}_j|\boldsymbol{c}_j)\right] - \frac{1}{M^2}\sum_{j=1}^{M}\sum_{k=1}^{M} \mathbb{E}_{\boldsymbol{s}_j\sim p(\boldsymbol{s})}\mathbb{E}_{\boldsymbol{c}_k\sim p(\boldsymbol{c})}\left[\log p(\boldsymbol{s}_j|\boldsymbol{c}_k)\right] \\
=& \mathbb{E}\left[\frac{1}{M}\sum_{j=1}^{M}\left[\log p(\boldsymbol{s}_j|\boldsymbol{c}_j) - \frac{1}{M}\sum_{k=1}^{M}\log p(\boldsymbol{s}_j|\boldsymbol{c}_k)\right]\right] = \mathbb{E}\left[\frac{1}{M}\sum_{j=1}^{M} R_j\right],
\end{aligned}$$

which is what we claim in Theorem 3.1. □

Proof of Lower Bounds in Eq. (4.6).

$$\begin{aligned}
\mathcal{I}(\boldsymbol{c};\boldsymbol{x}) &= \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{c})}[\log p(\boldsymbol{x}|\boldsymbol{c}) - \log p(\boldsymbol{x})] = \mathcal{H}(\boldsymbol{x}) + \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{c})}[\log p(\boldsymbol{x}|\boldsymbol{c})] \\
&= \mathcal{H}(\boldsymbol{x}) + \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{c})}[\log p(\boldsymbol{x}|\boldsymbol{c}) - \log q_\phi(\boldsymbol{x}|\boldsymbol{c}) + \log q_\phi(\boldsymbol{x}|\boldsymbol{c})] \\
&= \mathcal{H}(\boldsymbol{x}) + \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{c})}[\log p(\boldsymbol{x}|\boldsymbol{c}) - \log q_\phi(\boldsymbol{x}|\boldsymbol{c})] + \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{c})}[\log q_\phi(\boldsymbol{x}|\boldsymbol{c})] \\
&= \mathcal{H}(\boldsymbol{x}) + D_{\mathrm{KL}}(p(\boldsymbol{x}|\boldsymbol{c})\|q_\phi(\boldsymbol{x}|\boldsymbol{c})) + \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{c})}[\log q_\phi(\boldsymbol{x}|\boldsymbol{c})] \\
&\geq H(\boldsymbol{x}) + \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{c})}[\log q(\boldsymbol{x}|\boldsymbol{c})].
\end{aligned}$$

The inequality is based on the fact that the KL-divergence is always non-negative. The lower bound for $\mathcal{I}(\boldsymbol{s}; y)$ can be also derived in the similar way. □

8.2.2 Detailed Experimental Setups

We set the dimension of style embedding to be smaller than the content embedding, because the content carries more information than the style of sentences. The hyper-parameter β in our loss function is a formal expression of re-weighting the two objectives of disentanglement and autoencoding. In practice, we vary it from 0 to 1 with step 0.1 during the first 10 training epochs. At the beginning of the training, the output latent embeddings are not representative enough. Therefore, we choose a small weight on the disentanglement term to avoid obstructing the learning of representative embeddings. After the latent embedding is sufficiently trained, which can successfully reconstruct the input sentences, we slowly enlarge β for the disentanglement. After β reaches 1, we fix it until all the training epochs are finished.

8.2.3 Sample-based Embedding Divergences

In this section we introduce the implementation details of the calculation about label-embedding correlation. As mentioned in Section 5.4 , the distribution divergence between $p(\boldsymbol{c}|y = 0)$ and $p(\boldsymbol{c}|y = 1)$ measures the correlation between content embeddings and style labels. Assume $\boldsymbol{c}_1^{(0)}, \boldsymbol{c}_2^{(0)}, \ldots, \boldsymbol{c}_{N_0}^{(0)} \sim p(\boldsymbol{c}|y = 0)$, and $\boldsymbol{c}_1^{(1)}, \boldsymbol{c}_2^{(1)}, \ldots, \boldsymbol{c}_{N_1}^{(1)} \sim p(\boldsymbol{c}|y = 1)$, then the four metrics MAD, ED, WD, MMD are calculated based on the two groups of samples. With a ground distance $d(\cdot, \cdot)$, the implementaion of the above four metrics are demonstrated in following:

$$D_{\text{MAD}} = d(\frac{1}{N_0}\sum_{i=1}^{N_0}\boldsymbol{c}_i^{(0)}, \frac{1}{N_1}\sum_{j=1}^{N_1}\boldsymbol{c}_j^{(1)}). \tag{8.2}$$

