VoiceMixer: Adversarial Voice Style Mixup

Anonymous Author(s)
Affiliation
Address
email

Abstract

Although recent advances in voice conversion have shown significant improvement, there still remains a gap between the converted voice and target voice. A key factor that maintains this gap is the insufficient decomposition of content and voice style from the source speech. This insufficiency leads to the converted speech containing source speech style or losing source speech content. In this paper, we present VoiceMixer which can effectively decompose and transfer voice style through a novel information bottleneck and adversarial feedback. With self-supervised representation learning, the proposed information bottleneck can decompose the content and style with only a small loss of content information. Also, for adversarial feedback of each information, the discriminator is decomposed into content and style discriminator with self-supervision, then our model achieves better generalization to the voice style of the converted speech. The experimental results show the superiority of our model in disentanglement and transfer performance, and improve audio quality by preserving content information.

1 Introduction

Voice conversion (VC) is the task of transferring the target voice style to the source speech while keeping the content information of the source speech. VC is also called voice style transfer (VST), and it shares a long history with the objective to clone someone's voice. There is even a potential risk of usage in crime such a voice spoofing (Kinnunen et al., 2012), and also in various applications in entertainment (Nachmani and Wolf, 2019), education (Sisman et al., 2020), security (Wu and Li, 2016), and voice restoring (Yamagishi et al., 2012). Although deep learning made the breakthrough in the VC domain, there still remains challenging problems for real-world application such as low audio quality and similarity to target voice style.

Usually, traditional VC systems require the same utterances for a different speakers to train properly. However, it is hard to collect such parallel data for many speakers, and extension to many-to-many VC systems becomes a laborious task. To overcome this problem, several methods have been developed. First, generative adversarial networks (GAN) based models (Kaneko and Kameoka, 2018; Kaneko et al., 2019, 2020; Kameoka et al., 2018) use the adversarial feedback with cycle-consistent loss to train with non-parallel data. It is hard to train these models, and produce lacking audio quality and transfer performance. The flow-based VC model, Blow (Serrà et al., 2019), is also a non-parallel VC model with normalizing flows using the hyperconditioning mechanism.

Despite an effort to transfer the voice style in non-parallel settings, these models are not able to sufficiently disentangle content and style from the source speech, and thus the converted speech still contains the style of source speech. To overcome this limitation, AUTOVC (Qian et al., 2019) utilizes a simple autoencoder. The carefully designed fixed-length based information bottleneck disentangles the content and style information. For better disentanglement, IDE-VC (Yuan et al., 2021) followed the AUTOVC framework with information-theoretic guidance. AdaIN-VC (Chou et al., 2019) and AGAIN-VC (Chen et al., 2020) employs the instance normalization (Ulyanov et al.,

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

2016) to remove the global style information. Additionally, AGAIN-VC makes use of the activation function as an information bottleneck with a small size of content embedding. However, these models have a trade-off between the audio quality and the disentanglement performance. In the process of disentanglement, the loss of content information results in low audio quality with missing linguistic information. Also, they have to find the proper size of information bottleneck heuristically, and it can be different depending on the data.

Text transcriptions can be used to guide content embedding to learn only linguistic information (Zhang et al., 2019; Park et al., 2020). These models have to be jointly trained with the text-to-speech (TTS) model to encode the linguistic information based on the attention alignment from autoregressive TTS system (Shen et al., 2018). However, they require text transcriptions for training.

Recently, self-supervised representation learning is adopted to extract important representation in speech representation learning (Oord et al., 2018; Wang et al., 2020). Predicting the future latent representation can make the model learn useful information without labeled data. However, such self-supervised representation learning has not yet gotten the attention in a voice conversion task.

In this paper, we present VoiceMixer, which can decompose and transfer voice style through a novel similarity-based information bottleneck and adversarial feedback. We introduce self-supervised representation learning to disentangle and transfer voice style without any text transcription and conditional information extracted from the external feature extractor. Self-supervised learned similarity makes the information bottleneck disentangle the content and style without effort to find the proper downsampling size. Also, we propose an adversarial voice style mixup to learn the latent representation of the converted speech. We first disentangle the discriminator into content and style discriminator. The hidden representations of generator guide each discriminator as conditional information. Through adversarial feedback of disentangled discriminators, the generator has better generalization on the converted speech. The main contributions are as follows:

- We propose the similarity-based information bottleneck with self-supervised representation learning, which can disentangle content and style with only a small loss of content information. This preservation improves the audio quality of converted speech compared to previous methods.
- For better generalization of the converted speech, we propose an adversarial voice style mixup, which learns the converted speech by adversarial feedback with self-supervised guidance, even though the converted speech does not have ground-truth audio.
- For various subjective and objective evaluations, we demonstrate that VoiceMixer has better disentanglement and transfer performance than other baselines in both many-to-many and zero-shot voice style transfer scenarios on the real-world VCTK dataset.

