Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis

Yuxuan Wang [1] **Daisy Stanton** [1] **Yu Zhang** [1] **RJ Skerry-Ryan** [1] **Eric Battenberg** [1] **Joel Shor** [1] **Ying Xiao** [1] **Fei Ren** [1] **Ye Jia** [1] **Rif A. Saurous** [1]

Abstract

In this work, we propose "global style tokens" (GSTs), a bank of embeddings that are jointly trained within Tacotron, a state-of-the-art end-to-end speech synthesis system. The embeddings are trained with no explicit labels, yet learn to model a large range of acoustic expressiveness. GSTs lead to a rich set of significant results. The soft interpretable "labels" they generate can be used to control synthesis in novel ways, such as varying speed and speaking style – independently of the text content. They can also be used for style transfer, replicating the speaking style of a single audio clip across an entire long-form text corpus. When trained on noisy, unlabeled found data, GSTs learn to factorize noise and speaker identity, providing a path towards highly scalable but robust speech synthesis.

1. Introduction

The past few years have seen exciting developments in the use of deep neural networks to synthesize natural-sounding human speech (Zen et al., 2016; van den Oord et al., 2016; Wang et al., 2017a; Arik et al., 2017; Taigman et al., 2017; Shen et al., 2017). As text-to-speech (TTS) models have rapidly improved, there is a growing opportunity for a number of applications, such as audiobook narration, news readers, and conversational assistants. Neural models show the potential to robustly synthesize expressive long-form speech, and yet research in this area is still in its infancy.

To deliver true human-like speech, a TTS system must learn to model prosody. Prosody is the confluence of a number of phenomena in speech, such as paralinguistic information, intonation, stress, and style. In this work we focus on *style modeling*, the goal of which is to provide models the capability to choose a speaking style appropriate for the given context. While difficult to define precisely, style contains rich information, such as intention and emotion, and influences the speaker's choice of intonation and flow. Proper stylistic rendering affects overall perception (see e.g. "affective prosody" in (Taylor, 2009)), which is important for applications such as audiobooks and newsreaders.

Style modeling presents several challenges. First, there is no objective measure of "correct" prosodic style, making both modeling and evaluation difficult. Acquiring annotations for large datasets can be costly and similarly problematic, since human raters often disagree. Second, the high dynamic range in expressive voices is difficult to model. Many TTS models, including recent end-to-end systems, only learn an averaged prosodic distribution over their input data, generating less expressive speech especially for long-form phrases. Furthermore, they often lack the ability to control the expression with which speech is synthesized.

This work attempts to address the above issues by introducing "global style tokens" (GSTs) to Tacotron (Wang et al., 2017a; Shen et al., 2017), a state-of-the-art end-to-end TTS model. GSTs are trained without any prosodic labels, and yet uncover a large range of expressive styles. The internal architecture itself produces soft interpretable "labels" that can be used to perform various style control and transfer tasks, leading to significant improvements for expressive long-form synthesis. GSTs can be directly applied to noisy, unlabeled found data, providing a path towards highly scalable but robust speech synthesis.

2. Model Architecture

Our model is based on Tacotron (Wang et al., 2017a; Shen et al., 2017), a sequence-to-sequence (seq2seq) model that predicts mel spectrograms directly from grapheme or phoneme inputs. These mel spectrograms are converted to waveforms either by a low-resource inversion algorithm (Griffin & Lim, 1984) or a neural vocoder such as WaveNet (van den Oord et al., 2016). We point out that, for Tacotron, the choice of vocoder does not affect prosody, which is

[1]Google, Inc.. Correspondence to: Yuxuan Wang <yxwang@google.com>.

Sound demos can be found at https://google.github.io/tacotron/publications/global_style_tokens

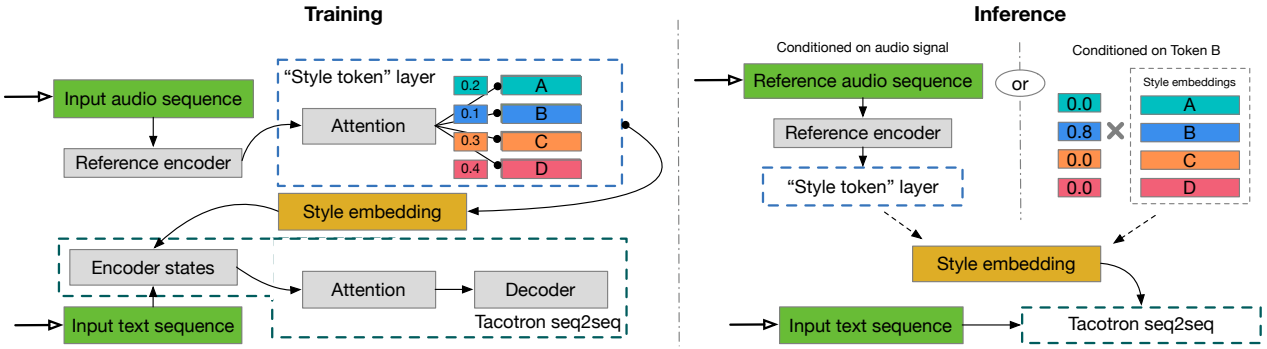


Figure 1. Model diagram. During **training**, the log-mel spectrogram of the training target is fed to the reference encoder followed by a style token layer. The resulting style embedding is used to condition the Tacotron text encoder states. During **inference**, we can feed an arbitrary reference signal to synthesize text with its speaking style. Alternatively, we can remove the reference encoder and directly control synthesis using the learned interpretable tokens.

modeled by the seq2seq model.