$$D_{\text{ED}} = \frac{2}{N_0 N_1}\sum_{i=1}^{N_0}\sum_{j=1}^{N_1} d(\boldsymbol{c}_i^{(0)}, \boldsymbol{c}_j^{(1)}) - \frac{1}{N_0^2}\sum_{i=1}^{N_0}\sum_{j=1}^{N_0} d(\boldsymbol{c}_i^{(0)}, \boldsymbol{c}_j^{(0)}) - \frac{1}{N_1^2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_1} d(\boldsymbol{c}_i^{(1)}, \boldsymbol{c}_j^{(1)}) \tag{8.3}$$

$$D_{\mathrm{WD}} = \min_{p_{ij}} \sum_{i=1}^{N_0} \sum_{j=1}^{N_1} p_{ij}\ d(\boldsymbol{c}_i^{(0)}, \boldsymbol{c}_j^{(1)}) \quad s.t. \sum_{i=1}^{N_0} p_{ij} = \frac{1}{N_1}, \quad \sum_{j=1}^{N_1} p_{ij} = \frac{1}{N_0}. \tag{8.4}$$

$$D_{\mathrm{MMD}} = \frac{1}{N_0^2} \sum_{i=1}^{N_0} \sum_{j=1}^{N_0} K(\boldsymbol{c}_i^{(0)}, \boldsymbol{c}_j^{(0)}) + \frac{1}{N_1^2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_1} K(\boldsymbol{c}_i^{(1)}, \boldsymbol{c}_j^{(1)}) - \frac{2}{N_0 N_1} \sum_{i=1}^{N_0} \sum_{j=1}^{N_1} K(\boldsymbol{c}_i^{(0)}, \boldsymbol{c}_j^{(1)}), \tag{8.5}$$

where $K(\cdot, \cdot)$ is a kernel function. Here we choose $K(\cdot, \cdot)$ from RBF kernel family with bandwidth $w = 1$.

For style embedding, the calculation formats are the same as in above equations. The style embeddings and content embeddings have different dimensions, which leads to the ground metric $d(\cdot, \cdot)$ inconsistent. Therefore, instead of using Euclidean distance, we use the cosine distance as the ground metric.

8.2.4 Details in Representation Quality Evaluation

For style preservation, we pretrain a style classifier on each dataset. The style classifier is built by a one-layer LSTM appended with a multi-head attention layer. The number of the attention head is set to 6. The classifiers reach 95% prediction accuracy on Yelp and 93% prediction accuracy on Personality-Captioning. We input transferred sentences into the classifier and test whether the predicted style label is the same as the target style label.

For human evaluation, we transferred 1000 sentences with randomly selected style labels. After the transferring, we ask 10 human annotators to justify the style label, content preservation and content fluency. The style label is 0 or 1 representing the positive or negative sentiment of the given sentence. The content preservation and the content fluency is scored between 0 to 5. To make the style accuracy compatible

with the other two scores, we scale it into range [0,5]. If the scores from the two annotators have a difference larger than 2, the scores will not be recorded. In this way, we ensure the evaluation criteria of annotators are similar.

8.3 Supplementary for Chapter 5

8.3.1 Proofs

Theorem 8.3.1 (Theorem 3.1)**.** *Let* $\boldsymbol{\mu}_v^{(-ui)} = \frac{1}{N_v}\sum_{k=1}^{N_v} \boldsymbol{s}_{vk}$ *if* $u \neq v$*; and* $\boldsymbol{\mu}_u^{(-ui)} = \frac{1}{N_u-1}\sum_{j\neq i} \boldsymbol{s}_{uj}$*. Then,*

$$\mathcal{I}(\boldsymbol{u};\boldsymbol{s}) \geq \mathbb{E}\Big[\frac{1}{N}\sum_{u=1}^{M}\sum_{i=1}^{N_u}\Big[-\|\boldsymbol{s}_{ui}-\boldsymbol{\mu}_u^{(-ui)}\|^2 - \frac{e^{-1}}{N}\sum_{v=1}^{M}[N_v \exp(-\|\boldsymbol{s}_{ui}-\boldsymbol{\mu}_v^{(-ui)}\|^2)]\Big]\Big]. \tag{8.6}$$

Proof of Theorem 5.3.1. By the condition in the Theorem, we have the given sample pairs $\{(u,\boldsymbol{s}_{ui})\}_{1\leq u\leq M, 1\leq i\leq N_u}$. Not that each pair of speaker identity and style embedding, $(u,\boldsymbol{s}_{ui})$, can be regarded as a sample from the joint distribution $p(\boldsymbol{u},\boldsymbol{s})$. To clearify the proof, we change the notation of random variables $\boldsymbol{u}$ and $\boldsymbol{s}$ to $\boldsymbol{U}$ and $\boldsymbol{S}$, which are distinct to samples $\{(u,\boldsymbol{s}_{ui})\} \sim p(\boldsymbol{U},\boldsymbol{S})$.

