1. Introduction

In recent years, several end-to-end neural text-to-speech models have significantly improved the quality and naturalness of synthesized speech [1, 2, 3, 4, 5]. However, the style of speech synthesized by these models is strongly coupled with the training dataset, which typically converges to a neutral prosody, affecting the expressiveness and naturalness of synthesized speech.

Hence, neural prosody transfer (PT) techniques were introduced that added style information to the synthesized audio from a reference audio. Several approaches were proposed for this task. For example, Skerry-Ryan et al. [6] conditioned the model with a style encoder to capture sentence-level time-independent features like emotion and style. This method is called the coarse-fined prosody transfer (CPT) model. [7, 8, 9] followed a similar methodology. Based on the CPT models, Hsu et al. [10] proposed a fine-grained prosody transfer model to focus on low-level time-dependent features such as stress, rhythm, melody, etc. This methodology was followed by [11, 12, 13]. However, both methodologies extract a latent style representation from a mel-spectrogram without supervision. The mel-spectrogram, however, is both the input of the style encoder and the target of training. That is, the style encoder might encode some textual information if its capacity exceeds its mission requirement. This phenomenon is called content leakage [14], and it results in flaws in the synthesized audio, including blurred or missed words. Liu et al. [15] used an ASR model to judge the training output, forcing the GST-Tacotron model [7] to learn to resolve unpaired text and reference audio with an additional ASR loss. Gururani et al. [16] decomposed the reference speech into several low-dimensional features as the input to the style encoder instead of mel-spectrograms. Hu et al. [14] attempted to minimize the mutual information between style embedding and text embedding to purify the former.

Content leakage is a challenge in many other research fields involving style transfer. Song et al. [17] investigated this problem in several natural language processing (NLP) models, indicating the surplus ability of modern embedding models to express. Ridgeway et al. [18] used an additional Leakage Filtering Loss to penalize a one-shot style transfer model for leaked content during training. Roy et al. [19] used a new method called Maximum Entropy Adversarial Representation Learning to minimize information leakage about a given sensitive attribute, exploring the potential of adversarial learning. Similarly, Hu et al. [14] also proved the validity of minimizing mutual information. Generally, these methods aim to tackle this problem by adding supervision; however, the supervisory components are computationally taxing and possibly inaccurate.

We proposed a new technique called Information Sieve, which can address content leakage while retaining the simplicity of the model and similarity of the transferred prosody with only a reconstruction loss, inspired by a voice conversion system called AutoVC [20]. The proposed system’s workflow is the same as that of GST-Tacotron, except for a new attachment to its style encoder. Specifically, we used a downsample–upsample structure to filter unwanted textual information from the reference output to condition the vanilla Tacotron [1]. Speech recovered from generated mel-spectrogram sounds natural and is as expressive as the reference speech. Objective evaluations demonstrate the effective mitigation of content leakage. The main contributions of this study are summarized below.

1. We demonstrate the capacity of the style encoder is superfluous with similar quality of synthesized audio and lower WER of our models. A Sieve to filter the redundant information without requiring an adequate output dimension for encoders to mitigate the content leakage.
2. We demonstrated that this method helps the GST layer assign more weights to certain tokens, compared with the approximately average distribution of weights by the original GST-Tacotron. This indicates this method helps each style token better capture its own style.
3. We used the modified style encoder’s output, concatenated with text embedding, as inputs for a Tacotron decoder, denoted as information sieve GST (15-GST). Experiments demonstrated the effective mitigation of content leakage with the preservation of its naturalness and expressiveness during prosody transfer.

2. Sieve Prosody Transfer Architecture

This approach leverages prior research on the encoding capacity of embedding models, indicating that the encoders can encode more information than the mission requirement, disturbing the decoder’s workflow. Specifically, the method filters the content encoder of an AutoVC, the inspiration for this approach, using an information sieve. Therefore, we hypothesize that style embedding could leak textual information due to its superfluosity. Further, the imposition of an adequate limit on the style encoder could help generate pure style embeddings, which can better condition the neural expressive TTS models. We test this hypothesis by incorporating an Information sieve style encoder, previously shown effective, into a vanilla Tacotron. The whole structure of the proposed model is illustrated in Figure 1.

