Mathmatical Formulation of Loss Function

We train the proposed TPSE-VC framework by only considering non-parallel training data. Here we indicate the **Nonparallel** data as $(X_1^n, X_2^n) \sim Nonparallel$ where superscript n denote the speech sampled from **Nonparallel** set. Further explanation, X_1^n mean the source speech and X_2^n indicate the target speech whose both speaker identities and content utterances are different from X_1^n. We will give a detailed description of loss functions based on the above definition in the following sections.

Reconstruction Loss

The reconstruction L1 loss helps $Generator$ to recover the origin spectrogram while the inputs are the same , it can facilitate consistency preserving of the converted spectrogram:

$$\mathcal{L}_{recon} = |G(X_1^n, X_1^n) - X_1^n|_1$$

where G denotes the generator. This reconstruction loss is an essential part for both accelerated optimization and superior performance, and ensures our model does not loss too much information during conversion process.

Content Loss and Style Loss

Besides using L1 loss in the spectrogram-level, we also consider the content and style constraints in the feature-level. Due to our U-Net like architecture, we get multi-scale features in different time resolutions, and our goal is changing the speaker style while preserving the lingual content. Then we calculate the sum of the $euclideandistances$ between the converted content or style features and its corresponding origin features.

$$\mathcal{L}_{content} = \sum_{\forall l}(| En_c(G(X_1^n, X_2^n)) - En_c(X_1^n) |_2)$$

$$\mathcal{L}_{style} = \sum_{\forall l} (|SPAttention^{(l)}(En_c(G(X_1^n, X_2^n)), En_s(G(X_1^n, X_2^n))) \ - SPAttention^{(l)}(En_c(X_1^n), En_s(X_2^n)) |_2)$$

where l is the index of output of encoders, which represents the multi-scale features in different time resolutions. And it does not seem necessary that using $SPAttention$ for the converted speech, but we believe that it is a chance for considering both style and content, and it helps to learn a ***diagonal attention*** while the content and style come from the same speech.

Adversarial Loss

In typical GAN, the discriminator distinguishes the speech spectrogram is real or not, while encouraging the generator to synthesize realistic speech spectrogram, on the other hand, the generator is expected to synthesize a realistic enough speech spectrogram to fool the discriminator. The generator and discriminator are trained alternatingly in an adversarial manner.

$$\mathcal{L}_{adv}^{G} = -0.5 * (D(G(X_1^n, X_2^n)) + D(G(X_1^n, X_1^n)))$$

$$\mathcal{L}_{adv}^{D} = D(G(X_1^n, X_2^n)) - D(X_2^n) + \lambda(| \bigtriangledown D(\alpha X_2^n + \ (1-\alpha)G(X_1^n, X_2^n))| - 1)^2$$

where D and G denote $Discriminator$ and $Generator$ respectively. Note that in this paper, the Wasserstein GAN with gradient penalty (WGAN-GP) is adopted instead of the original GAN to mitigate the

training instability issue.

Total Loss

The overall loss function is a weighted sum of individual loss functions described above, which can be defined as: $$mathcal{L}*{total}^{G} = *mathcal{L}*{recon} + \lambda_1 \mathcal{L}*{content} + *lambda_2 *mathcal{L}* {style} + \lambda_3 \mathcal{L}_{adv}^{G} $$

$$\mathcal{L}*{total}^{D} = *lambda_3 *mathcal{L}*{adv}^{D} $$

where λ_1, λ_2 and λ_3 are the hyperparameters which control the relative importance of each other.