For a sample pair $(u,\boldsymbol{s}_{ui})$, by the NWJ lower bound, we have

$$\begin{aligned}
\mathcal{I}(\boldsymbol{U};\boldsymbol{S}) \geq& \mathbb{E}_{p(\boldsymbol{U},\boldsymbol{S})}[f(\boldsymbol{U},\boldsymbol{S})] - e^{-1}\mathbb{E}_{p(\boldsymbol{U})p(\boldsymbol{S})}[e^{f(\boldsymbol{U},\boldsymbol{S})}] \\
=& \mathbb{E}_{p(\boldsymbol{S})}\Big[\mathbb{E}_{p(\boldsymbol{U}|\boldsymbol{S})}[f(\boldsymbol{U},\boldsymbol{S})] - e^{-1}\mathbb{E}_{p(\boldsymbol{U})}[e^{f(\boldsymbol{U},\boldsymbol{S})}]\Big] \\
=& \mathbb{E}_{\boldsymbol{s}_{ui}\sim p(\boldsymbol{S})}\Big[\mathbb{E}_{p(\boldsymbol{U}|\boldsymbol{S}=\boldsymbol{s}_{ui})}[f(\boldsymbol{U},\boldsymbol{S}=\boldsymbol{s}_{ui})] - e^{-1}\mathbb{E}_{p(\boldsymbol{U})}[e^{f(\boldsymbol{U},\boldsymbol{S}=\boldsymbol{s}_{ui})}]\Big],
\end{aligned} \tag{8.7}$$

with a score function $f(\boldsymbol{U},\boldsymbol{S})$. Given $\boldsymbol{S}=\boldsymbol{s}_{ui}$, $\mathbb{E}_{p(\boldsymbol{U}|\boldsymbol{S}=\boldsymbol{s}_{ui})}[f(\boldsymbol{U},\boldsymbol{S}=\boldsymbol{s}_{ui})]$ has an unbiased estimation $f(\boldsymbol{U}=u,\boldsymbol{S}=\boldsymbol{s}_{ui})$; $\mathbb{E}_{p(\boldsymbol{U})}[e^{f(\boldsymbol{U},\boldsymbol{S}=\boldsymbol{s}_{ui})}]$ has an unbiased estimation

by taking average of all possible values $v \sim p(\boldsymbol{U})$ in samples $\{(u, \boldsymbol{s}_{ui})\}$,

$$\mathbb{E}_{p(\boldsymbol{U})}[e^{f(\boldsymbol{U},\boldsymbol{S}=\boldsymbol{s}_{ui})}] = \mathbb{E}\Big[\sum_{v=1}^{M} \frac{N_v}{N} e^{f(\boldsymbol{U}=v,\boldsymbol{S}=\boldsymbol{s}_{ui})}\Big]. \tag{8.8}$$

With the two estimations, (8.7) becomes

$$\mathcal{I}(\boldsymbol{U};\boldsymbol{S}) \geq \mathbb{E}\Big[f(u,\boldsymbol{s}_{ui}) - e^{-1}\sum_{v=1}^{M} \frac{N_v}{N} e^{f(v,\boldsymbol{s}_{ui})}\Big]. \tag{8.9}$$

Specifically, we select score function $f(\boldsymbol{U} = v, \boldsymbol{S} = \boldsymbol{s}) = -\|\boldsymbol{s} - \boldsymbol{\mu}_v^{(-ui)}\|^2$, then (8.9) becomes

$$\mathcal{I}(\boldsymbol{U};\boldsymbol{S}) \geq \mathbb{E}[\hat{\mathcal{I}}_{ui}] = \mathbb{E}\Big[-\|\boldsymbol{s}_{ui} - \boldsymbol{\mu}_u^{-(ui)}\|^2 - e^{-1}\sum_{v=1}^{M} \frac{N_v}{N} e^{-\|\boldsymbol{s}_{ui}-\boldsymbol{\mu}_v^{(-ui)}\|^2}\Big]. \tag{8.10}$$