2 Background

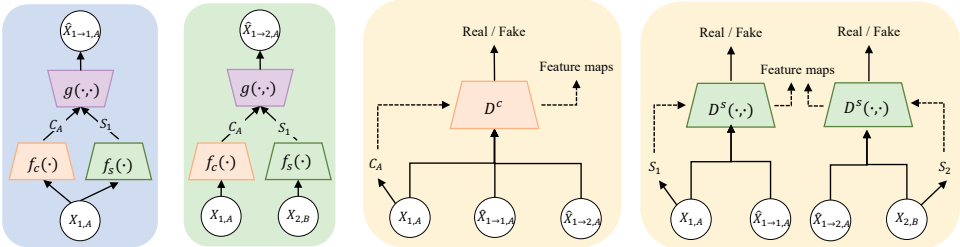
AUTOVC disentangles content and style information from the source speech, and transfers the voice style of target speech through information bottleneck (Qian et al., 2019). The simple autoencoder framework of AUTOVC consists of three modules; a content encoder $f_c(\cdot)$, speaker encoder $f_s(\cdot)$, and a decoder $g(\cdot,\cdot)$. During training, this model only requires self-reconstruction with a fixed-length information bottleneck to disentangle the content and style information.

$$\boldsymbol{C}_A = f_c(\boldsymbol{X}_{1,A}),\ \boldsymbol{S}_1 = f_s(\boldsymbol{X}_{1,A}),\ \hat{\boldsymbol{X}}_{1\to1,A} = g(\boldsymbol{C}_A, \boldsymbol{S}_1) \tag{1}$$

Here, $\boldsymbol{X}_{1,A}$ refers to the utterance "A" from the source speaker "1". $\boldsymbol{C}_A$ denotes content information of the utterance "A", $\boldsymbol{S}_1$ denotes speaker information in the speaker "1", and $\hat{X}_{1\to1,A}$ is self-reconstructed speech which contains the content information $\boldsymbol{C}_A$ and matches the speaker characteristics $\boldsymbol{S}_1$. Although it is a very simple way to decompose each information, a proper information bottleneck size τ is necessary. Formally, this can be represented as follows:

$$\boldsymbol{H}(:, \lfloor t/\tau \rfloor) = \boldsymbol{C}_A(:, t) \tag{2}$$

where $\boldsymbol{H}$ denotes the downsampled feature for time indices $t \in \{1, \cdots T\}$. When the fixed-length information bottleneck size τ is "too narrow", the model has higher reconstruction quality but has poor voice style transfer performance. On the other hand, when the τ is "too wide", the model has higher voice style transfer performance but has lower reconstruction quality. In the process of separating content and style information, some content information is lost even with proper bottleneck size. Therefore, missing some content information in converted voice is inevitable.



(a) Reconstruction (b) Conversion (c) Content discriminator (d) Style discriminator

Figure 1: Overall frameworks of VoiceMixer.

3 VoiceMixer

In this paper, we propose a similarity-based information bottleneck by self-supervised representation learning. For adversarial feedback, we disentangle the discriminator to train content and style separately with self-supervised guidance. By disentangling the discriminator for each information, it can be possible to train the converted speech which does not have ground-truth audio. It is worth noting again that using other supervised features (e.g., pitch contour or text transcription) help the model to disentangle each information, but our model uses self-supervised representation learning without additional features. We describe the details of architecture, similarity-based information bottleneck, and the adversarial voice style mixup in the following subsections.

3.1 Generator

For the generator, we follow the autoencoder frameworks of AUTOVC. As shown in Figure 1a, the generator consists of a content encoder $f_c(\cdot)$ which extracts the content embedding from speech, a speaker encoder $f_s(\cdot)$ which extracts a speaker embedding from speech, and a decoder $g(\cdot,\cdot)$ which generates the speech from content and speaker embeddings represented in Equation 1.

3.2 Similarity-based information bottleneck

Unlike information bottleneck in Equation 2, we downsampled the content embedding according to the similarity between the content embeddings. We assume that the content encoder produces similar content embedding from similar phoneme, and thus we downsampled the adjacent phonetic information to be mapped together. We calculate the similarity $\boldsymbol{Q} = (\boldsymbol{q}_1, \cdots, \boldsymbol{q}_T)$ between content embedding sequence $\boldsymbol{C} = (\boldsymbol{c}_1, \cdots, \boldsymbol{c}_T)$ and shifted content embedding sequence $\boldsymbol{C}_{shift} = (\boldsymbol{c}_2, \cdots, \boldsymbol{c}_{T+1})$ as:

$$\boldsymbol{q}_t = sig(\frac{\boldsymbol{c}_t \cdot \boldsymbol{c}_{t+1}}{\|\boldsymbol{c}_t\| \, \|\boldsymbol{c}_{t+1}\|}/\rho), \tag{3}$$

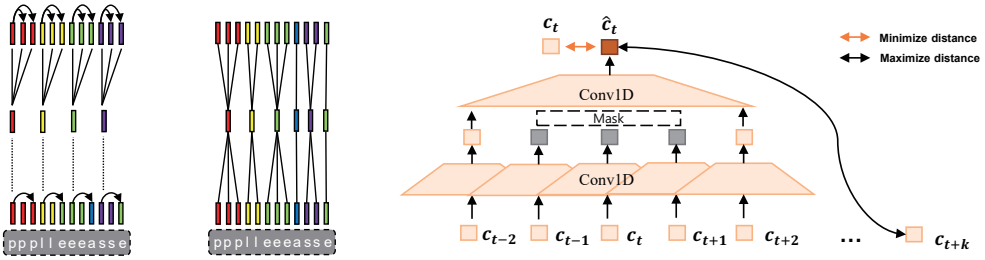
where sig denotes the sigmoid function and ρ is the temperature parameter. Then, we extract the similarity-based duration $\boldsymbol{\mathcal{D}} = (d_1, \cdots, d_N)$ where $\boldsymbol{d}_n$ is cumulative sum until the similarity q_t is under the average similarity, and the $\boldsymbol{d}_{n+1}$ is computed again from q_{t+1} until time step T.