Our proposed GST model, illustrated in Figure 1, consists of a reference encoder, style attention, style embedding, and sequence-to-sequence (Tacotron) model.

2.1. Training

During training, information flows through the model as follows:

- The **reference encoder**, proposed in (Skerry-Ryan et al., 2018), compresses the prosody of a variable-length audio signal into a fixed-length vector, which we call the *reference embedding*. During training, the reference signal is ground-truth audio.
- The reference embedding is passed to a style token layer, where it is used as the query vector to an **attention module**. Here, attention is not used to learn an alignment. Instead, it learns a similarity measure between the reference embedding and each token in a bank of **randomly initialized embeddings**. This set of embeddings, which we alternately call *global style tokens*, GSTs, or token embeddings, are shared across all training sequences.
- The attention module outputs a set of combination weights that represent the contribution of each style token to the encoded reference embedding. The weighted sum of the GSTs, which we call the *style embedding*, is passed to the text encoder for conditioning at every timestep.
- The style token layer is jointly trained with the rest of the model, driven only by the reconstruction loss from the Tacotron decoder. GSTs thus do not require any explicit style or prosody labels.

2.2. Inference

The GST architecture is designed for powerful and flexible control in inference mode. In this mode, information can flow through the model in one of two ways:

1. We can directly condition the text encoder on certain tokens, as depicted on the right-hand side of the inference-mode diagram in Figure 1 ("Conditioned on Token B"). This allows for style control and manipulation without a reference signal.
2. We can feed a different audio signal (whose transcript does not need to match the text to be synthesized) to achieve style transfer. This is depicted on the left-hand side of the inference-mode diagram in Figure 1 ("Conditioned on audio signal").

These will be discussed in more detail in Section 6.

3. Model Details

3.1. Tacotron Architecture

For our baseline and GST-augmented Tacotron systems, we use the same architecture and hyperparameters as (Wang et al., 2017a) except for a few details. We use phoneme inputs to speed up training, and slightly change the decoder, replacing GRU cells with two layers of 256-cell LSTMs; these are regularized using zoneout (Krueger et al., 2017) with probability 0.1. The decoder outputs 80-channel log-mel spectrogram energies, two frames at a time, which are run through a dilated convolution network that outputs linear spectrograms. We run these through Griffin-Lim for fast waveform reconstruction. It is straightforward to replace Griffin-Lim by a WaveNet vocoder to improve the audio fidelity (Shen et al., 2017).

The baseline model achieves a 4.0 mean opinion score

(MOS), outperforming the 3.82 MOS reported in (Wang et al., 2017a) on the same evaluation set. It is thus a very strong baseline.

3.2. Style Token Architecture

3.2.1. Reference Encoder

The reference encoder is made up of a convolutional stack, followed by an RNN. It takes as input a log-mel spectrogram, which is first passed to a stack of six 2-D convolutional layers with 3×3 kernel, 2×2 stride, batch normalization and ReLU activation function. We use 32, 32, 64, 64, 128 and 128 output channels for the 6 convolutional layers, respectively. The resulting output tensor is then shaped back to 3 dimensions (preserving the output time resolution) and fed to a single-layer 128-unit unidirectional GRU. The last GRU state serves as the reference embedding, which is then fed as input to the style token layer.

3.2.2. Style Token Layer

The style token layer is made up of a bank of style token embeddings and an attention module. Unless stated otherwise, our experiments use 10 tokens, which we found sufficient to represent a small but rich variety of prosodic dimensions in the training data. To match the dimensionality of the text encoder state, each token embedding is 256-D. Similarly, the text encoder state uses a tanh activation; we found that applying a tanh activation to GSTs before applying attention led to greater token diversity. The content-based tanh attention uses a softmax activation to output a set of combination weights over the tokens; the resulting weighted combination of GSTs is then used for conditioning. We experimented with different combinations of conditioning sites, and found that replicating the style embedding and simply adding it to every text encoder state performed the best.

While we use content-based attention as a similarity measure in this work, it is trivial to substitute alternatives. Dot-product attention, location-based attention, or even combinations of attention mechanisms may learn different types of style tokens. In our experiments, we found that using multi-head attention (Vaswani et al., 2017) significantly improves style transfer performance, and, moreover, is more effective than simply increasing the number of tokens. When using h attention heads, we set the token embedding size to be $256/h$ and concatenate the attention outputs, such that the final style embedding size remains the same.

4. Model Interpretation

4.1. End-to-End Clustering/Quantization

Intuitively, the GST model can be thought of as an end-to-end method for decomposing the reference embedding into a set of basis vectors or soft clusters – i.e. the style tokens. As mentioned above, the contribution of each style token is represented by an attention score, but can be replaced with any desired similarity measure. The GST layer is conceptually somewhat similar to the VQ-VAE encoder (van den Oord et al., 2017), in that it learns a quantized representation of its input. We also experimented with replacing the GST layer with a discrete, VQ-like lookup table layer, but have not seen comparable results yet.

This decomposition concept can also be generalized to other models, e.g. the factorized variational latent model in (Hsu et al., 2017), which exploits the multi-scale nature of a speech signal by explicitly formulating it within a factorized hierarchical graphical model. Its sequence-dependent priors are formulated by an embedding table, which is similar to GSTs but without the attention-based clustering. GSTs could potentially be used to reduce the required samples to learn each prior embedding.