Since the selection of index ui is arbitrary, we take an average on all $\hat{\mathcal{I}}_{ui}$,

$$\begin{aligned}\mathcal{I}(\boldsymbol{U};\boldsymbol{S}) \geq & \frac{1}{N}\sum_{u=1}^{M}\sum_{i=1}^{N_u} \mathbb{E}_{(u,\boldsymbol{s}_{ui})\sim p(\boldsymbol{U},\boldsymbol{S})}[\hat{\mathcal{I}}_{ui}] = \mathbb{E}[\frac{1}{N}\sum_{u=1}^{M}\sum_{i=1}^{N_u}\hat{\mathcal{I}}_{ui}] \\ = & \mathbb{E}\Big[\frac{1}{N}\sum_{u=1}^{M}\sum_{i=1}^{N_u}\Big[-\|\boldsymbol{s}_{ui} - \boldsymbol{\mu}_u^{(-ui)}\|^2 - \frac{e^{-1}}{N}\sum_{v=1}^{M}[N_v \exp(-\|\boldsymbol{s}_{ui} - \boldsymbol{\mu}_v^{(-ui)}\|^2)]\Big]\Big],\end{aligned} \tag{8.11}$$

where the right-hand side of equation (5.7) is derived. □

Theorem 8.3.2 (Theorem 3.2)**.** *Assume that given $\boldsymbol{s} = \boldsymbol{s}_u$, samples $\{(\boldsymbol{x}_{ui}, \boldsymbol{c}_{ui})\}_{i=1}^{N_u}$ are observed. With a variational distribution $q_\phi(\boldsymbol{x}|\boldsymbol{s},\boldsymbol{c})$, we have $\mathcal{I}(\boldsymbol{x};\boldsymbol{c}|\boldsymbol{s}) \geq \mathbb{E}[\hat{\mathcal{I}}]$, where*

$$\hat{\mathcal{I}} = \frac{1}{N}\sum_{u=1}^{M}\sum_{i=1}^{N_u}\Big[\log q_\phi(\boldsymbol{x}_{ui}|\boldsymbol{c}_{ui},\boldsymbol{s}_u) - \log(\frac{1}{N_u}\sum_{j=1}^{N_u} q_\phi(\boldsymbol{x}_{uj}|\boldsymbol{c}_{ui},\boldsymbol{s}_u))\Big]. \tag{8.12}$$

Proof of Theorem 5.3.2 . Given $\boldsymbol{s} = \boldsymbol{s}_u$, we observe sample pair $\{\boldsymbol{x}_{ui}, \boldsymbol{c}_{ui}\}_{i=1}^{N_u}$. By the InfoNCE lower bound [OLV18], with a score function f, we have

$$\mathcal{I}(\boldsymbol{x}; \boldsymbol{c}|\boldsymbol{s} = \boldsymbol{s}_u) \geq \mathbb{E}\Big[\frac{1}{N_u}\sum_{i=1}^{N_u}\Big[f(\boldsymbol{x}_{ui}, \boldsymbol{c}_{ui}) - \log\left(\frac{1}{N_u}\sum_{j=1}^{N_u} e^{f(\boldsymbol{x}_{uj}, \boldsymbol{c}_{ui})}\right)\Big]\Big]. \tag{8.13}$$

We select $f(\boldsymbol{x}, \boldsymbol{c}) = \log q_\phi(\boldsymbol{x}|\boldsymbol{c}, \boldsymbol{s} = \boldsymbol{s}_u)$, then

$$\mathcal{I}(\boldsymbol{x}; \boldsymbol{c}|\boldsymbol{s} = \boldsymbol{s}_u) \geq \mathbb{E}\Big[\frac{1}{N_u}\sum_{i=1}^{N_u}\Big[\log q_\phi(\boldsymbol{x}_{ui}|\boldsymbol{c}_{ui}, \boldsymbol{s}_u) - \log\left(\frac{1}{N_u}\sum_{j=1}^{N_u} q_\phi(\boldsymbol{x}_{uj}|\boldsymbol{c}_{ui}, \boldsymbol{s}_u)\right)\Big]\Big]. \tag{8.14}$$

Taking expectation of $\boldsymbol{s}$ on both sides, we derive

$$\mathcal{I}(\boldsymbol{x}; \boldsymbol{c}|\boldsymbol{s}) \geq \mathbb{E}\Big[\frac{1}{N}\sum_{u=1}^{M}\sum_{i=1}^{N_u}\Big[\log q_\phi(\boldsymbol{x}_{ui}|\boldsymbol{c}_{ui}, \boldsymbol{s}_u) - \log\left(\frac{1}{N_u}\sum_{j=1}^{N_u} q_\phi(\boldsymbol{x}_{uj}|\boldsymbol{c}_{ui}, \boldsymbol{s}_u)\right)\Big]\Big]. \tag{8.15}$$

