Expressive Text-to-Speech with Pitch and Rhythm From Inferred Style Embeddings

Arsh Zahed

Spring 2020

Abstract

We present Mellotron TPP-GST, an extension to Mellotron, that adds Text-Pitch Predicted Global-Style Tokens (TPP-GST) during inference to predict global style of utterances from text alone. We train a network to predict global style tokens using embeddings of text and fundamental frequency contours. With this approach, we are able to generate an expressive text-to-speech model that is able to follow arbitrary pitch contours and rhythms, without needing any reference utterances or random sampling of style tokens during inference or singing data during training. This additionally seems to outperform methods that require random sampling of style tokens during inference time.

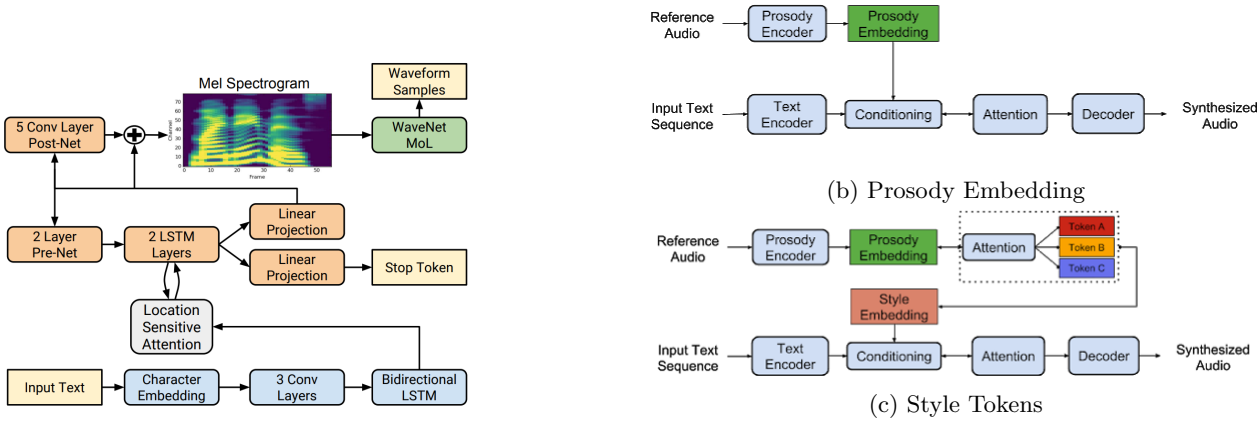
1 Introduction

Current state-of-the-art deep learning voice synthesis consists of two steps. First, input features, such as text, are passed through a neural network to generate a mel-spectrum. This is then sent to a second model, known as a neural vocoder, transforming the mel-spectrum into a speech audio. Recent research on neural vocoders has focused on improving computational efficiency, leading to Waveglow (3), the current state-of-the art neural vocoder. On the other hand recent work in the use of Deep Learning for Text-To-Speech (TTS) has focused on generating mel-spectrums that exhibit style and tonality when propagated through a neural vocoder. The introduction of Tacotron 2(6), a model that takes text input and generates a mel-spectrum, has inspired expressive and realistic TTS models of recent history.

Mellotron (1), an extension of Tacotron 2, introduces speaker identity, F0 contours and rhythmic alignment as features. This allows for granular control of pitch and rhythm, allowing the model to generate singing utterances without having any singing vocals in the dataset. However, it ignores other factors of tonality, such as emphasis and prosody. TP-GST (4), a separate state-of-the-art method for stylistic voice synthesis, improves generated speech by predicting the Global Style Tokens used in Tacotron 2 to better inform generated style tokens during inference time. Traditionally, Tacotron 2 embeds the prosody of training data and forms a set of (usually 10) style tokens. Then the style embedding of any utterance is represented through a learnt convex combination of those style tokens. However, this either requires access to the ground-truth audio to generate, or must be randomly sampled. TP-GST uses a set of neural networks to predict the embedding from speech alone, bypassing the need for audio and avoiding the drawbacks of randomly sampling. Inspired by the advancements of both of these works, we augment Mellotron with the innovations of TP-GST to provide granular control over pitch and rhythm, while also maintaining realistic prosody.