4.2. Memory-Augmented Neural Network

GST embeddings can also be viewed as an external memory that stores style information extracted from training data. The reference signal guides memory writes at training time, and memory reads at inference time. We may leverage recent advances from memory-augmented networks (Graves et al., 2014) to further improve GST learning.

5. Related Work

Prosody and speaking style models have been studied for decades in the TTS community. However, most existing models require explicit labels, such as emotion or speaker codes (Luong et al., 2017). While a small amount of research has explored automatic labeling, learning is still supervised, requiring expensive annotations for model training. AuToBI, for example, (Rosenberg, 2010) aims to produce ToBI (Silverman et al., 1992) labels that can be used by other TTS models. However, AuToBI still needs annotations for training, and ToBI, as a hand-designed label system, is known to have limited performance (Wightman, 2002).

Cluster-based modeling (Eyben et al., 2012; Jauk, 2017) is related to our work. Jauk (2017), for example, uses i-vectors (Dehak et al., 2011) and other acoustic features to cluster the training set and train models in different partitions. These methods rely on a complex set of hand-designed features, however, and require training a neutral voice model in a separate step.

As mentioned previously, (Skerry-Ryan et al., 2018) introduces the reference embedding used in this work, and shows that it can be used to transfer prosody from a reference signal. This embedding does not enable interpretable style control, however, and we show in Section 6 that it

generalizes poorly on some style transfer tasks.

Our work substantially extends the research in (Wang et al., 2017b), but there are several fundamental differences. First, (Wang et al., 2017b) uses a single frame from the Tacotron *decoder* as the query to learn tokens. It thus only models "local" variations that primarily correspond to F0. GSTs instead use a summary of the entire reference signal as input, and are thus able to uncover both local and global attributes that are essential for expressive synthesis. Second, in contrast to the decoder-side conditioning in (Wang et al., 2017b), the design of GSTs allows textual input to be conditioned on a disentangled style embedding. We show crucial implications of this for style control and transfer in Section 6.2. Finally, GSTs can be applied to both clean recordings and noisy found data. We discuss this and its significance in detail in Section 7.

6. Experiments: Style Control and Transfer

In this section, we measure the ability of GSTs to control and transfer speaking style, using the inference methods from Section 2.2.

We train models using 147 hours of American English audiobook data. These are read by the 2013 Blizzard Challenge speaker, Catherine Byers, in an animated and emotive storytelling style. Some books contain very expressive character voices with high dynamic range, which are challenging to model.

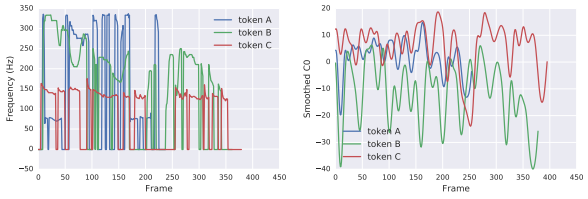
As is common for generative models, objective metrics often do not correlate well with perception (Theis et al., 2015). While we use visualizations for some experiments below, we strongly encourage readers to listen to the samples provided on our demo page.

6.1. Style Control

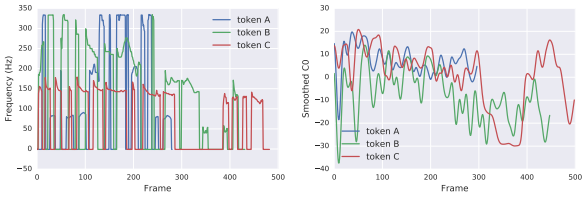
6.1.1. Style Selection

The simplest method of control is conditioning the model on an individual token. At inference time, we simply replace the style embedding with a specific, optionally scaled token.

Conditioning in this manner has several benefits. First, it allows us to examine which style attributes each token encodes. Empirically, we find that each token can represent not just pitch and intensity, but also a variety of other attributes, such as speaking rate and emotion. This can be seen in Figure 2, which shows two sentences synthesized with three different style tokens (scale=0.3) from a 10-token GST model. The plots show that F0 and C0 (energy) curves are quite different across style tokens. However, the F0 and C0 contours generated by each token follow a clear relative trend, despite the fact that input sentences A and B are completely different.



(a) Sentence A



(b) Sentence B

Figure 2. F0 and C0 (log scale) of two different sentences, synthesized using three tokens. Independent of the text content, the same token exhibits the same F0/C0 trend relative to the other tokens.

Indeed, perceptually, the red token corresponds to a lower-pitch voice, the green token to a decreasing pitch, and the blue token to a faster speaking rate (note the total audio duration in both plots).

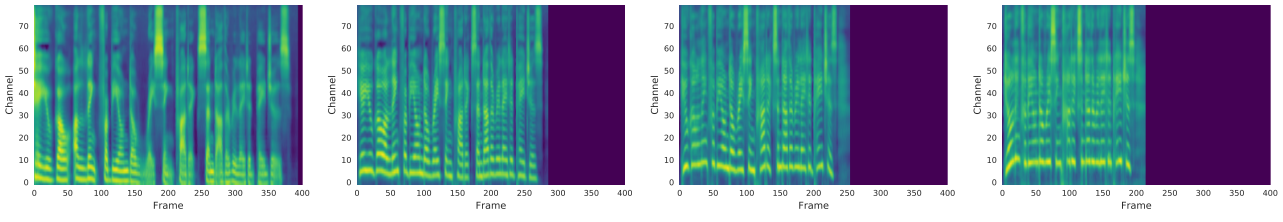
Single-token conditioning also reveals that not all tokens capture single attributes: while one token may learn to represent speaking rate, others may learn a mixture of attributes that reflect stylistic co-occurrence in the training data (a low-pitched token, for example, can also encode a slower speaking rate). Encouraging more independent style attribute learning is an important focus of ongoing work.