□

Theorem 8.3.3 (Theorem 3.3)**.** *If $p(\boldsymbol{s}|\boldsymbol{c})$ provides the conditional distribution between variables $\boldsymbol{s}$ and $\boldsymbol{c}$, then*

$$\mathcal{I}(\boldsymbol{s}; \boldsymbol{c}) \leq \mathbb{E}\Big[\frac{1}{N}\sum_{u=1}^{M}\sum_{i=1}^{N_u}\Big[\log p(\boldsymbol{s}_{ui}|\boldsymbol{c}_{ui}) - \frac{1}{N}\sum_{v=1}^{M}\sum_{j=1}^{N_v}\log p(\boldsymbol{s}_{ui}|\boldsymbol{c}_{vj})\Big]\Big]. \tag{8.16}$$

Proof of Theorem 5.3.3. By the upper bound in Cheng *et al.* [CMS+20], we have

$$\mathcal{I}(\boldsymbol{s}; \boldsymbol{c}) \leq \mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c})] - \mathbb{E}_{p(\boldsymbol{s})p(\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c})]. \tag{8.17}$$

With embedding samples $\{\boldsymbol{s}_{ui}, \boldsymbol{c}_{ui}\}_{1\leq u\leq M, 1\leq i\leq N_u}$, the right-hand side of (8.17) can be estimated by

$$\mathcal{I}(\boldsymbol{s}; \boldsymbol{c}) \leq \mathbb{E}\Big[\frac{1}{N}\sum_{u=1}^{M}\sum_{i=1}^{N_u}\Big[\log p(\boldsymbol{s}_{ui}|\boldsymbol{c}_{ui}) - \frac{1}{N}\sum_{v=1}^{M}\sum_{j=1}^{N_v}\log p(\boldsymbol{s}_{ui}|\boldsymbol{c}_{vj})\Big]\Big]. \tag{8.18}$$

□

Discussion on variational approximation As mentioned in Section 4.3.3, we approximate $p(\boldsymbol{s}|\boldsymbol{c})$ with a variational distribution $q_\theta(\boldsymbol{s}|\boldsymbol{c})$ in equation (5.10), since the closed form of $p(\boldsymbol{s}|\boldsymbol{c})$ is unknown. We claim that with $q_\theta(\boldsymbol{s}|\boldsymbol{c})$ as a good approximation of $p(\boldsymbol{s}|\boldsymbol{c})$, equation (5.10) remains a MI upper bound. We calculate the difference between $\mathcal{I}(\boldsymbol{s};\boldsymbol{c})$ and the approximated version of (8.17):

$$
\begin{aligned}
\Delta :=& \mathcal{I}(\boldsymbol{s};\boldsymbol{c}) - [\mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log q_\theta(\boldsymbol{s}|\boldsymbol{c})] - \mathbb{E}_{p(\boldsymbol{s})p(\boldsymbol{c})}[\log q_\theta(\boldsymbol{s}|\boldsymbol{c})]] \\
=& \mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c}) - \log p(\boldsymbol{s})] - \mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log q_\theta(\boldsymbol{s}|\boldsymbol{c})] + \mathbb{E}_{p(\boldsymbol{s})p(\boldsymbol{c})}\Big[\log q_\theta(\boldsymbol{s}|\boldsymbol{c})\Big] \\
=& \Big[\mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log p(\boldsymbol{s}|\boldsymbol{c})] - \mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log q_\theta(\boldsymbol{s}|\boldsymbol{c})]\Big] - \Big[\mathbb{E}_{p(\boldsymbol{s})}[\log p(\boldsymbol{s})] - \mathbb{E}_{p(\boldsymbol{s})p(\boldsymbol{c})}[\log q_\theta(\boldsymbol{s}|\boldsymbol{c})]\Big] \\
=& \mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log \frac{p(\boldsymbol{s}|\boldsymbol{c})}{q_\theta(\boldsymbol{s}|\boldsymbol{c})}] - \mathbb{E}_{p(\boldsymbol{s})p(\boldsymbol{c})}[\log \frac{p(\boldsymbol{s})}{q_\theta(\boldsymbol{s}|\boldsymbol{c})}] \\
=& \mathrm{KL}(p(\boldsymbol{s}|\boldsymbol{c})\|q_\theta(\boldsymbol{s}|\boldsymbol{c})) - \mathrm{KL}(p(\boldsymbol{s})\|q_\theta(\boldsymbol{s}|\boldsymbol{c})).
\end{aligned}
$$