To the best of the authors’ knowledge, few researchers have investigated the effectiveness of an information sieve in expressive neural TTS models. The closest to this work is CopyCat[21], which has a similar structure but accepts a VAE structure’s output as the input, with the VAE potentially working as an information bottleneck. Meanwhile, its style encoder accepts mel-spectrograms, source–speaker embeddings, and phoneme embeddings simultaneously as its input, undermining the effectiveness of a pure information sieve.

2.1. Definitions and Model Framework

As general mathematical annotations, upper-case letters such as $X$ refer to random variables or vectors, and lower-case letters to deterministic values or instances of random variables. $X(1 \rightarrow T)$ refers to a random process, with $(1 \rightarrow T)$ denoting a collection of time indices running from 1 to $T$. We used $E_x(\cdot)$ to denote the text encoder, $E_s(\cdot)$ to denote the style encoder, and $D(\cdot, \cdot)$ to denote the decoder receiving output from the two encoders. The inputs to these modules would be different during training and inference.

Training: We denoted the input to the style encoder as $C$ and that to the text encoder as $Y$, with $X$ and $Y$ being parallel to each other. Then, for each pair of input, we have

$$C_1 = E_s(X), S_1 = E_x(Y), \hat{X}_{1 \rightarrow 1} = D(C_1, S_1).$$

Here, $C_1$, $S_1$, and $\hat{X}_{1 \rightarrow 1}$ are random processes. The model learns to reconstruct itself by reconstruction loss, which in turn consists of linear loss and mel loss.

$$\min_{D(\cdot)} = L_{linear, recon} + L_{mel, recon}$$

(2)

The loss is the same with the original GST-Tacotron.

Inferring: During actual synthesis, text $X_1$ is fed into the text encoder, and a random reference audio $Y_2$ is into the style encoder. $X_1$ and $Y_2$ are not parallel to each other. The decoder receives both inputs to synthesize the audio of a certain style.

$$C_1 = E_s(X_1), S_2 = E_x(Y_2), \hat{X}_{1 \rightarrow 2} = D(C_1, S_2).$$

(3)

where $C_1$ and $\hat{X}_{1 \rightarrow 2}$ are both random processes, and $S_2$ is a reference vector extracted by $E_s(\cdot)$.

2.2. Sieve Reference Encoder

We reemphasize our goal to solve content leakage. During the training, the reference encoder learns to extract style from a mel-spectrogram $X$, and the dimension of its output $Z$ controls the volume of information it carries. When dealing with content leakage, we found it problematic to directly reduce the output $Z$ dimension as it worsened the quality of the synthesized audio. While increasing the $Z$ dimension, the leaked content led to apparent errors in the synthesized audio, such as inserted words, missed words, etc. Therefore, we introduce an information sieve, limiting the information flown out from the style encoder to the decoder along the time axis. It compels the decoder to rely on the text encoder to reconstruct the voice content and depend on the sieve style encoder to reconstruct the style.

We downsampled $Z$ along the time axis at a fixed rate $\tau$ to derive $Z^\tau \in \mathbb{R}^{(T/\tau) \times H}$. Then, we upsampled $Z^\tau$ to length $T$ to reconstruct $\hat{Z} \in \mathbb{R}^{T \times H}$. Unlike AutoVC, which downsamples from a bidirectional LSTM unit’s hidden states, this method downsampled from a unidirectional GRU unit’s hidden states based on the assumption that post-order words do not affect pre-ordered words in an auto-regressive model. Hence, we used $\hat{Z}$, which picks out timestamps like $\{\tau-1, 2\tau-1, ..., T-1\}$ to retain key information. The structure is depicted in Figure 2.
2.3. Instance Normalization with Conv2D