In addition to providing interpretability, style token conditioning can also improve synthesis quality. Consider the problem of long-form synthesis on training data with lots of prosodic variation. Many TTS models learn to generate the "average" prosodic style, which can be problematic for expressive datasets, since the very variation that characterizes them is collapsed. This can also lead to undesirable side effects, such as pitch continuously declining towards the end of each sentence. We find that conditioning on "lively"-sounding tokens can address both of these problems, significantly improving the prosodic variation.

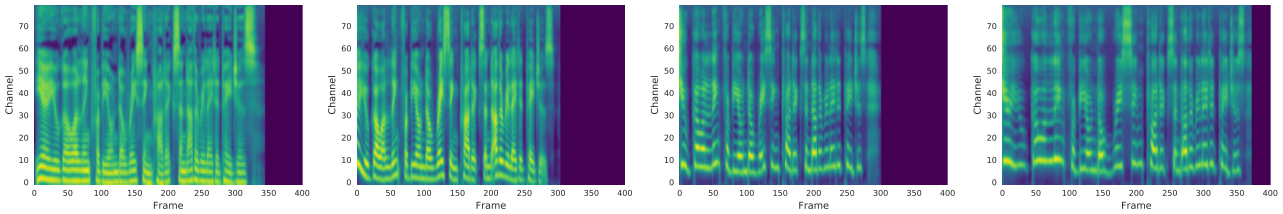
Audio examples of style selection can be found here.

6.1.2. Style scaling

Another method for controlling style token output is via scaling. We find that multiplying a token embedding by a scalar value intensifies its style effect. (Note that large scaling values may lead to unintelligible speech, which suggests future work on improving stability.) This is illustrated in Figure 3, which shows spectrograms of utterances synthesized by two different tokens. Perceptually, these tokens encode

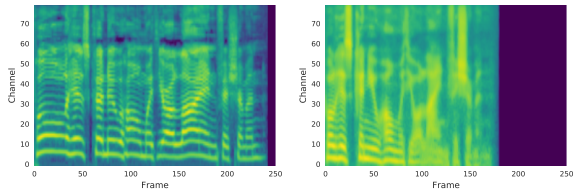


(a) Token A (speed)



(b) Token B (animated)

Figure 3. Effect of token scaling. From left to right, we scale the two tokens by -0.3, 0.1, 0.3, 0.5, respectively. Note that the model seems to exhibit the reverse effect (e.g. fast to slow or animated to calm) with a negative scale, which is never seen during training.



(a) Raw audio (b) Baseline Tacotron

(c) Direct conditioning (d) GST conditioning

Figure 4. Log-mel spectrograms for parallel style transfer.

two different speaking styles: a faster speaking rate (3(a)), and more animated speech (3(b)). Figure 3(a) shows that increasing the scaling factor of the faster speaking rate token causes a gradual compression of the spectrogram in the time domain. Similarly, Figure 3(b) shows that increasing the scaling factor of the animated speech token yields commensurate increases in pitch variation. These style scaling effects hold even for negative values (speaking rate becomes slower, and speech becomes calmer), despite the fact that the model only sees positive (softmax) values during training.

Audio examples of style scaling can be found here.

6.1.3. Style Sampling

We can also control synthesis during inference by modifying the attention module weights inside the style token layer. Since the GST attention produces a set of combination weights, these may be refined manually to yield a desired interpolation. We can also use randomly generated softmax weights to sample the style space. The sampling diversity can be controlled by tuning the softmax temperature.

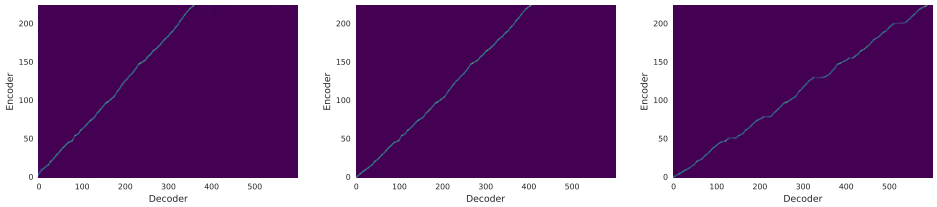
6.1.4. Text-side style control/morphing

While the same style embedding is added to all text encoder states during training, this doesn't need to be the case in inference mode. As our audio samples demonstrate, this allows us to do piecewise style control or morphing by conditioning on one or more tokens for different segments of input text.

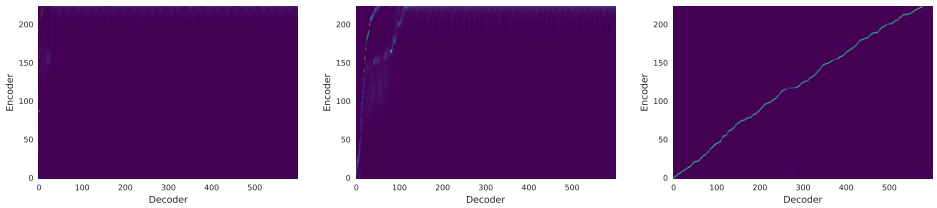
Audio examples of style morphing can be found here.