When $q_\theta(\boldsymbol{s}|\boldsymbol{c})$ is a good approximation to $p(\boldsymbol{s}|\boldsymbol{c})$, the divergence $\mathrm{KL}(p(\boldsymbol{s}|\boldsymbol{c})\|q_\theta(\boldsymbol{s}|\boldsymbol{c}))$ can be smaller than $\mathrm{KL}(p(\boldsymbol{s})\|q_\theta(\boldsymbol{s}|\boldsymbol{c}))$. Then Δ remains negative, which indicates $[\mathbb{E}_{p(\boldsymbol{s},\boldsymbol{c})}[\log q_\theta(\boldsymbol{s}|\boldsymbol{c})] - \mathbb{E}_{p(\boldsymbol{s})p(\boldsymbol{c})}[\log q_\theta(\boldsymbol{s}|\boldsymbol{c})]]$ still be an MI upper bound.

8.3.2 Experiements

More ablation study on bottleneck design

We kept the same bottleneck design as AUTOVC to have a fair comparison for the effectiveness of the proposed disentangled learning scheme. To further provide evidence of effectiveness of IDE-VC , we also conducted an ablation study in which the bottleneck is widened in Table 8.3. Specifically, we use set sampling rate as 4 and conduct experiments under the zero-shot setup. The results demonstrated that the bottleneck design has little impact on the disentanglement

Table 8.3: Ablation study with bottleneck design for zero-shot VST. We set sampling rate as 4. Performance is measured by objective metrics.

	Distance	Verification[%]
AUTOVC	6.59	41
IDE-VC	**6.24**	**80**

ability of the proposed model.

More ablation study on visualization We further provide t-SNE visualization for content embedding from IDE-VC and AUTOVC in Figure 8.1 and Figure 8.2 under same hyperparameter setups. Comparing between the two t-SNE plottings, the content embeddings generated with IDE-VC are more indistinguishable for different speakers than the ones from AUTOVC, which proves that the proposed model has stronger ability to eliminate speaker-related information in content embedding.

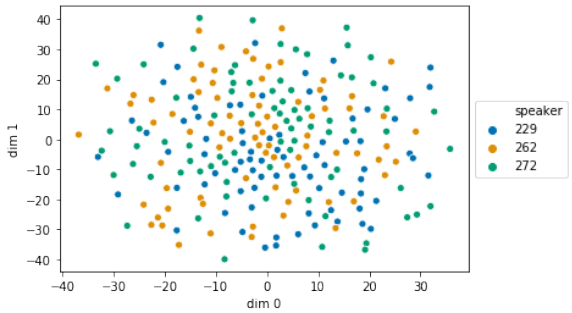

Figure 8.1: t-SNE visualization for content embedding from IDE-VC . The embeddings are extracted from the voice samples of 3 different speakers.

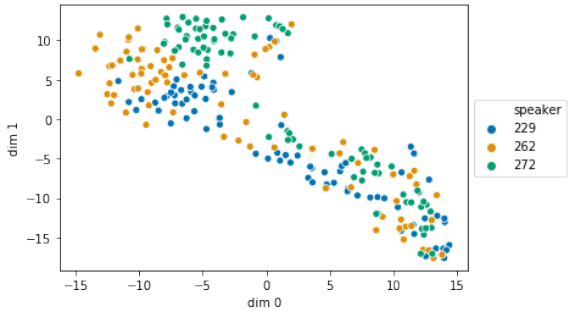

Figure 8.2: t-SNE visualization for content embedding from AUTOVC. The embeddings are extracted from the voice samples of 3 different speakers.

Speaker encoder pretraining Our speaker encoder is pretrained with GE2E loss on a combination of VoxCeleb1 [NCZ17] and Librispeech [PCPK15] datasets, in total of 3549 speakers.

Implementation details In our experiments, we use official implementation of AdaIN-VC[1], AUTOVC[2] and Blow[3]. Specifically, same pretrained speaker encoder is used in AUTOVC [QZC+19] and our model for fair comparison. Blow model is trained with 100 epochs and suggested hyperparameters, the training takes over 10 GPU days on Nvidia V100 in comparison with 1 GPU day on Nvidia Xp for our model. For StarGAN-VC, we use an open source implementation[4], which achieves better performance according to multiple previous works [QZC+19, SPP19]. All above models are trained on all 109 speakers in VCTK dataset, and same splits are used for testing and validation.

For our model, we use loss on validation set to conduct grid search on hyperparameter β, and we use $\beta = 5$ in final experiment. The other hyperparameters are set as the same as in AUTOVC [QZC+19].

Sample speeches We also provide several sample conversed speeches on https://idevc.github.io/.