Gaussian down/up-sampling Assume that the center of same content have the largest information of that content, then we apply the Gaussian downsampling to focus attention to the center. Given the content embedding to be downsampled $\boldsymbol{C}$, duration $\boldsymbol{\mathcal{D}}$, and learnable range parameter $\boldsymbol{\sigma_d} = (\sigma_{d,1}, \cdots, \sigma_{d,N})$ like (Shen et al., 2020), we compute downsampled sequence $\boldsymbol{H} = (\boldsymbol{h}_1, \cdots, \boldsymbol{h}_N)$ as:

$$\alpha_i = \frac{d_i}{2} + \sum_{j=1}^{i-1} d_j, \quad w_{ni} = \frac{\mathcal{N}(n; \alpha_i, \sigma_{d,i}^2)}{\sum_{j=1}^{T} \mathcal{N}(n; \alpha_j, \sigma_{d,j}^2)}, \quad \boldsymbol{h}_n = \sum_{i=1}^{T} w_{ni} \boldsymbol{c}_i \tag{4}$$

After that, we use Gaussian upsampling as a TTS model following (Shen et al., 2020) to upscale $\boldsymbol{H}$ to upsampled content sequence $\tilde{\boldsymbol{C}} = (\tilde{\boldsymbol{c}}_1, \cdots, \tilde{\boldsymbol{c}}_T)$ with the same duration of $\boldsymbol{\mathcal{D}}$, range parameter for upsampling $\boldsymbol{\sigma_u} = (\sigma_{u,1}, \cdots, \sigma_{u,N})$, and then $\tilde{\boldsymbol{C}}$ is fed to $g(\cdot,\cdot)$ to generate the mel-spectrogram as:

$$w_{ti} = \frac{\mathcal{N}(t; \alpha_i, \sigma_{u,i}^2)}{\sum_{j=1}^{N} \mathcal{N}(t; \alpha_j, \sigma_{u,j}^2)}, \quad \tilde{\boldsymbol{c}}_t = \sum_{i=1}^{N} w_{ti} \boldsymbol{h}_i, \quad \hat{\boldsymbol{X}}_{1\to1,A} = g(\tilde{\boldsymbol{C}}_A, \boldsymbol{S}_1) \tag{5}$$



(a) (b) (c)

Figure 2: (a) Fixed-length information bottleneck. (b) Similarity-based information bottleneck. (c) Context network for self-supervised representation learning on content embedding.

3.3 Auxiliary losses for similarity

Contrastive loss To increase the similarity between the adjacent content embeddings, we train the content encoder with self-supervised representation learning. The content embedding is fed to a context network to learn a content representation illustrated in Figure 2. To train in non-autoregressive manner, we utilize the masked convolutional blocks (Liu et al., 2020) to predict a content embedding $\hat{\boldsymbol{C}} = (\hat{\boldsymbol{c}}_1, \cdots, \hat{\boldsymbol{c}}_T)$ from the adjacent content embeddings, and the contrastive loss for positive sample is defined to minimize distance between predicted content embedding $\hat{\boldsymbol{C}}$ and content embedding $\boldsymbol{C}$:

$$\mathcal{L}_{sim^+} = \mathbb{E}\Big[-\frac{1}{T}\sum_{i}^{T} \log sig\Big(\frac{\boldsymbol{c}_i \cdot \hat{\boldsymbol{c}}_i}{\|\boldsymbol{c}_i\| \, \|\hat{\boldsymbol{c}}_i\|}/\rho\Big)\Big] \tag{6}$$

where sig denotes the sigmoid function and ρ represents the temperature parameter.

To remove style information on the content embedding in an unsupervised manner, we prevent context network to predict future representation prediction of content embedding. While negative samples are uniformly sampled from the same utterance in (Baevski et al., 2020), we only sample the k-th future content embedding as a negative sample to prevent maximizing distance between the contents similar to each other. We maximize cosine distance between predicted content embedding and k-th future representation of content embedding, and the contrastive loss for negative sample is:

$$\mathcal{L}_{sim^-} = \mathbb{E}\Big[\frac{1}{T}\sum_{i}^{T} \log sig\Big(\frac{\boldsymbol{c}_{i+k} \cdot \hat{\boldsymbol{c}}_i}{\|\boldsymbol{c}_{i+k}\| \, \|\hat{\boldsymbol{c}}_i\|}/\rho\Big)\Big]. \tag{7}$$

Adversarial speaker classification To enforce speaker disentanglement on the content embedding, we apply adversarial speaker classification in a supervised manner (using speaker label $\boldsymbol{y}_i$) as:

$$\mathcal{L}_{advsc} = \mathbb{E}\Big[-\frac{1}{T}\sum_{i}^{T} \boldsymbol{y}_i log(cls(\boldsymbol{c}_i))\Big] \tag{8}$$

where cls denotes speaker classifier. Following (Hsu et al., 2019), we use a gradient reversal layer before the content embedding is fed to the speaker classifier to switch the sign of gradient.

3.4 Disentangled discriminator with self-supervised guidance

Unlike the previous GAN-based VC model which uses the cycle-consistency training to preserve linguistic information by two-way generation, we follow the autoencoder based reconstruction method for training. For adversarial feedback, we divide the discriminator into the content discriminator $D^c(\cdot,\cdot)$ and style discriminator $D^s(\cdot,\cdot)$ to disentangle content and style, respectively. To guide each discriminator for each attribute, we condition the content embedding to the content discriminator and style embedding to the style discriminator as a self-supervised conditional information illustrated in Figure 1. For the training objectives, we use the LSGAN (Mao et al., 2017) as followed:

$$\begin{aligned}\mathcal{L}_{D,adv} = \mathbb{E}\Big[&\|D^c(\boldsymbol{X}_{1,A}, \boldsymbol{C}_A) - 1\|_2 + \|D^c(\hat{\boldsymbol{X}}_{1\to1,A}, \boldsymbol{C}_A)\|_2 \\ &+ \|D^s(\boldsymbol{X}_{1,A}, S_1) - 1\|_2 + \|D^s(\hat{\boldsymbol{X}}_{1\to1,A}, S_1)\|_2\Big]\end{aligned} \tag{9}$$