6.2. Style Transfer

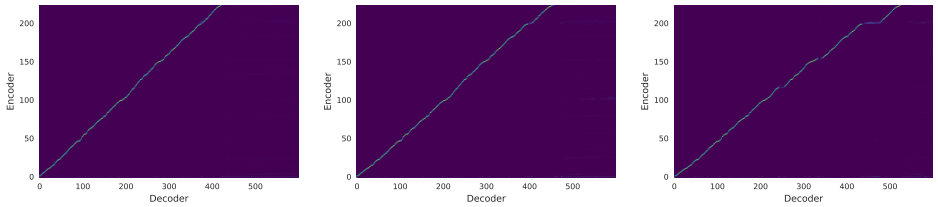
Style transfer is an active area of research that aims to synthesize a phrase in the prosodic style of a reference signal (Wu et al., 2013; Nakashika et al., 2016; Kinnunen et al., 2017). The property that a GST model can be conditioned on any convex combination of style tokens lends itself well to this task; at inference time (method 2 of Section 2.2), we can simply feed a reference signal to guide the choice of token combination weights. The experiments below use 4-head GST attention.



(a) 10-token GST



(b) Direct conditioning (128-D)



(c) 256-token GST

Figure 5. Robustness in non-parallel style transfer. Left to right: attention alignments obtained from feeding three references whose text lengths are 10, 96, 321 characters, respectively. The target text length is 258 characters.

6.2.1. Parallel Style Transfer

Figure 4 shows spectrograms for a *parallel transfer* task, where the text to synthesize matches the text of the reference signal. The GST model spectrogram is at the bottom right, compared to three other baselines: (a) the ground-truth input signal (i.e. the reference); (b) inference performed by a baseline Tacotron model (which infers acoustics only from text); and (c) inference as performed by (Skerry-Ryan et al., 2018), a Tacotron system which conditions the text encoder directly on an 128-D reference embedding.

We see that, given only text input, the baseline Tacotron model does not closely match the prosodic style of the reference signal. By contrast, the direct conditioning method of (Skerry-Ryan et al., 2018) results in nearly time-aligned fine prosody transfer. The GST model is somewhere in between: while its output duration and formant transitions don't precisely match those of the reference, the overall spectrotemporal envelopes do. Perceptually, GSTs resemble the prosodic style of the reference.

Audio examples of parallel style transfer can be found here.

6.2.2. Non-Parallel Style Transfer

We next show results for a *non-parallel transfer* task, in which a TTS system must synthesize arbitrary text in the prosodic style of a reference signal. We chose three different reference signals for this task, and tested how well a GST model replicated each style when synthesizing the same target phrase. Since long-form synthesis can benefit significantly from proper stylistic rendering, we used a long (258-character) target phrase. We chose source phrases of varying lengths (10, 96, and 321 characters, respectively). Figure 5 shows alignment matrices for synthesis conditioned on each source signal.

The top row shows a 10-token GST model. This model robustly generalizes to all three conditioning inputs, as evidenced by the good alignment plots. The bottom row shows a 256-token GST model exhibiting the same behavior; we include this model to show that GSTs remain robust even when the number of tokens (256) is larger than the reference embedding dimensionality (128).

The middle row shows a model with direct reference embedding conditioning. The attention matrices show that this model fails when conditioned on the shorter source phrases, since it tries to squeeze its synthesis into the same time interval as that of the reference. While the model successfully aligns when conditioned on the longest input, intelligibility is poor for some words: the per-utterance embedding captures too much information (such as timing and phonetics) from the source, hurting generalization.

Table 1. SxS subjective preference (%) and p-values of GST audiobook synthesis against a Tacotron baseline. Each row shows GST inference conditioned a different reference signal (A and B). p-values are given for both a 3-point and 7-point rating system.

	PREFERENCE (%)			P-VALUE	
	BASE	NEUTRAL	GST	3-POINT	7-POINT
SIGNAL A	32.9	26.5	40.6	P=0.0552	P=0.0131
SIGNAL B	33.1	21.9	45.0	P=0.0038	P=0.0003

Table 2. Robust MOS as a function of the percentage of interference in the training set. The total training set size is the same.

NOISE %	BASELINE TACOTRON	GST
50%	2.819 ± 0.269	4.080 ± 0.075
75%	1.819 ± 0.227	3.993 ± 0.074
90%	1.609 ± 0.131	4.031 ± 0.082
95%	1.353 ± 0.090	3.997 ± 0.066

To evaluate the quality of this method at scale, we ran side-by-side subjective tests of non-parallel GST style transfer against a Tacotron baseline. We used an evaluation set of 60 audiobook sentences, including many long phrases. We generated two sets of GST output by conditioning the model on two different narrative-style reference signals, unseen during training. A side-by-side subjective test indicated that raters preferred both sets of GST synthesis against a Tacotron baseline, as shown in Table 1.

The performance of GSTs on non-parallel style transfer is significant, since it allows using a source signal to guide robust stylistic synthesis of arbitrary text.

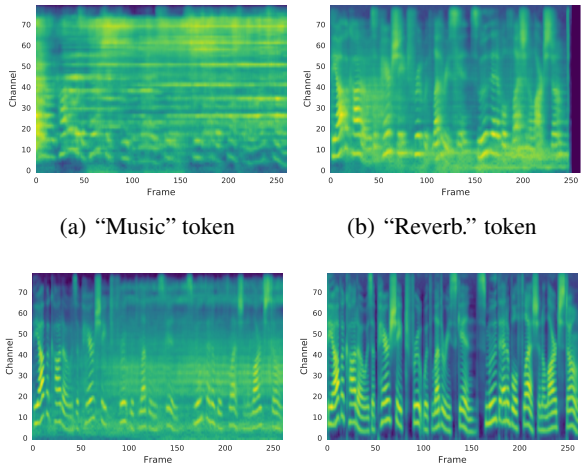
Audio examples of non-parallel style transfer can be found here.