2 Related Works

Within recent history, there have been several recent works in the field of expressive voice sythesis and TTS. In this section we outline the most important works related to Mellotron TPP-GST. We encourage interested readers to read the referenced papers to find additional related works.



(a) Tacotron 2 model

(b) Prosody Embedding

(c) Style Tokens

Figure 1: Tacotron 2 and Style Embedding Extensions

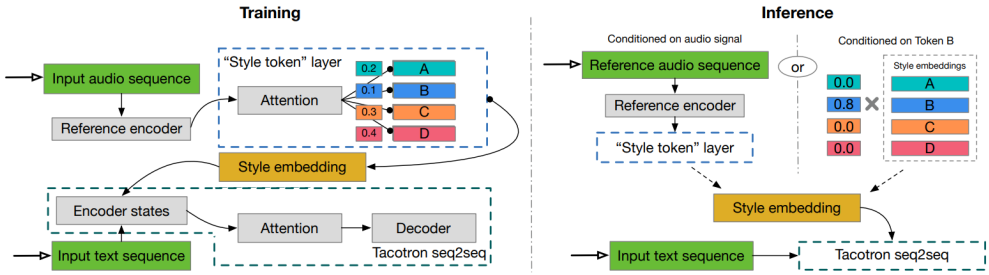
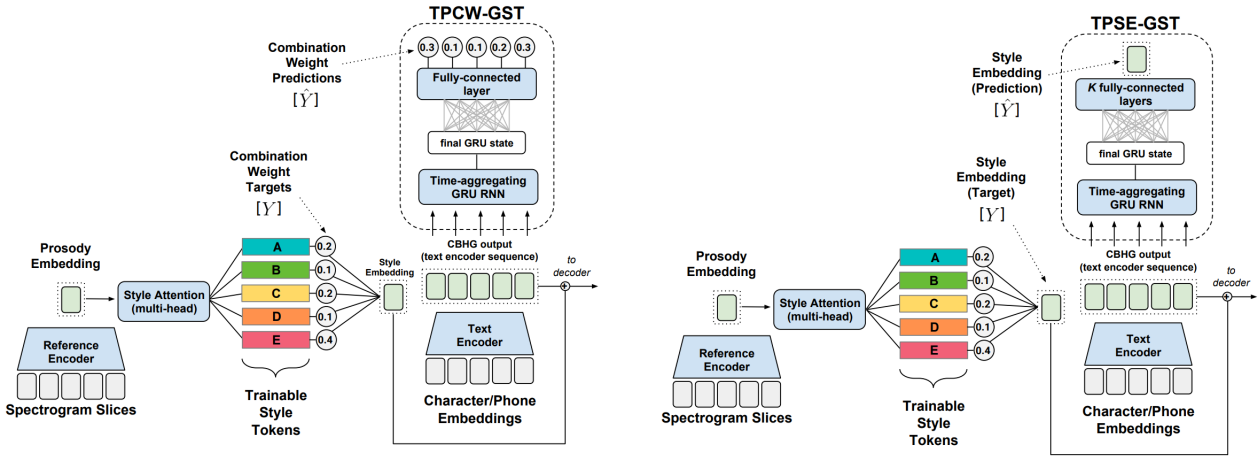


Figure 2: Global Style Tokens. During training, the ground-truth audio is used as the input to train the reference encoder (which generates the prosody embedding) and the attention mechanism. During inference, an audio sample with similar length and style to the desired result can be used as reference, or the style embedding can be picked.

2.1 Tacotron 2 and Style Tokens

Nearly all recent advancements related to TTS have involved Tacotron 2 (6), as seen in Figure 4c. Tacotron2 is a seq2seq model. In the encoder, the input text is embedded, then fed through three 5-D convolutional layers, with the goal to improve receptive field and recognize longer contexts (e.g. N-grams). This is then passed through a bidirectional LSTM. The output of the encoder is sent through an attention mechanism, which is then passed into the decoder. The decoder consists of two recurrent layers, a post-net and a pre-net, involving fully connected and convolutional layers. The result is a mel-spectrogram that can be used by a neural vocoder to generate an audio signal.