It is based on the hypothesis that prosody varies with the channel, while batch normalization will over-smooth the prosody of different audio. Hence, we assume that prosody can be better retained by normalizing it with each channel’s mean and standard deviation but not the batch’s. Therefore, we normalized a set of two-dimensional (2D) convolutional layers by expressing the mean and deviation as follows:

\[
\mu_{ti} = \frac{1}{HW} \sum_{w=1}^{W} \sum_{h=1}^{H} x_{twh}, \forall w \in W, \forall h \in H
\]

\[
\sigma_{ti}^2 = \frac{1}{HW} \sum_{l=1}^{W} \sum_{m=1}^{H} (x_{lwh} - \mu_{ti})^2, \forall w \in W, \forall h \in H
\]

where \( H \) is the length of the time axis input, \( W \) is the length of the axis of the mel channels, \( w \) is the \( w \)th step for a kernel along the time axis, and \( h \) is the \( h \)th step along the mel-channel axis. \( t \) is the length of the convolution kernel along the mel-channel axis, \( i \) is the length of the kernel along the mel-channel axis, and \( x \) is the convolutional result defined by \( t, i, w, h \). Then, we normalized each element by

\[
x'_{twh} = \frac{x_{twh} - \mu_{ti}}{\sigma_{ti}}, \forall w \in W, \forall h \in H
\]

We applied instance normalization to all the six convolutional layers of the style encoder to strengthen its effect.

3. Experiments

3.1. Dataset and Preprocessing

The dataset for this study was collected from a public expressive dataset called Blizzard Challenge 2013 (BC2013), which is a professionally recorded, female, single-speaker expressive dataset called Blizzard Challenge 2013 (BC2013), which is a professionally recorded, female, single-speaker expressive dataset.

The dataset for this study was collected from a public expressive dataset called Blizzard Challenge 2013 (BC2013), which is a professionally recorded, female, single-speaker expressive dataset. For the sake of accuracy, we only used its segmented part, containing a total of 9,733 valid (text-utterance) pairs. During the training and inference, we did not incorporate any form of style-related label information.

We divided the 9,733 pairs into 3 parts, 9,550 pairs for training, 50 pairs for validation, and 133 pairs for testing. All the evaluations were conducted on texts from the test set. The mel-spectrograms were computed using 80 mel bins, 50-ms window size, and 12.5-ms hop size. We previously scaled the audio to a range of [-4, 4] in the log-scale. We eliminated all of the silence and paused audio to sound clear. We transformed raw text input into ArpaBet phonemes with ‘sp’ as pause and ‘sil’ as the end of sentences using an internal front-end model.

As described above, we experimented with the vanilla Tacotron combined with the proposed refined style encoder. The IS-GST follows the same training procedure as the standard Tacotron. We set the learning rate to 0.001, batch size to 16, and \( \tau \) to 32. We trained each model in approximately 180,000 steps, which ensures the model converges while preventing overfitting. The conditioning inputs for the training were obtained from the ground-truth audio paired with textual inputs. During inference, a reference can be randomly chosen from the training, validation, and test sets. Users can also manually set the weight for each token. Finally, we used a MelGAN [22] vocoder to generate waveforms from the predicted mel-spectrograms. The vocoder was separately trained on the ground-truth training data from the BC 2013 dataset.

3.2. Results and Evaluation

We evaluated the proposed models for prosody transfer using both subjective listening tests and objective metrics. To further test each modification, we conducted an ablation test and compared the results with the ground-truth audio. We list the sources of the samples below using their abbreviations.

- Ground Truth: Ground-truth audio from the test set.
- Original GST-Tacotron (GST): GST-Tacotron model [7].
- ASR-guided-Tacotron (ASR-G): shown in [15].
- I-G: IST-GST model.
- S-I-G: IST-GST model with an information sieve layer.
- Instance Normalization GST-Tacotron (I-G): Simply replace batch normalization used in GST-Tacotron’s style encoders convolution layers with instance normalization.
- Instance Normalization GST-Tacotron (S-I-G): IST-GST model’s style encoder upgraded with an information sieve layer; meanwhile, its convolution layers use instance normalization instead of batch normalization.

3.2.1. Subjective Evaluations

We begin with a mean opinion score (MOS) evaluation of speech naturalness. We used texts from the test set and generated speech using the aforementioned models compared with the ground-truth test audio. For each MOS test, a total of 20 English-speaking listeners were invited. Each listener was presented with 10 samples from each source and then asked to rate the naturalness of the audio on a scale of 1–5 in intervals of 0.5. The results are listed in Table 1. The information sieve model was rated more natural than the original GST model. The I-G model did not benefit significantly from instance normalization as it could not effectively eliminate mispronunciations. When combined, the positive effects of instance normalization and information sieve were coamplified as to get close to the MOS of the SOTA ASR-G. To view the demonstration.