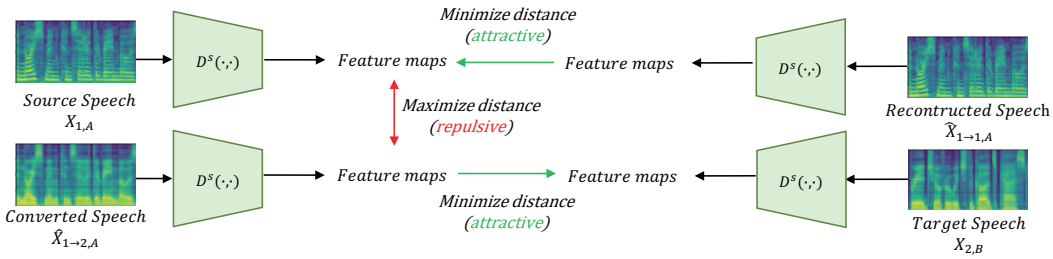


Figure 3: Style feature matching loss for reconstructed and converted mel-spectrogram.

$$\mathcal{L}_{G,adv} = \mathbb{E}\Big[\|D^c(\hat{\boldsymbol{X}}_{1\to1,A}, \boldsymbol{C}_A) - 1\|_2 + \|D^s(\hat{\boldsymbol{X}}_{1\to1,A}, \boldsymbol{S}_1) - 1\|_2\Big] \tag{10}$$

Feature Matching for reconstruction We use the feature matching loss to train the generator, which minimizes the distance of the discriminator's features between a ground truth and a generated speech. For each discriminator, we use the content feature matching loss $\mathcal{L}_{content}$ from the content discriminator for content and the style feature matching loss $\mathcal{L}_{style}$ from the style discriminator for style.

$$\mathcal{L}_{content}(G; D^c) = \mathbb{E}\Big[\sum_{i=1}^{K} \frac{1}{N_i} \|D_i^c(\boldsymbol{X}_{1,A}, \boldsymbol{C}_A) - D_i^c(\hat{\boldsymbol{X}}_{1\to1,A}, \boldsymbol{C}_A)\|_1\Big] \tag{11}$$

$$\mathcal{L}_{style}(G; D^s) = \mathbb{E}\Big[\sum_{i=1}^{K} \frac{1}{N_i} \|D_i^s(\boldsymbol{X}_{1,A}, \boldsymbol{S}_1) - D_i^s(\hat{\boldsymbol{X}}_{1\to1,A}, \boldsymbol{S}_1)\|_1\Big] \tag{12}$$

where K denotes the number of discriminator blocks and N_i is the number of features in i-th discriminator block. The total loss for reconstructed mel-spectrogram is defined as:

$$\mathcal{L}_{rec} = \mathcal{L}_{G,adv} + \lambda_c \mathcal{L}_{content}(G; D^c) + \lambda_s \mathcal{L}_{style}(G; D^s) + \lambda_{mel} \mathcal{L}_{mel} \tag{13}$$

where $\mathcal{L}_{mel}$ is mean absolute error between $X_{1,A}$ and $\hat{X}_{1\to1,A}$.

3.5 Adversarial Voice Style Mixup

By introducing the disentangled discriminator for each information, we can train the reconstructed speech for each disentangled feature. However, our goal is to convert voice by disentangling the source style and transferring the target style. To learn the latent representations of the converted speech, we proposed an adversarial voice style mixup, which can train the converted speech by using the disentangled discriminator with a self-supervised condition. Even though converted speech does not have ground-truth (GT) samples, the converted mel-spectrogram can be trained with adversarial feedback through each discriminator. It is worth noting that the model only uses a self-supervised hidden representation of the generator as conditional features without any external feature extractor for conditional information. The GAN losses for the converted mel-spectrogram are defined as:

$$\mathcal{L}^*_{D,adv} = \mathbb{E}\Big[\|D^c(\hat{\boldsymbol{X}}_{1\to2}, \boldsymbol{C}_A)\|_2 + \|D^s(\hat{\boldsymbol{X}}_{1\to2}, \boldsymbol{S}_2)\|_2\Big], \tag{14}$$

$$\mathcal{L}^*_{G,adv} = \mathbb{E}\Big[\|D^c(\hat{\boldsymbol{X}}_{1\to2}, \boldsymbol{C}_A) - 1\|_2 + \|D^s(\hat{\boldsymbol{X}}_{1\to2}, \boldsymbol{S}_2) - 1\|_2\Big] \tag{15}$$

Feature Matching for Mixup We can also use the feature matching loss for a converted speech by the disentangled discriminator. For the content representation, the feature distance of content discriminator between converted speech and source speech is minimized as:

$$\mathcal{L}^*_{content}(G; D^c) = \mathbb{E}\Big[\sum_{i=1}^{K} \frac{1}{N_i} \|D_i^c(\boldsymbol{X}_{1,A}, \boldsymbol{S}_1) - D_i^c(\hat{\boldsymbol{X}}_{1\to2,A}, \boldsymbol{S}_2)\|_1\Big] \tag{16}$$

For the style representation, the feature distance of style between converted and target speech is minimized as following:

$$\mathcal{L}^*_{style^+}(G;D^s) = \mathbb{E}\Big[\sum_{i=1}^{K}\frac{1}{N_i}\|D^s_i(\boldsymbol{X}_{2,B},\boldsymbol{S}_2) - D^s_i(\hat{\boldsymbol{X}}_{1\to 2,A},\boldsymbol{S}_2)\|_1\Big] \tag{17}$$