By itself, Tacotron 2 does dont explicitly aim to encourage tonality or prosody. To do this, Prosody Embeddings (7) and Style Embeddings (2) are introduced. As seen if Figures 4d and 1c, a reference audio is passed through a network to generate a prosody embedding. Initial work used this directly as an input to the attention layer between the encoder and decoder (7). To build on this, global style tokens can be introduced (2). These are vectors that represent a global space of possible style and are learnt through an attention mechanism from the prosody embedding. A convex combination of the style tokens is used to represent the style embedding for a specific utterance. Thus, it the style embedding is a representation of style for a specific utterance projected onto the space spanned by the tokens. As before, a reference audio can be used to generate the style embedding, or the style embedding can be hand-picked or generated randomly by sampling from a random normal with dimensionality equivalent to the number of global style tokens (usually 10), applying a softmax and using the result as weights for the each token. This process can be seen in



(a) TPCW-GST

(b) TPSE-GST

Figure 3: TP-GST Forms

Figure 2.

2.2 Text Predicted Global Style Tokens (TP-GST)

In the previously discussed architecture, generating style embedding during inference time is problematic. Users are likely to not have access to an appropriate reference audio or the ability to hand-pick style embedding for every utterance. Randomly picking style-embeddings simply imparts a random style, which can be undesirable, or at times even destroy the generated audio. To tackle this, TP-GST introduces a module to predict style embeddings based off text (4).

To form predictions, the result of the text embedding is first passed through a Gated Recurrent Unit (GRU) to form a summary vector. This turns the variable length text into a fixed-length vector. Then there are two options. The first is Text Predicted Combination Weights (TPCW), which predicts the weights to combine the Global Style Tokens. In TPCW, the result of the GRU is sent through a single fully-connected layer, whose outputs are treated as logits over the global style tokens. The loss is then the cross-entropy between the logits and the ground-truths weights. This guarantees that the generated style embedding will be in the space of possible style-embedding achievable by training data. The other option is Text Predicted Style Embedding, which directly predicts the style embedding, skipping the combination weights. This does not guarantee that resulting style embedding is a convex-combination of of the style tokens, but tends to work just as well. In TPSE, the result of the GRU is sent through a k-layer fully connected network, with ReLU activations in the hidden layers and a tanh activation in the output layer. The loss here is the L_1 distance between the predicted embedding and ground-truth embedding. Both methods can be seen in Figure 3, but Mellotron TPP-GST will build off TPSE-GST. It is worth noting that TP-GST calculates the gradient for the new module separately from the gradient of the the rest of the network, thus the module can be detached at any point and training with the module produces an equivalent Tacotron model.

2.3 Mellotron

Mellotron is another extension of Tacotron 2. While TP-GST controls the global style of the generated utterance, Mellotron aims to control granular style of the generated utterance by including fundamental frequency contours (hereby known as pitch contours) and rhythmic alignment. In particular, a data point i consists of a mel-spectrogram $\text{mel}^{(i)}$, text $T^{(i)}$, speaker identity $S^{(i)}$, pitch contour $P^{(i)}$, rhythm allignment $R^{(i)}$. Then let $Z_{\text{mel}^{(i)}}$ be the ground-truth style embedding of the mel-spectrogram. Additionally, let θ be the model parameters of Mellotron. Mellotron then attempts to maximize the log-likelihood of the training

data set.

$$\max_{\theta} \sum_{i \in \mathcal{D}} \log P\left(\text{mel}^{(i)} \;\middle|\; T^{(i)}, S^{(i)}, P^{(i)}, R^{(i)}, Z_{\text{mel}^{(i)}} \;;\; \theta\right) \tag{1}$$

The architecture is nearly identical to that of Tacotron 2, except the speaker identity is concatenated to encoder outputs, and the pitch contour is passed through a convolutional layer and then channel-wise concatenated to decoder inputs.

3 Method

Mellotron TPP-GST is a combination of Mellotron and TP-GST (specifically TPSE-GST). We preserve all the same inputs as Mellotron, but additionally train a module to predict style embeddings. However, as opposed to simply predicting style embeddings from text, we additionally introduce pitch contours as feature to predict the sptyle embedding, hence the name Text-Pitch Predicted Gloabal Style Tokens. Letting f_ϕ represent the model that predicts style tokens, we tackle the joint optimization problem

$$\max_{\phi,\theta} \sum_{i \in \mathcal{D}} \log P\left(\text{mel}^{(i)} \;\middle|\; T^{(i)}, S^{(i)}, P^{(i)}, R^{(i)}, Z_{\text{mel}^{(i)}} \;;\; \theta\right) + \left\| Z_{\text{mel}^{(i)}} - f_\phi\left(T^{(i)}, P^{(i)}\right) \right\|_1 \tag{2}$$