<table>
<thead>
<tr>
<th>Model</th>
<th>MOS score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Truth</td>
<td>4.22 ± 0.019</td>
</tr>
<tr>
<td>GST</td>
<td>3.54 ± 0.024</td>
</tr>
<tr>
<td>ASR-G</td>
<td>3.93 ± 0.041</td>
</tr>
<tr>
<td>I-G</td>
<td>3.82 ± 0.031</td>
</tr>
<tr>
<td>S-I-G</td>
<td>3.68 ± 0.028</td>
</tr>
<tr>
<td>GST 3.54 ± 0.024</td>
<td>3.91 ± 0.029</td>
</tr>
</tbody>
</table>

Table 1: MOS Naturalness scores for the models in 95% confidence interval.
This figure demonstrates a common phenomenon while we were trying to reproduce the original GST-Tacotron, that is the average weight per token. This is even harder for us to distinguish each token’s unique effect since we incorporate multi-head attention to refine results.

Next, we evaluated the prosody transfer from the reference audio to the synthesized speech for an arbitrary sentence. For convenience, we merely considered the synthesized audio from the S-I-G model and the original GST model to conduct a side-by-side test. We conducted parallel prosody transfer, where texts and audio are one-to-one matched, and unparallel prosody transfer, where texts and audio are randomly paired. We constructed triplets (A, X, Y), where A is the reference audio, and X and Y are utterances synthesized using the same text but by S-I-G and the original GST model, respectively. For each test, 10 English-speaking listeners were invited and presented with ten triplets. They were asked to choose the one more similar to the prosody of the reference audio. Figure 3 illustrates the results of the listening tests. The S-I-G model transfers prosody better than the original model, according to the higher percentage of votes obtained. Because this model is a CPT model without fine-grained features, it is an acceptable result. Moreover, it still surpasses the original GST model.

3.2.2. Objective Evaluations

The objectives of this model were to mitigate content leakage in the original GST Tacotron and make the tokens in the GST layer more specific and orthogonal to one another for better style control. Hence, we conducted two objective evaluations.

1. Compute the WER and word information lost (WIL) [23] for all the models trained.
2. Use audio of different styles to demonstrate the orthogonality of the style tokens.

For the first experiment, we selected 50 sentences from the test set, the ground-truth audio of which were 3–7 s long, to compute the WER. A reference neutral prosody was used uniformly for all models and the 50 sentences. The results are listed in Table 2. Clearly, the WERs of S-G and S-I-G models were merely 30% of that of the original GST model, whereas that of the I-G model was approximately the same with the original GST models, proving that instance normalization does not contribute to mitigating content leakage. This result proves the effectiveness of the information sieve. To further prove the accuracy of this model, we computed the WIL, which reflects similar results. In the second experiment, we selected several pieces of audio of the Strong style to demonstrate the weight of tokens per reference audio. Compared with the almost equal weight distribution of the original GST-Tacotron in Figure 4, the information sieve model occasionally demonstrates a stronger allocation preference for certain tokens, assigning much more weight than other inactivated tokens when accepting different pieces of reference audio, which helps distinguish each token shown in Figure 5.

4. Conclusions

We demonstrated a simple yet effective method that mitigates content leakage while still synthesizing highly natural, expressive speech and conducted style transfer. By adding a simple information sieve layer on a style encoder to condition a neural TTS model, we could transfer prosody from the reference audio, or a manual combination of several tokens, to any synthesized speech. The sieve successfully tackled content leakage while preserving key style information. Furthermore, we shall focus on searching for better parameters to control the sieve to accurately retain useful information while rejecting the rest for any given mission and building a mathematical theory to better explain the sieve structure.

5. Acknowledgements

This work was supported by the National Key R&D Program of China under Grant 2018YFB1305200, the Tianjin Municipal Science and Technology Project under Grant 18ZXZNGX00330 and the National Natural Science Foundation of China under Grant 61771333.
6. References