7. Experiments: Unlabeled Noisy Found Data

Studio-quality data can be both economically and time consuming to record. While the internet holds vast amounts of rich real-life expressive speech, it is often noisy and difficult to label. In this section, we demonstrate how GSTs can be used to train robust models directly from noisy found data, without modifications.

7.1. Artificial Noisy Data

As a first experiment, we artificially generate training sets by adding noise to clean speech. The motivation here is to simulate real noisy data while performing controlled experiments. To achieve this, we pass the single-speaker US English proprietary dataset from (Wang et al., 2017a) into a room simulator (Kim et al., 2017), which adds varying types of background noise and room reverberations. The signal-to-noise ratio (SNR) ranges from 5-25 dB, and the T60s of room reverberation ranges from 100-900 ms. We create four different training sets where 50%, 75%, 90% and 95% of the input is noisified, respectively.



(a) "Music" token (b) "Reverb." token

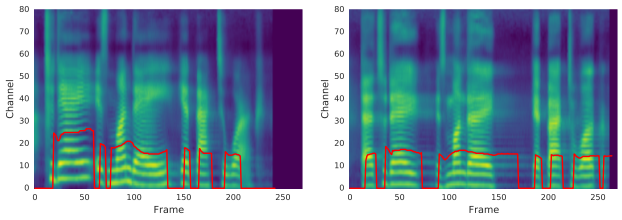
(c) "Noise" token (d) Clean token

Figure 6. Noisy and clean tokens uncovered.

After training a GST-augmented Tacotron on these datasets, we run inference in the first mode described in Section 2.2. Instead of providing a reference signal, we condition the model on each individual style token, which gives us an interpretable, audible sense of what each token has learned. Interestingly, we find that different noises are treated as styles and "absorbed" into different tokens. We illustrate the spectrograms from a few tokens in Figure 6. We can see (and hear) that these tokens clearly correspond to different interference types, such as music, reverberation and general background noise. Importantly, this method reveals that a subset of the learned tokens also correspond to completely clean speech. This means that we can synthesize clean speech for arbitrary text input by conditioning the model on a single, clean style token.

To demonstrate this, we run inference using a manually-identified clean style token (scaled to 0.3), and then evaluate the output using MOS naturalness tests. We use the same 100-phrase evaluation set as (Wang et al., 2017a), collecting 8 ratings each from crowdsourced native speakers. Table 2 shows MOS results for both a baseline Tacotron and a "clean-token" GST model. While the baseline Tacotron achieves a 4.0 MOS when the dataset is 100% clean, MOS decreases as interference increases, dropping to a low score of 1.353. Because the model has no prior knowledge of speech or noise, it blindly models all statistics in the training set, resulting in substantial amounts of noise during synthesis.

By contrast, the GST model achieves about 4.0 MOS in



(a) Token A (b) Token B

Figure 7. Log-mel spectrograms (overlaid with F0 tracks) of two randomly chosen tokens from a GST model trained on the TED data. The two tokens uncover two different speakers.

all noise conditions. Note that the number of tokens needs to increase along with the percentage of noise to achieve this result. For example, a 10-token GST model yields clean tokens when trained on a 50% noise dataset, but the noisier datasets required a 20-token model. Future work may explore how to adapt the number of tokens automatically to a given data distribution.

Audio examples from these models can be found here.

7.2. Real Multi-Speaker Found Data

Our second experiment uses real data. This dataset is made up of audio tracks mined from 439 official TED YouTube channel videos. The tracks contain significant acoustic variations, including channel variation (near- and far-field speech), noise (e.g. laughs), and reverberation. We use an endpointer to segment the audio tracks into short clips, followed by an ASR model to create <text, audio> training pairs. Despite the fact that the ASR model generates a significant number of transcription and misalignment errors, we perform no other preprocessing. The final training set is about 68 hours long and contains about 439 speakers.

Without using any metadata as labels, we train a baseline Tacotron and a 1024-token GST model for comparison. As expected, the baseline fails to learn, since the multi-speaker data is too varied. The GST model results are presented in Figure 7. This shows spectrograms for the same phrase overlaid with F0 tracks, generated by conditioning the model on two randomly chosen tokens. Examining the trained GSTs, we find that different tokens correspond to different speakers. This means that, to synthesize with a specific speaker's voice, we can simply feed audio from that speaker as a reference signal. See Section 7.3 for more quantitative evaluations.

Finally, we exploit the fact that most of the talks are in English, but a small fraction are in Spanish. For this experiment, we compare baseline and GST-enabled noisy data models on a cross-lingual style transfer task. For a baseline, we train a multi-speaker Tacotron similar to (Ping et al., 2017), using video IDs as a proxy for speaker labels. Conditioned on a Spanish speaker label, we then synthesize 100 English phrases. For the GST system, we feed a reference signal from the same Spanish speaker and synthesize the same 100 English phrases. While the Spanish accent from the speaker is not preserved, we find that the GST model produces completely intelligible English speech with a similar pitch range as the speaker. By contrast, the multi-speaker Tacotron output is much less intelligible.

Table 3. WER for the Spanish to English unsupervised language transfer experiment. Note that WER is an underestimate of the true intelligibility score; we only care about the relative differences.