We call it *"Attractive style loss"* which minimizes the style feature distance between the same style of the same speaker. As shown in Figure 3, we also introduce *"Repulsive style loss"* to maximize the style feature distance between the different style of the converted speech and source speech as:

$$\mathcal{L}^*_{style^-}(G;D^s) = \mathbb{E}\Big[-\sum_{i=1}^{K}\frac{1}{N_i}\|D^s_i(\boldsymbol{X}_{1,A},\boldsymbol{S}_1) - D^s_i(\hat{\boldsymbol{X}}_{1\to 2,A},\boldsymbol{S}_2)\|_1\Big] \tag{18}$$

When the content encoder does not disentangle the source speaker, the converted speech may contain the style of the source speaker. Thus, this repulsive style loss constrains the converted speech to have the style of source speaker. The total loss for converted mel-spectrogram is defined as:

$$\mathcal{L}_{con} = \mathcal{L}^*_{G,adv} + \lambda_c\mathcal{L}^*_{content}(G;D^c) + \lambda_s\mathcal{L}^*_{style^+}(G;D^s) + \lambda_{s^-}\mathcal{L}^*_{style^-}(G;D^s) \tag{19}$$

Our final objectives for the discriminators and generator are represented as:

$$\mathcal{L}_D = \mathcal{L}_{D,adv} + \lambda_{con}\mathcal{L}^*_{D,adv} \tag{20}$$

$$\mathcal{L}_G = \mathcal{L}_{rec} + \lambda_{con}\mathcal{L}_{con} + \lambda_{sim^+}\mathcal{L}_{sim^+} + \lambda_{sim^-}\mathcal{L}_{sim^-} + \lambda_{advsc}\mathcal{L}_{advsc} \tag{21}$$

4 Experiment and result

We evaluated our model with the VCTK dataset, which has 46 hours of audio from 109 speakers (Veaux et al., 2017). We divided the dataset into 98 speakers as base speakers for many-to-many VST and 10 speakers as the novel speakers for zero-shot VST. The base speaker is split into train and test sets. For the non-parallel dataset setting, the training set consists of different utterances for all of the speakers, and then the test set consists of 25 same utterances. We transform the mel-spectrogram with 80 bins from the audio downsampled at 22,050Hz. The spectrogram is inverted to a waveform by the pre-trained HiFi-GAN (Kong et al., 2020). For many-to-many VST, we randomly choose 20 speakers with equal distribution of male and female from the base speakers. For zero-shot VST, we randomly choose 10 speakers from the base speakers, and 10 speakers from the novel speakers. For each setting, a single utterance is selected from each speaker, and then all the possible pairs of utterances (20 x 20 = 400) are produced. The generated speech is evaluated by the following metrics.

4.1 Evaluation metrics

Subjective metrics We conduct the naturalness and similarity mean opinion score (MOS) test. For the naturalness of speech (Naturalness), converted samples are evaluated by at least 20 raters on a scale of 1 to 5. For the similarity of converted speech (Similarity), both converted speech and the speech of target speaker are presented to at least 20 raters, and the raters evaluate on a scale of 1 to 4. We also report the score as a percentage from the binary rating introduced in (Serrà et al., 2019).

Objective metrics For the similarity measurement, we conduct 3 objective metrics; the equal error rate of the automatic speaker verification (ASV EER), the mel-cepstral distance (MCD_{13}) (Kubichek, 1993), and the $F0$ root mean square error (RMSE_{f0}). We use the pre-trained ASV model (Chung et al., 2020) trained by the large scale dataset, VoxCeleb2 (Chung et al., 2018). We compute the EER at which both acceptance and rejection errors are equal from the sample pairs from the converted and target speech (400 x 20 = 8000). We apply the dynamic time warping (DTW) to calculate the MCD_{13} and RMSE_{f0} between converted and target speech, which has different time alignment. For the naturalness measurement, we evaluate Fréchet DeepSpeech Distance (FDSD) (Bińkowski et al., 2020; Gritsenko et al., 2020), which is the distance between the high-level features of GT and generated audio from the pre-trained DeepSpeech2 (Amodei et al., 2016). Because DeepSpeech2 is a speech recognition model trained with connectionist temporal classification loss to classify the text sequence, the hidden representations are related to linguistic information. Thus, we use FDSD between the converted speech and the source speech for objective naturalness measurement.

4.2 Implementation details

The generator consists of a speaker encoder, content encoder, similarity-based information bottleneck, and decoder. We train the speaker encoder during training and the speaker embedding is extracted from the speaker encoder which has the same architecture as the reference encoder of (Skerry-Ryan et al., 2018). The source speech is fed to the content encoder consisting of a pre-net and 3 blocks of the multi-receptive field fusion module (MRF) (Kong et al., 2020). The pre-net is a 2-layer linear network with 384 channels. The output concatenated with source speaker embedding is fed to the MRF module. The MRF module returns the sum of output from 384 channels of multiple convolutional layers with multiple dilations and multiple receptive filed. We use the combination of two dilation of [1, 3], and two receptive filed of [3, 7] for the MRF module. Before the features are fed to MRF module, the instance normalization (IN) (Ulyanov et al., 2016) is applied.