Note that from this equation, the gradient of the loss with respect to ϕ is independent of the gradient of the loss with respect to θ, thus the addition of this module does not affect training. This also means we can use a pre-trained version of Mellotron, or swap out Mellotron for any potential future model that uses style embeddings. During inference, we can simply replace $Z_{\text{mel}^{(i)}}$ with the predicted style embedding and sample from the distribution defined by Mellotron

$$\widehat{\text{mel}} \sim P\left(\cdot \;\middle|\; T, S, P, R, f_\phi(T, P) \;;\; \theta\right) \tag{3}$$

With this method, no reference audio or random sampling is required at inference. Instead, the style token is determined solely from the input text and pitch contour. This allows for larger range of applications of the system, while still maintaining meaningful prosody and style.

To be more precise on how define f_ϕ, the embedded text and pitch contour are used as features. Both are sent through independent bi-directional GRU layers with 64-units each. The final outputs of both are taken and concatenated. This is then passed through k fully connected layers (we use 4 hidden layers with 512 units each) with ReLU activation for all hidden layers and an output layer with tanh activation. Thus, for k layers, the TPP-GST model can be described as

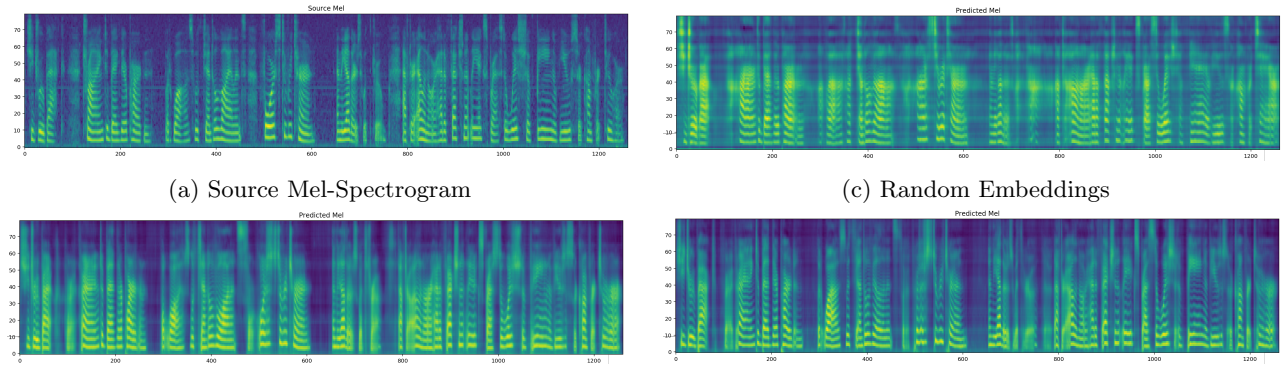
$$T_G = \text{GRU}_1(T)_{[-1]} \tag{4}$$
$$P_G = \text{GRU}_2(P)_{[-1]} \tag{5}$$
$$E_0 = T_G || P_G \tag{6}$$
$$E_i = \text{ReLU}(W_i E_{i-1} + b_i), \quad i \in \{1, \ldots, k-1\} \tag{7}$$
$$f_\phi(T, P) = \tanh(W_k E_{k-1} + b_k) \tag{8}$$

where $X_{[-1]}$ represents the last column of the matrix X and $||$ indicates concatenation.

4 Results

4.1 Tacotron 2 and Style Tokens

When calculating the Mel Cepstral Distortion, we find that mel-spectrogram conditioned style embeddings, TPP-GST predicted embeddings, and randomly sampled embeddings all lead to equivalent results. This is likely due to the complex nature of speech. As can be seen in Figure 4, the generated mel-spectrograms are comparable. Often, seemingly similar speech clips can have large magnitude differences, and vice versa. For this reason, most literature in the field relies on Mean Opinion Score (MOS). However, we currently do not



(a) Source Mel-Spectrogram

(c) Random Embeddings

(b) Mel-Spectrogram Conditioned Embeddings

(d) TPP-GST Embeddings

Figure 4: Source and Generated Mel-Spectrograms

have the resources or time to conduct such a study. Thus, we subjectively inspect results.