MODEL	WER (INS/DEL/SUB)
GST	18.68 (6.13/2.37/10.18)
MULTI-SPEAKER	56.18 (3.75/20.27/32.14)

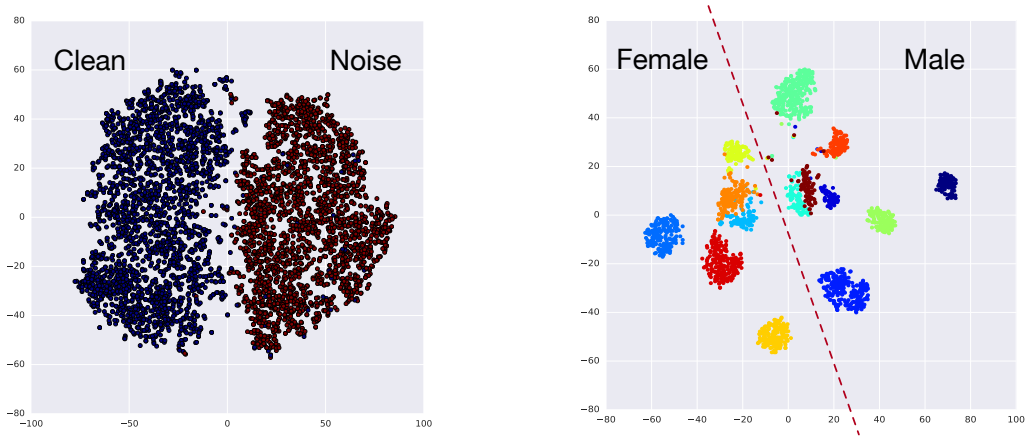
To evaluate this result objectively, we compute word error rates (WER) of an English ASR model on the synthesized speech. As shown in Table 3, the WER of the GST utterances is much lower than that of the multi-speaker model.

The results strongly corroborate that GSTs learn embeddings disentangled from text content. Though this is an exciting early result, an in-depth study of using GST for prosody-preserving language transfer is in order.

7.3. Quantitative Evaluations

We use t-SNE (Maaten & Hinton, 2008) to visualize the style embeddings learned from both the artificial noise and TED datasets. Figure 8(a) shows that the embeddings learned from the artificial noisy dataset (50% clean) are clearly separated into two classes. Figure 8(b) shows style embeddings for 2,000 randomly drawn samples containing 14 TED talk data speakers. We see that samples are well separated into 14 clusters, each corresponding to an individual speaker. Female and male speakers are linearly separable.

We also use style embeddings as features to perform noise and speaker classification with Linear Discriminative Analysis. Results are shown in Table 4. For noise classification, GSTs uncover the true label with 99.2% accuracy. For speaker classification, we use TED video IDs as true labels and compare with the i-vector method (Dehak et al., 2011), a standard representation used in modern speaker verification systems. For this task, the test set contains 431 speakers. While both trained and tested on short utterances (mean duration 3.75 secs), we can see that GSTs are comparable with i-vectors. This is an encouraging result, given that i-vectors were specifically designed for speaker classification. We speculate that GST has the potential to be applied to speaker diarization.



(a) 50% noisy data (b) Multi-speaker TED

Figure 8. Style embedding visualization using t-SNE.

Table 4. Classification accuracy (noise-vs-clean and TED speaker ID) using GSTs and i-vectors. Despite being trained within a generative model, GSTs encode rich discriminative information.

EMBEDDING	ARTIFICIAL DATA	TED (431 SPEAKERS)
GST	99.2%	75.0%
I-VECTOR	/	73.4%

7.4. Implications

The results above have important implications for future TTS research on found data. First, due to the robustness of GSTs to both acoustic and textual noise, the design of automated data mining pipelines may be greatly simplified. Accurate segmentation and ASR models, for example, are no longer necessary to build high-quality TTS models. Second, style attributes, such as emotion, are often very difficult to label for large-scale noisy data. Using GSTs or weights to automatically generate style annotations may substantially reduce the human-in-the-loop efforts.

8. Conclusions and Discussions

This work has introduced Global Style Tokens, a powerful method for modeling style in end-to-end TTS systems. GSTs are intuitive, easy to implement, and learn without explicit labels. We have shown that, when trained on expressive speech data, a GST model yields interpretable embeddings that can be used to control and transfer style. We have also demonstrated that, while originally conceived to model speaking styles, GSTs are a general technique for uncovering latent variations in data. This was corroborated by experiments on unlabeled noisy found data, which showed that the GST model learns to decompose various noise and speaker factors into separate style tokens.

There is still much to be investigated, including improving the learning of GST, and using GST weights as targets to predict from text. Finally, while we only applied GST to Tacotron in this work, we believe it can be readily used by other types of end-to-end TTS models. More generally, we envision that GST can be applied to other problem domains that benefit from interpretability, controllability and robustness. For example, GST may be similarly employed in text-to-image and neural machine translation models.

Acknowledgements

The authors thank Aren Jansen, Rob Clark, Zhifeng Chen, Ron J. Weiss, Mike Schuster, Yonghui Wu, Patrick Nguyen, and the Machine Hearing, Google Brain and Google TTS teams for their helpful discussions and feedback.

References

Arik, Sercan O, Chrzanowski, Mike, Coates, Adam, Diamos, Gregory, Gibiansky, Andrew, Kang, Yongguo, Li, Xian, Miller, John, Raiman, Jonathan, Sengupta, Shubho, et al. Deep voice: Real-time neural text-to-speech. *ICML*, 2017.

Dehak, Najim, Kenny, Patrick J, Dehak, Réda, Dumouchel, Pierre, and Ouellet, Pierre. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

Eyben, Florian, Buchholz, Sabine, and Braunschweiler, Norbert. Unsupervised clustering of emotion and voice styles for expressive tts. In *ICASSP*, pp. 4009–4012. IEEE, 2012.