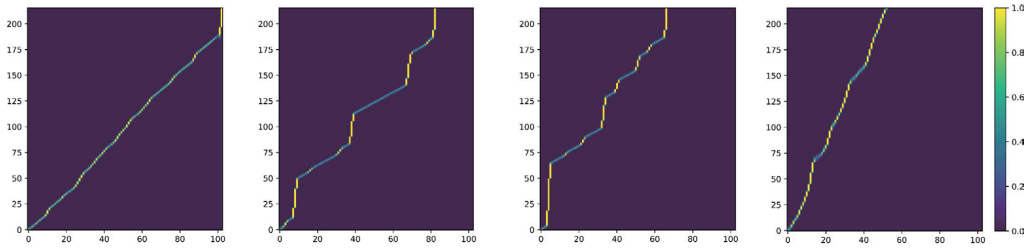
The similarity-based information bottleneck has 2 sigma predictors for down/up-sampling. Both sigma predictors consist of three convolutional layers followed by a linear layer with a softplus activation function, similar to (Shen et al., 2020). The sigma predictor for downsampling uses the similarity-based duration as input. The sigma predictor for upsampling uses the same duration and downsampled content embedding as input. The content embedding is fed to both adversarial speaker classifier and contrastive encoder. The adversarial speaker classifier consists of five 1D convolutional layers followed by a linear layer to predict speaker identity. To remove the speaker identity on content embedding, a gradient reversal layer is used before the first layer of adversarial speaker classifier. For the contrastive encoder, we use three masked convolution blocks of (Liu et al., 2020) with 384 channels, receptive field size of 23, and mask size of [5, 7, 9]. We set the k as 24 (about 0.3s) which is over the average duration of consonant-vowel syllables (Steinschneider et al., 2013).

After the similarity-based information bottleneck, the upsampled feature concatenated with the target speaker embedding is fed to the decoder, which consists of a conditional layer, 3 blocks of MRF module, and the linear layer. The conditional layer is a single 1D convolutional layer with 384 channels. The MRF module is the same architecture with the encoder without applying the IN to the feature before fed to the MRF module. Finally, the mel-spectrogram is predicted by the linear layer.

The content discriminator consists of 4 blocks which have a speech-side block and content condition side block following (Lee et al., 2021). Each block has 2 layer 1D convolutional layer. The hidden representation of the condition-side block is added to the speech-side hidden representation of [256, 512, 1024, 1024]. The style discriminator consists of 4 blocks which have a speech-side block and style-condition side linear layers. Each output of the linear layer is added to the speech-side hidden representation of [256, 512, 1024, 1024]. We report more details of hyperparameter in Appendix A.

4.3 Information bottleneck alignment

To compare the similarity-based information bottleneck using a different combination of auxiliary loss, Figure 4 show the alignment between source speech and downsampled content embedding. The model without both $\mathcal{L}_{sim^+}$ and $\mathcal{L}_{sim^-}$ almost shows the diagonal alignment; it means that content embedding is not extracted according to only content information. The model without $\mathcal{L}_{advsc}$ shows more closer to attention alignment of Tacotron2 (Shen et al., 2018). These results show that using $\mathcal{L}_{sim^+}$ and $\mathcal{L}_{sim^-}$ is more important to disentangle content and speaker information than $\mathcal{L}_{advsc}$. Without using any text transcript and target duration, our model with total auxiliary loss produces an alignment similar to phonetic alignment, and shows better performance as shown in Table 3.



(a) w/o $\mathcal{L}_{sim^+}$ and $\mathcal{L}_{sim^-}$ (b) w/o $\mathcal{L}_{advsc}$ (c) VoiceMixer (d) Phoneme alignment

Figure 4: Alignment of similarity-based information bottleneck

Table 1: Many-to-many VST evaluation results.

Method	Naturalness	Similarity	ASV EER	MCD_{13}	RMSE_{f0}	FDSD
GT	4.07±0.03	93.7%	5.5%	-	-	-
HiFi-GAN (Vocoder)	3.92±0.03	92.8%	7.6%	3.19	36.22	0.546
StarGAN-VC	3.48±0.04	42.5%	22.7%	7.97	33.98	13.561
AGAIN-VC	3.60±0.03	52.2%	14.3%	6.75	41.39	3.185
AUTOVC (τ=16)	3.65±0.03	52.1%	18.3%	6.70	44.05	5.753
AUTOVC (τ=32)	3.64±0.03	52.1%	14.0%	6.44	39.93	10.703
AUTOVC + $\mathcal{L}_{advsc}$ (τ=16)	3.63±0.03	54.6%	14.7%	6.58	39.40	6.036
Blow	3.12±0.04	33.2%	52.0%	6.74	44.55	15.112
VoiceMixer (Ours)	3.78±0.03	55.9%	12.5%	6.77	42.76	2.080

Table 2: Zero-shot VST evaluation results.

Method	Naturalness	Similarity	ASV EER	MCD_{13}	RMSE_{f0}	FDSD
GT	4.08±0.03	96.3%	4.4%	-	-	-
HiFi-GAN (Vocoder)	4.03±0.03	95.4%	6.0%	3.34	38.33	0.576
AGAIN-VC	3.29±0.03	58.2%	15.0%	6.96	44.81	3.261
AUTOVC (τ=16)	3.40±0.03	46.3%	25.0%	6.92	46.33	5.227
AUTOVC (τ=32)	3.24±0.03	59.7%	20.7%	6.65	39.93	9.741
AUTOVC + $\mathcal{L}_{advsc}$ (τ=16)	3.39±0.03	58.9%	21.9%	6.78	42.08	5.675
VoiceMixer (Ours)	3.75±0.03	63.4%	18.5%	7.04	44.70	2.416

4.4 Audio quality and style transfer performance

For the many-to-many VST evaluation, we compared our model with four VC models; StarGAN-VC (Kameoka et al., 2018), AGAIN-VC (Chen et al., 2020), Blow (Serrà et al., 2019), and AUTOVC (Qian et al., 2019). All models are trained with the same dataset, and the implementation details are described in the Appendix A. We train AUTOVC in various settings with each having different sizes of information bottleneck. For a fair comparison, we also implement the AUTOVC model with an adversarial speaker classifier. Table 1 shows that our model outperforms other models in Naturalness and FDSD metrics. Our model also shows better transfer performance on the Similarity and ASV EER. When disentangling the content and style information, AGAIN-VC and AUTOVC models lose a lot of content information, and thus the converted speech has lower Naturalness and higher FDSD score. The adversarial speaker classifier on the content embedding can help the disentanglement performance of models, but the naturalness can be degraded.