Additionally, we would like to note that the model was not able to train for long enough to draw any solid conclusions, positive or negative. We trained on a GTX 1080 with 8 GB of VRAM and 16 GB of RAM. This setup proved to be rather weak for the collective training of both the TPP-GST Module and Mellotron simultaneously, and would require several days of training to reach convergence. Thus, we were unable to reach even 1 epoch of training prior to stopping our training process. Because of this, we encourage readers to take all results with a grain of salt. We plan to further train the model in the future, as well as conduct a thorough hyperparameter search. Additionally, we plan to implement a system to use F0 Frame Error as a metric in the future, as this is also standard in literature.

We trained our model for 8 hours on the LibriTTS dataset. We were able to reach 2000 iterations within this time. Example generations can be found at here[1]. For each utterance, we generate samples for the mel-spectrogram conditioned style embeddings, TPP-GST predicted embeddings, and randomly sampled embeddings. The mel-spectrogram condition embeddings represent the goal result. The mel-spectrogram captures significantly more information than text and pitch contours alone. As can be heard, randomly sampled embeddings often lead to undesirable prosody and style in the generated speech. This can range from a raspy voice to boosted low frequencies. On the other hand, the TPP-GST generations are signficantly more consistent in prosody and style. While it does not seem to be equivalent to conditoin on the ground truth mel-spectrogram, we anticipate it to get much closer with further training. Additionally, we would like to mention that all generations are ill-formed; this is due to the lack of training of time.

5 Conclusion

In this work, we showed that by incorporating both the granular style control of Mellotron, and the global style of TP-GST, we can generate much more realistic and expressive speech, while not requiring any reference audio or random sampling during inference time. This allows for a much wider set of applications for the Mellotron model, as most users will not readily have appropriate reference audio or the time and/or ability to hand-pick style embeddings. Through subjective review, Mellotron TPP-GST seems to outperform random sampling, and come close to using reference audio, while only taking text and pitch contour as inputs. Lastly, the TPP-GST module can be swapped out to work with any new model that uses the same style tokens.

In the future, we plan to train the model further on a more sufficient computational setup, tune hyperparameters, implement the F0 Frame Error Metric to evaluate our performance, and conduct a Mean Opinion Score study on the generations of the Mellotron TPP-GST compared to its counterparts. Additionally,

[1] https://drive.google.com/drive/folders/1mdOmgsPkbbFlSsHmpkcauU8RJcxllcrn?usp=sharing

we hope to conduct a wider meta-study to analyze the fail cases of state-of-the-art text-to-speech models on specific phonemes, words, vowels and consonants, as we have not seen any such study when reviewing literature. All code pertaining to this project can be found here[2].

References

[1] Rafael Valle, Jason Li, Ryan Prenger, Bryan Catanzaro *"Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens"*. 2019, arXiv:1910.11997

[2] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ SkerryRyan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A Saurous, *"Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis,"* 2018, arXiv preprint arXiv:1803.09017, 2018

[3] Ryan Prenger, Rafael Valle, and Bryan Catanzaro, *"Waveglow: A flow-based generative network for speech synthesis,"* in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 3617–3621.

[4] D. Stanton, Y. Wang and R. Skerry-Ryan, *"Predicting Expressive Speaking Style from Text in End-To-End Speech Synthesis,"* 2018 IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 2018, pp. 595-602.

[5] Oleksii Kuchaiev and Jason Li and Huyen Nguyen and Oleksii Hrinchuk and Ryan Leary and Boris Ginsburg and Samuel Kriman and Stanislav Beliaev and Vitaly Lavrukhin and Jack Cook and Patrice Castonguay and Mariya Popova and Jocelyn Huang and Jonathan M. Cohen, *"NeMo: a toolkit for building AI applications using Neural Modules,"* 2019, arxiv:1909.09577

[6] 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)*. IEEE, 2018, pp. 4779–4783.

[7] Skerry-Ryan, RJ, Battenberg, Eric, Xiao, Ying, Wang, Yuxuan, Stanton, Daisy, Shor, Joel, Weiss, Ron J., Clark, Rob, and Saurous, Rif A. *"Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron,"*. arXiv preprint, 2018.

[2] https://github.com/azahed98/mellotron