Graves, Alex, Wayne, Greg, and Danihelka, Ivo. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Griffin, Daniel and Lim, Jae. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

Hsu, Wei-Ning, Zhang, Yu, and Glass, James. Unsupervised learning of disentangled and interpretable representations from sequential data. In *Advances in Neural Information Processing Systems*, 2017.

Jauk, Igor. *Unsupervised learning for expressive speech synthesis*. PhD thesis, Universitat Politècnica de Catalunya, 2017.

Kim, Chanwoo, Misra, Ananya, Chin, Kean, Hughes, Thad, Narayanan, Arun, Sainath, Tara, and Bacchiani, Michiel. Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home. *Proc. INTERSPEECH. ISCA*, 2017.

Kinnunen, Tomi, Juvela, Lauri, Alku, Paavo, and Yamagishi, Junichi. Non-parallel voice conversion using i-vector plda: Towards unifying speaker verification and transformation. In *ICASSP*, 2017.

Krueger, David, Maharaj, Tegan, Kramár, János, Pezeshki, Mohammad, Ballas, Nicolas, Ke, Nan Rosemary, Goyal, Anirudh, Bengio, Yoshua, Larochelle, Hugo, Courville, Aaron, et al. Zoneout: Regularizing RNNs by randomly preserving hidden activations. In *Proc. ICLR*, 2017.

Luong, Hieu-Thi, Takaki, Shinji, Henter, Gustav Eje, and Yamagishi, Junichi. Adapting and controlling dnn-based speech synthesis using input codes. In *ICASSP*, pp. 4905–4909. IEEE, 2017.

Maaten, Laurens van der and Hinton, Geoffrey. Visualizing data using t-sne. *Journal of machine learning research*, 9 (Nov):2579–2605, 2008.

Nakashika, Toru, Takiguchi, Tetsuya, Minami, Yasuhiro, Nakashika, Toru, Takiguchi, Tetsuya, and Minami, Yasuhiro. Non-parallel training in voice conversion using an adaptive restricted boltzmann machine. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 24(11):2032–2045, November 2016.

Ping, Wei, Peng, Kainan, Gibiansky, Andrew, Arik, Sercan O, Kannan, Ajay, Narang, Sharan, Raiman, Jonathan, and Miller, John. Deep voice 3: 2000-speaker neural text-to-speech. *arXiv preprint arXiv:1710.07654*, 2017.

Rosenberg, Andrew. AuToBI-a tool for automatic ToBI annotation. In *Interspeech*, pp. 146–149, 2010. URL http://eniac.cs.qc.cuny.edu/andrew/autobi/.

Shen, Jonathan, Pang, Ruoming, Weiss, Ron J, Schuster, Mike, Jaitly, Navdeep, Yang, Zongheng, Chen, Zhifeng, Zhang, Yu, Wang, Yuxuan, Skerry-Ryan, RJ, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *arXiv preprint arXiv:1712.05884*, 2017.

Silverman, Kim, Beckman, Mary, Pitrelli, John, Ostendorf, Mori, Wightman, Colin, Price, Patti, Pierrehumbert, Janet, and Hirschberg, Julia. ToBI: A standard for labeling english prosody. In *Second International Conference on Spoken Language Processing*, 1992.

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.

Taigman, Yaniv, Wolf, Lior, Polyak, Adam, and Nachmani, Eliya. Voice synthesis for in-the-wild speakers via a phonological loop. *arXiv preprint arXiv:1707.06588*, 2017.

Taylor, Paul. *Text-to-speech synthesis*. Cambridge university press, 2009.

Theis, Lucas, Oord, Aäron van den, and Bethge, Matthias. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

van den Oord, Aäron, Dieleman, Sander, Zen, Heiga, Simonyan, Karen, Vinyals, Oriol, Graves, Alex, Kalchbrenner, Nal, Senior, Andrew, and Kavukcuoglu, Koray. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*, 2016.

van den Oord, Aaron, Vinyals, Oriol, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pp. 6309–6318, 2017.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, and Polosukhin, Illia. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.

Wang, Yuxuan, Skerry-Ryan, RJ, Stanton, Daisy, Wu, Yonghui, Weiss, Ron J., Jaitly, Navdeep, Yang, Zongheng, Xiao, Ying, Chen, Zhifeng, Bengio, Samy, Le, Quoc, Agiomyrgiannakis, Yannis, Clark, Rob, and Saurous, Rif A. Tacotron: Towards end-to-end speech synthesis. In *Proc. Interspeech*, pp. 4006–4010, August 2017a. URL https://arxiv.org/abs/1703.10135.

Wang, Yuxuan, Skerry-Ryan, RJ, Xiao, Ying, Stanton, Daisy, Shor, Joel, Battenberg, Eric, Clark, Rob, and

Saurous, Rif A. Uncovering latent style factors for expressive speech synthesis. *ML4Audio Workshop, NIPS*, 2017b.

Wightman, Colin W. Tobi or not tobi? In *Speech Prosody 2002, International Conference*, 2002.

Wu, Zhizheng, Chng, Eng Siong, and Li, Haizhou. Conditional restricted boltzmann machine for voice conversion. In *ChinaSIP*, 2013.

Zen, Heiga, Agiomyrgiannakis, Yannis, Egberts, Niels, Henderson, Fergus, and Szczepaniak, Przemysław. Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices. *Proceedings Interspeech*, 2016.