8.3.3 Evaluation Details

Verification score In details, in Resemblyzer, a pre-trained speaker encoder is provided $\boldsymbol{s} = E_{sr}(\boldsymbol{x})$. The voice profile for each speaker u is first computed as $\boldsymbol{s}_u = \frac{1}{N_t}\sum_n^{N_t} E_{sr}(\boldsymbol{x}_{un})$, in which N_t represents the number of speeches each speaker has in testing set, $\boldsymbol{x}_{un}$ represents the n-th speech of speaker u in testing set. For each speech conversed from i-th speech of speaker u to j-th speech of speaker v, represented as $\hat{\boldsymbol{x}}_{u_i \to v_j}$, the speaker embedding is computed with Resemblyzer: $\hat{\boldsymbol{s}}_{u_i \to v_j} = E_{sr}(\hat{\boldsymbol{x}}_{u_i \to v_j})$.

[1]https://github.com/jjery2243542/adaptive_voice_conversion

[2]https://github.com/auspicious3000/autovc

[3]https://github.com/joansj/blow

[4]https://github.com/liusongxiang/StarGAN-Voice-Conversion

Dot product is used to compute similarity between the speaker embedding and the voice profile. If among all speakers in testing set, the speaker embedding of the conversed speech has highest similarity score with the target speaker's profile $\boldsymbol{s}_v$, we view it as a success conversion. The portion of success conversion among all conversion trials is reported as verification score.

Details in evaluation for zero-shot VST Based on setting in AdaIN-VC [CL19], subjective evaluation is performed on converted voice between male to male, male to female, female to male and female to female speakers. To reduce variance, 3 speakers are selected for each gender, thus, in total 36 pairs of speakers. The speakers of these pairs were unseen during training. Following the setting in AdaIN-VC [CL19], the converted result of each pair was transfered from our proposed model with only one source utterance and one target utterance.

Dynamic Time Wrapping When evaluating the voices generated by neural networks from latent embeddings, the mismatching problem in time alignment occurs due to the time shift and different speaking speeds in the generation. Important voice patterns may be neglected when directly calculating the Euclidean distance between the generated voice and the ground-truth. One effective solution to the sequential time-alignment problem is the Dynamic Time Wrapping (DTW) algorithm [BC94], which has been widely applied in speech recognition and matching [MBE10, Cha12, DNP13]. The DTW algorithm seeks the optimal matching path $\boldsymbol{P}^* \in \mathcal{P}(T, S)$ that minimizes the sequential matching cost between two time series $\boldsymbol{x} = (x^1, x^2, \ldots, x^T)$ and $\boldsymbol{y} = (y^1, y^2, \ldots, y^S)$ (*e.g.*, the purple path in the right of Figure 8.3). A consecutive matching path $\boldsymbol{P} \in \mathcal{P}(T, S)$ denotes a sequence of index pairs $\boldsymbol{P} = (\boldsymbol{p}^1, \boldsymbol{p}^2, \ldots, \boldsymbol{p}^L)$, in which each pair $\boldsymbol{p}^l = (t_l, s_l)$ matches x^{t_l} and y^{s_l} following the time order (*i.e.*, $\boldsymbol{p}^1 = (1, 1)$, $\boldsymbol{p}^L = (T, S)$, $0 \leq t_{l+1} - t_l \leq 1$, and $0 \leq s_{l+1} - s_l \leq 1$). The DTW score is $\mathcal{S}_{\text{DTW}}(\boldsymbol{x}, \boldsymbol{y}) = \min_{\boldsymbol{P} \in \mathcal{P}(T,S)} \sum_{l=1}^{L} d(x^{t_l}, y^{s_l})$,

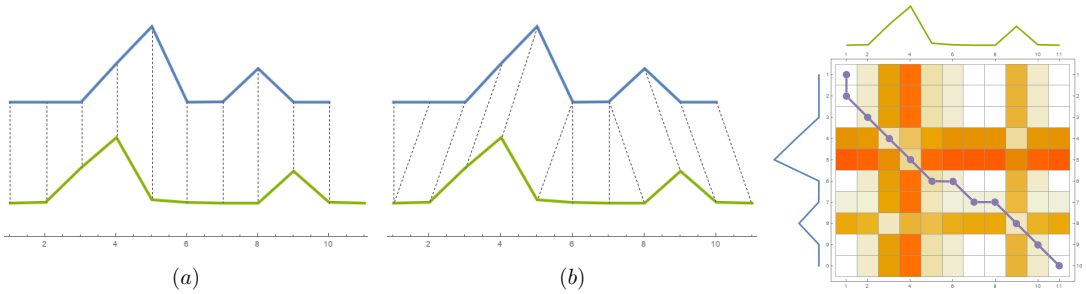

Figure 8.3: Left: (a) The naive Euclidean distance matching between two time series might miss the important voice patterns because of the time shift and different speakers' speaking speeds; (b) DTW automatically find the optimal matching that captures important voice pattens. Right: DTW converts the time sequence matching problem into the minimal cost path searching on the distance matrix.

where $d(\cdot, \cdot)$ is a ground distance measuring the dissimilarity of any two points in time series. The optimization needed for calculating the DTW score can be solved efficiently by dynamic programming.

Human Evaluation Following Wester *et al.* [WWY16], we use the naturalness of the speech and the similarity of the transferred speech to target identity as subjective metrics. Figure 8.4 and Figure 8.5 shows the contents of the two human evaluation webpage layouts respectively.

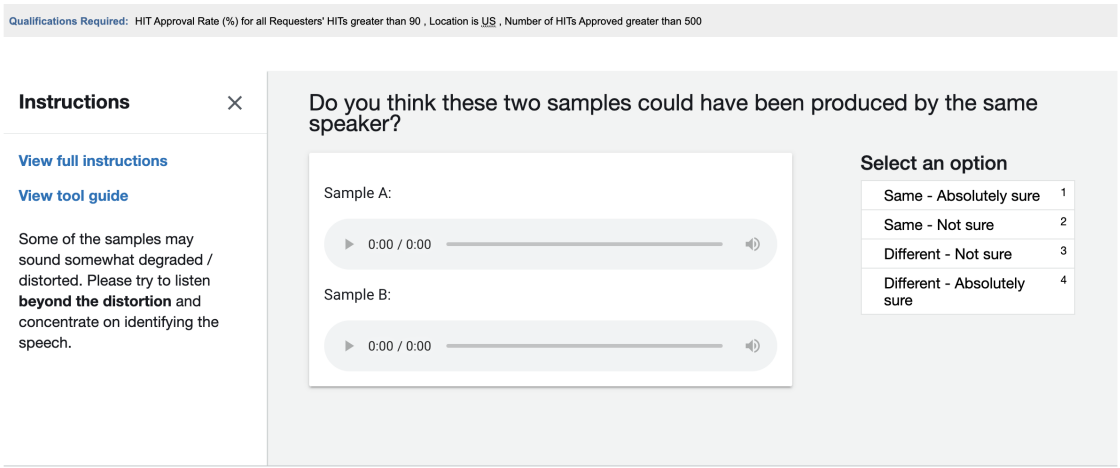

Figure 8.4: Human evaluation: similarity

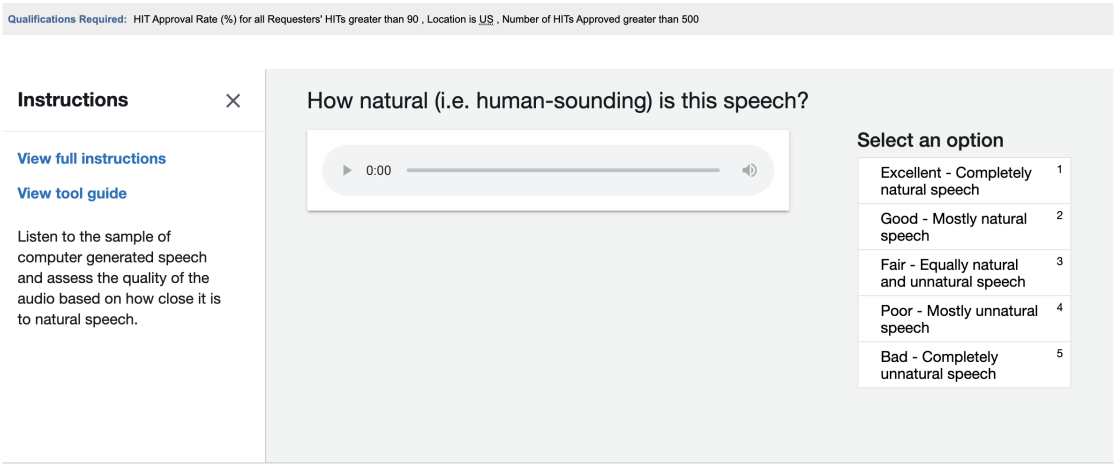

Figure 8.5: Human evaluation: naturalness

8.3.4 Data Processing Inequality

Theorem 8.3.4. *If three variables* $\boldsymbol{x} \to \boldsymbol{y} \to \boldsymbol{z}$ *follow a markov chain, then* $\mathcal{I}(\boldsymbol{x}; \boldsymbol{y}) \geq \mathcal{I}(\boldsymbol{x}; \boldsymbol{z})$.