For the zero-shot VST evaluation, we compared our model with two VC models; AGAIN-VC and AUTOVC as shown in Table 2. We also implement various AUTOVC trained with different information bottleneck size and adversarial speaker classification loss. Our model has better performance in Naturalness and FDSD. In terms of similarity, even though the AGAIN-VC has higher performance in ASV EER, our model has better performance in Similarity. In terms of AUTOVC, it is hard to select the proper down-sampling factor, which has a trade-off between naturalness and similarity. Thus, it is important to note that our proposed similarity-based information bottleneck need not find the proper factor, which is learned by self-supervised representation learning with a small loss of content information. Our audio samples are available on the demo page.[1]

4.5 Ablation study

We conducted ablation studies for the information bottleneck and adversarial feedback in Table 3. We evaluate each model for the same zero-shot VST setting of Table 2. In terms of similarity-based information bottleneck, absence of $\mathcal{L}_{advsc}$ or using fixed-length information bottleneck (IB) makes it harder to disentangle the content and style in content embedding. For better disentanglement, it is essential to use $\mathcal{L}_{sim^+}$ with $\mathcal{L}_{sim^-}$, and it induces the information bottleneck to downsample the

[1] `https://anonymous-speech.github.io/voicemixer/index.html`

Table 3: Ablation studies for zero-shot voice style transfer

Method	Naturalness	Similarity	ASV EER	MCD_{13}	$RMSE_{f0}$	FDSD
VoiceMixer (Ours)	3.72±0.03	63.7%	18.5%	7.04	44.70	2.416
w/o $\mathcal{L}_{advsc}$	3.74±0.03	41.3%	32.2%	7.33	47.63	1.217
w/o $\mathcal{L}_{sim^+}$ and $\mathcal{L}_{sim^-}$	3.21±0.03	42.3%	38.0%	7.92	54.11	4.634
w/o $\mathcal{L}_{sim^-}$	3.18±0.03	45.5%	24.2%	7.72	45.05	11.354
w fixed-length IB (τ=16)	3.35±0.03	51.8%	28.5%	7.38	46.14	2.827
w fixed-length IB (τ=32)	3.28±0.03	56.8%	20.7%	6.97	44.16	2.823
w/o GAN	3.70±0.03	58.5%	20.5%	6.95	44.79	2.666
w/o disentangled discriminator	3.68±0.03	60.4%	18.3%	7.02	45.23	3.502
w/o $\mathcal{L}^*_{style^-}$	3.69±0.03	55.0%	21.6%	7.02	45.66	2.173

content embedding similar to the phoneme alignment as shown in Figure 4. Additionally, when trained without all auxiliary loss or any information bottleneck, these models are not able to convert any voice, but only reconstruct source speech. Using GAN makes the model have better performance on all of the metrics. The model trained with a single discriminator (instead of disentangled discriminator) shows lower performance in FDSD, in this regard, disentangled discriminator encourages better generalization for each attribute. Removing $\mathcal{L}^*_{style^-}$ results in lower performance in both subjective and objective similarity metrics even though the FDSD decreases.

4.6 Content and speaker disentanglement

We conduct speaker classification on content embedding to evaluate the disentanglement performance compared to AGAIN-VC and AUTOVC. Table 4 represents the classification results. Our model shows the lowest accuracy, which mean our model has better disentanglement performance by removing speaker identify on content embedding despite having the largest feature dimensions of 384 (The content embedding of AUTOVC has 64 dimensions). Even though AUTOVC trained with adversarial speaker classifier has comparable disentanglement performance, AUTOVC loses a lot of content information in their fixed-length based information bottleneck as shown in Table 1. AUTOVC also have to heuristically find optimal downsampling size for a good balance between content and style. On the other hand, our proposed model finds proper downsampling size based on the similarity of content embedding learned by self-supervised representation learning with only a small loss of content information.

Table 4: Speaker classification accuracy on content embedding of the autoencoder based VC models.

Method	Accuracy[%]
AGAIN-VC	27.31
AUTOVC (τ=16)	10.27
AUTOVC (τ=32)	4.47
AUTOVC + L_{advsc} (τ=16)	3.11
VoiceMixer (Ours)	1.47

5 Conclusion

We presented a VoiceMixer, which can decompose and transfer voice style by similarity-based information bottleneck and adversarial feedback with self-supervised representation learning. Without an effort to find the proper size of information bottleneck carefully, our model is able to learn proper downsampling factor with self-supervised representation learning. We successfully demonstrated that our novel information bottleneck can decompose content and style information from the source speech with a small loss of content information. Moreover, the alignment of information bottleneck is similar to phonetic alignment despite not using any text transcript and target phoneme duration. we also show the adversarial voice style mixup makes it possible to learn the latent representation of converted speech which does not have ground-truth speech, and it improves the overall generalization.

While there remains a gap between target and converted voice, we believe pre-trained speaker encoder with large-scale dataset could improve the VST performance. Moreover, we see our self-supervised learning based speech disentanglement extending to other tasks. For future work, we will apply our speech disentanglement to TTS without text (Dunbar et al., 2019), which is a challenging task to synthesize the speech without any text transcript for use of the untranscribed large-scale speech data.

