**INTRODUCTION**

- **Voice Conversion (VC):** converts a source speaker’s speech to sound like a target speaker’s voice.
  - VC preserves target’s speaker identity and source’s phonetic context.
  - One-shot VC methods typically disentangle speaker identity and phonetic context. Then speaker identity representation is modified while keeping phonetic context constant.
- **Challenges:** The models cannot fully disentangle these factors as shown in a previous study.
- **Proposal:** We posit that the senone posteriorgrams (PPG) from an already-trained ASR model can be used in lieu of learned phonetic context representations.
  - We focus on learning only the speaker representation.
  - We present a one-shot voice conversion technique by modifying the learned speaker identity representation.
  - Through experiments, we show that modification of these factors allows better disentanglement and hence transformation of voice.

**MODEL**

- Our proposed model consists of an encoder and a decoder (RNNs)
  - Encoder’s input is MCEPs, outputting a speaker embedding vector: \( z = E(X) \)
  - Decoder takes the generated speaker embedding along with PPG sequence as input, and generates the acoustic features: \( X = D(P, z) \)
  - We train the model by optimizing the training loss:
    \[
    \ell(X, X') = \sum_{i=1}^{N} ||x_i - x'_i||^2
    \]
    
- To compute VC:
  - Compute \( z^e \) and \( z^u \) of the source and target utterances.
  - Compute average diff vector \( z^{diff} = z^e - z^u \)
  - Add average diff vector to source \( z^{clean}_i = z^e_i + z^{diff} \)

**EXPERIMENT**

- We used the TIMIT corpus as the training data.
- To compute the phonetic posteriorgrams, we use Kaldi
- We use librospeech as speech corpus to train ASR.

**Encoder**

<table>
<thead>
<tr>
<th>recurrent layer</th>
<th>GRU-1024, Dropout</th>
</tr>
</thead>
<tbody>
<tr>
<td>output layer</td>
<td>FC-(D_z), ReLU</td>
</tr>
</tbody>
</table>

**Decoder**

<table>
<thead>
<tr>
<th>dense block</th>
<th>input PPGs, FC-1024, ReLU, Dropout</th>
</tr>
</thead>
<tbody>
<tr>
<td>combine layer</td>
<td>dense output + speaker embedding (z)</td>
</tr>
<tr>
<td>dense block</td>
<td>FC-1024, ReLU, Dropout</td>
</tr>
<tr>
<td>recurrent layer</td>
<td>GRU-1024, Dropout</td>
</tr>
<tr>
<td>output layer</td>
<td>FC-(D_x)</td>
</tr>
</tbody>
</table>

The network architectures of our encoder and decoder models.

**EXPERIMENT: SPEAKER SIMILARITY**

- 50 listeners listen and rate A and B with score ranging from +2 (definitely same) to -2 score (definitely different)
- The results show proposed and FHVAE achieving 0.20±0.11 and -0.10±0.12
- The proposed model performs statistically significantly better than FHVAE in all comparison pairs

**EXPERIMENT: SPEECH QUALITY**

- We show the speech quality Comparative Mean Opinion Score (CMOS) in which 50 listeners score which sample quality is better by using +2 (much better) to -2 (much worse) score.
- We found statistically significant preference scores for F2M condition.
- We did not find statistically significant preference scores for other conditions.

**Visualization of speaker embedding:** Blue dots are male speakers and red dots are females. FHVAE (left) vs. Proposed (right).

**EXPERIMENT:**

- We use Factorized Hierarchical Variational Autoencoder (FHVAE) [1] as baseline.
- We observe that:
  - The proposed model’s computed speaker embeddings for different speakers fall further apart compared to FHVAE.
  - Also they are more evenly distributed compared to VAEs which tend to be more densely distributed.
  - The gender clusters have a better separation margin.
- This subjectively depicts a more robust speaker embedding quality.
- The voice conversion samples are available at:
  - https://shamidreza.github.io/is19samples