References

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182. PMLR, 2016.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33, 2020.

Mikołaj Bińkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C. Cobo, and Karen Simonyan. High fidelity speech synthesis with adversarial networks. In *International Conference on Learning Representations*, 2020.

Yen-Hao Chen, Da-Yi Wu, Tsung-Han Wu, and Hung-yi Lee. Again-vc: A one-shot voice conversion using activation guidance and adaptive instance normalization. *arXiv preprint arXiv:2011.00316*, 2020.

Ju-chieh Chou, Cheng-chieh Yeh, and Hung-yi Lee. One-shot voice conversion by separating speaker and content representations with instance normalization. *arXiv preprint arXiv:1904.05742*, 2019.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *Proc. Interspeech 2018*, pages 1086–1090, 2018.

Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han. In defence of metric learning for speaker recognition. In *Proc. Interspeech 2020*, pages 2977–2981, 2020.

Ewan Dunbar, Robin Algayres, Julien Karadayi, Mathieu Bernard, Juan Benjumea, Xuan-Nga Cao, Lucie Miskic, Charlotte Dugrain, Lucas Ondel, Alan W. Black, Laurent Besacier, Sakriani Sakti, and Emmanuel Dupoux. The Zero Resource Speech Challenge 2019: TTS Without T. In *Proc. Interspeech 2019*, pages 1088–1092, 2019. doi: 10.21437/Interspeech.2019-2904.

Alexey A Gritsenko, Tim Salimans, Rianne van den Berg, Jasper Snoek, and Nal Kalchbrenner. A spectral energy distance for parallel speech synthesis. In *Advances in Neural Information Processing Systems*, 2020.

Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Yu-An Chung, Yuxuan Wang, Yonghui Wu, and James Glass. Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5901–5905. IEEE, 2019.

Hirokazu Kameoka, 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)*, pages 266–273. IEEE, 2018.

Takuhiro Kaneko and Hirokazu Kameoka. Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2100–2104. IEEE, 2018.

Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6820–6824. IEEE, 2019.

Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. Cyclegan-vc3: Examining and improving cyclegan-vcs for mel-spectrogram conversion. *arXiv preprint arXiv:2010.11672*, 2020.

Tomi Kinnunen, Zhi-Zheng Wu, Kong Aik Lee, Filip Sedlak, Eng Siong Chng, and Haizhou Li. Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech. In *ICASSP 2012-2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4401–4404. IEEE, 2012.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17022–17033, 2020.

R Kubichek. Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, volume 1, pages 125–128, 1993.

Sang-Hoon Lee, Hyun-Wook Yoon, Hyeong-Rae Noh, Ji-Hoon Kim, and Seong-Whan Lee. Multi-spectrogan: High-diversity and high-fidelity spectrogram generation with adversarial style combination for speech synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Alexander H Liu, Yu-An Chung, and James Glass. Non-autoregressive predictive coding for learning speech representations from local dependencies. *arXiv preprint arXiv:2011.00406*, 2020.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

Eliya Nachmani and Lior Wolf. Unsupervised singing voice conversion. pages 2583–2587, 2019.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Seung-won Park, Doo-young Kim, and Myun-chul Joe. Cotatron: Transcription-guided speech encoder for any-to-many voice conversion without parallel data. *arXiv preprint arXiv:2005.03295*, 2020.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *International Conference on Machine Learning*, pages 5210–5219. PMLR, 2019.

Joan Serrà, Santiago Pascual, and Carlos Segura. Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. *arXiv preprint arXiv:1906.00794*, 2019.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

Jonathan Shen, Ye Jia, Mike Chrzanowski, Yu Zhang, Isaac Elias, Heiga Zen, and Yonghui Wu. Non-attentive tacotron: Robust and controllable neural tts synthesis including unsupervised duration modeling. *arXiv preprint arXiv:2010.04301*, 2020.

Berrak Sisman, Junichi Yamagishi, Simon King, and Haizhou Li. An overview of voice conversion and its challenges: From statistical modeling to deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:132–157, 2020.

RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A. Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *International Conference on Machine Learning*, pages 4700–4709, 2018.

Mitchell Steinschneider, Kirill V. Nourski, and Yonatan I. Fishman. Representation of speech in human auditory cortex: Is it special? *Hearing Research*, 305:57–73, 2013. ISSN 0378-5955. doi: https://doi.org/10.1016/j.heares.2013.05.013. Communication Sounds and the Brain: New Directions and Perspectives.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. 2017.

Luyu Wang, Kazuya Kawakami, and Aaron van den Oord. Contrastive predictive coding of audio with an adversary. *Proc. Interspeech 2020*, pages 826–830, 2020.

Zhizheng Wu and Haizhou Li. On the study of replay and voice conversion attacks to text-dependent speaker verification. *Multimedia Tools and Applications*, 75(9):5311–5327, 2016.

Junichi Yamagishi, Christophe Veaux, Simon King, and Steve Renals. Speech synthesis technologies for individuals with vocal disabilities: Voice banking and reconstruction. *Acoustical Science and Technology*, 33(1):1–5, 2012.

Siyang Yuan, Pengyu Cheng, Ruiyi Zhang, Weituo Hao, Zhe Gan, and Lawrence Carin. Improving zero-shot voice style transfer via disentangled representation learning. In *International Conference on Learning Representations*, 2021.

Jing-Xuan Zhang, Zhen-Hua Ling, and Li-Rong Dai. Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:540–552, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 1
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We attach the training and inference code at the supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and Appendix A
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes]