Paper #2729

Wave-Tacotron: Spectrogram-free end-to-end text-to-speech synthesis

Google Research

Ron J. Weiss, RJ Skerry-Ryan, Eric Battenberg, Soroosh Mariooryad, and Diederik P. Kingma

{ronw, rjryan, ebattenberg, soroosh, durk}@google.com

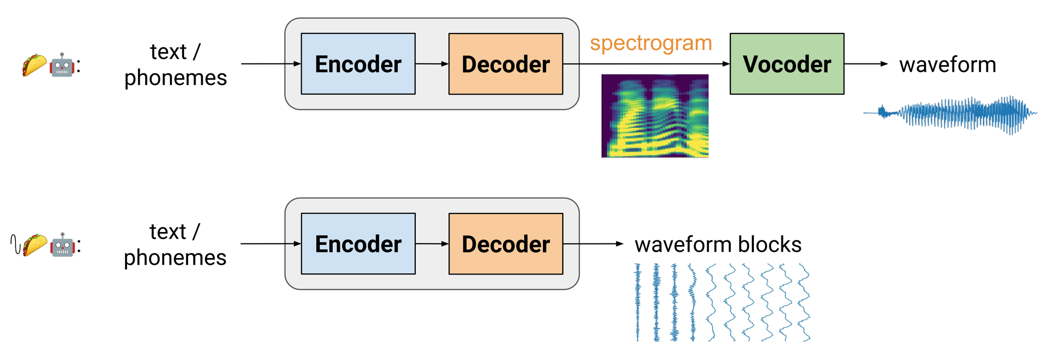
Overview

Summary

- TTS* in **one sequence-to-sequence model**
  - block-autoregressive normalizing flow, no vocoder
  - *normalized-text- or phoneme-to-speech
- Directly predict 40ms *waveform blocks* at each decoder step
  - no overlap, no spectrograms
- End-to-end training, maximizing likelihood
- High fidelity output
  - trails Tacotron+WaveRNN baseline
  - higher sample variation, captures modes of training data?
- ~10x faster than real-time synthesis on TPU

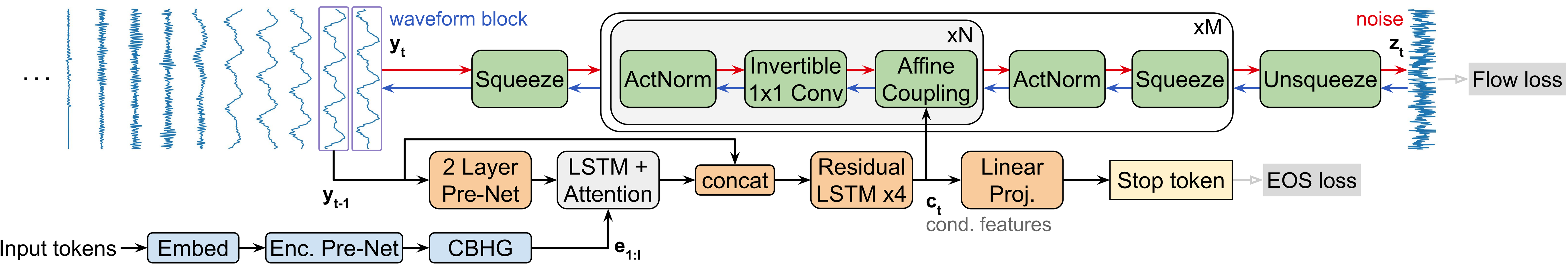
Background

- Tacotron [1] [2]: phoneme input, mel spectrogram frame output
  - autoregressive decoder, each step generates new frame
  - separate vocoder, inverts spectrogram to waveform
    - e.g., WaveRNN [3], sample-by-sample autoregressive



- Wave-Tacotron: generate sequence of non-overlapping waveform blocks
  - K = 960 samples (40 ms at 24 kHz)

Model



Architecture

- Replace decoder post-net and vocoder with conditional normalizing flow

  $P(y_t \mid c_t) = P(y_t \mid y_{1:t-1}, e_{1:I})$

  $= P(y_t \mid$ previous waveform blocks, text$)$

- Tacotron encoder/decoder predicts flow conditioning input
- Train end-to-end, maximize likelihood of training data
- Block-autoregressive generation
  - waveform samples in each block generated in parallel

Normalizing flow

- Model **joint distribution** of K samples: $P(y_{t1}, y_{t2}, \ldots, y_{tK} \mid c_t)$
  - similar to FloWaveNet [4], WaveGlow [5] neural vocoders
- Invertible network
  - training: transform waveform block into noise
  - sampling: transform noise sample into waveform block using inverse
- Change of variables $\mathbf{y}_t = g(\mathbf{z}_t; c_t)$
- Maximize likelihood $P(\mathbf{y}_t \mid c_t) = P(\mathbf{z}_t \mid c_t)\ |\det(d\mathbf{z}_t / d\mathbf{y}_t)|$
- Coupling layers [6] ➔ fast inverse, Jacobian determinant

Training

- Teacher forced conditioning
- At each step: transform waveform block $\mathbf{y}_t$ into noise $\mathbf{z}_t$
- Flow loss

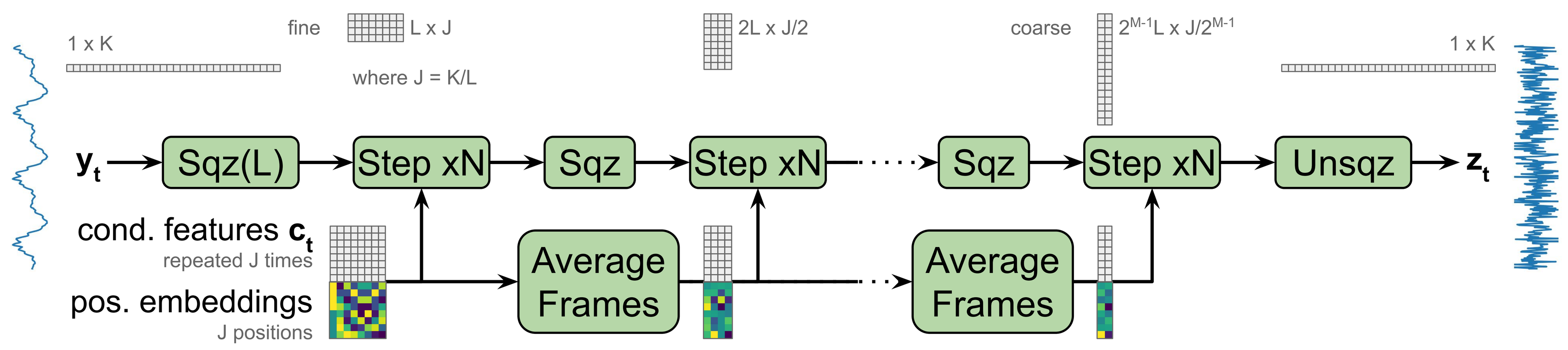
  $-\log P(\mathbf{y}) = \mathrm{sum}_t -\log P(\mathbf{y}_t \mid c_t)$

  $= \mathrm{sum}_t -\log \underbrace{N(g^{-1}(\mathbf{y}_t; c_t); \mathbf{0}, \mathbf{I})}_{\text{spherical Gaussian}} - \log \underbrace{|\det(dg^{-1}(\mathbf{y}_t; c_t) / d\mathbf{y}_t)|}_{\text{Jacobian determinant}}$

- EOS *stop token* classifier loss: P(t is last frame)

Multiscale network [7]

- Squeeze waveform block into frames, length L = 10 samples
- M = 5 stages, each processes signal at different scale
  - N = 12 steps per stage
  - deep convnet: M N = 60 total steps
- Sinusoidal position embeddings encode position in each frame



Sampling

- Invert the flow network
  - take inverse of each layer, reverse order
- At each step
  - sample noise vector $\mathbf{z}_t \sim N(\mathbf{0}, T\,\mathbf{I})$
  - generate waveform block with flow $\mathbf{y}_t = g(\mathbf{z}_t; c_t)$
  - autoregressive conditioning on previous output $\mathbf{y}_{t-1}$
- concatenate blocks $\mathbf{y}_t$ to form final signal $\mathbf{y} = \mathrm{vstack}(\mathbf{y}_t)$

Experiments

Data

- US English, single female speaker, sampled at 24 kHz
  - 39 hours training, 601 utterances held out
- Baselines
  - Tacotron-PN (postnet) + Griffin-Lim (similar to [1])
  - Tacotron + WaveRNN (similar to [2])
  - Tacotron + Flow vocoder
    - fully parallel (similar flow to Wave-Tacotron, 6 stages)
- Subjective listening tests rating speech naturalness
  - MOS on 5 point scale

Generation speed

- Seconds to generate 5 seconds of speech
  - 90 input tokens, batch size 1
- Wave-Tacotron ~10x faster than real-time on TPU (2x on CPU)
  - slower as frame size K decreases (more autoregressive steps)
- ~10x faster than Tacotron + WaveRNN on TPU (25x on CPU)
- ~2.5x slower than fully parallel vocoder on CPU

Model	K	Vocoder	TPU	CPU
Tacotron-PN		Griffin-Lim, 100 iterations	0.14	0.88
Tacotron-PN		Griffin-Lim, 1000 iterations	1.11	7.71
Tacotron		WaveRNN	5.34	63.38
Tacotron		Flowcoder	0.49	0.97
Wave-Tacotron	13.3ms	–	0.80	5.26
Wave-Tacotron	26.6ms	–	0.64	3.25
Wave-Tacotron	40.0ms	–	0.58	2.52
Wave-Tacotron	53.3ms	–	0.55	2.26

Results

- Tacotron + WaveRNN best
  - char / phoneme roughly on par
- Wave-Tacotron trails by ~0.2 points
  - phoneme > char
  - network uses capacity to model detailed waveform structure instead of pronunciation?
- Large gap to Tacotron-PN and Tacotron + Flowcoder

Model	Vocoder	Input	MOS
Ground truth	–	–	4.56 ± 0.04
Tacotron-PN	Griffin-Lim	char	3.68 ± 0.08
Tacotron-PN	Griffin-Lim	phoneme	3.74 ± 0.07
Tacotron	WaveRNN	char	4.36 ± 0.05
Tacotron	WaveRNN	phoneme	$\mathbf{4.39 \pm 0.05}$
Tacotron	Flowcoder	char	3.34 ± 0.07
Tacotron	Flowcoder	phoneme	3.31 ± 0.07
Wave-Tacotron	–	char	4.07 ± 0.06
Wave-Tacotron	–	phoneme	4.23 ± 0.06

Ablations

- 2 layer decoder LSTM
  256 channels in coupling layers
- Optimal sampling temperature T = 0.7
- Deep multiscale flow is critical
- Varying block size K
  - quality starts degrading for K > 40 ms

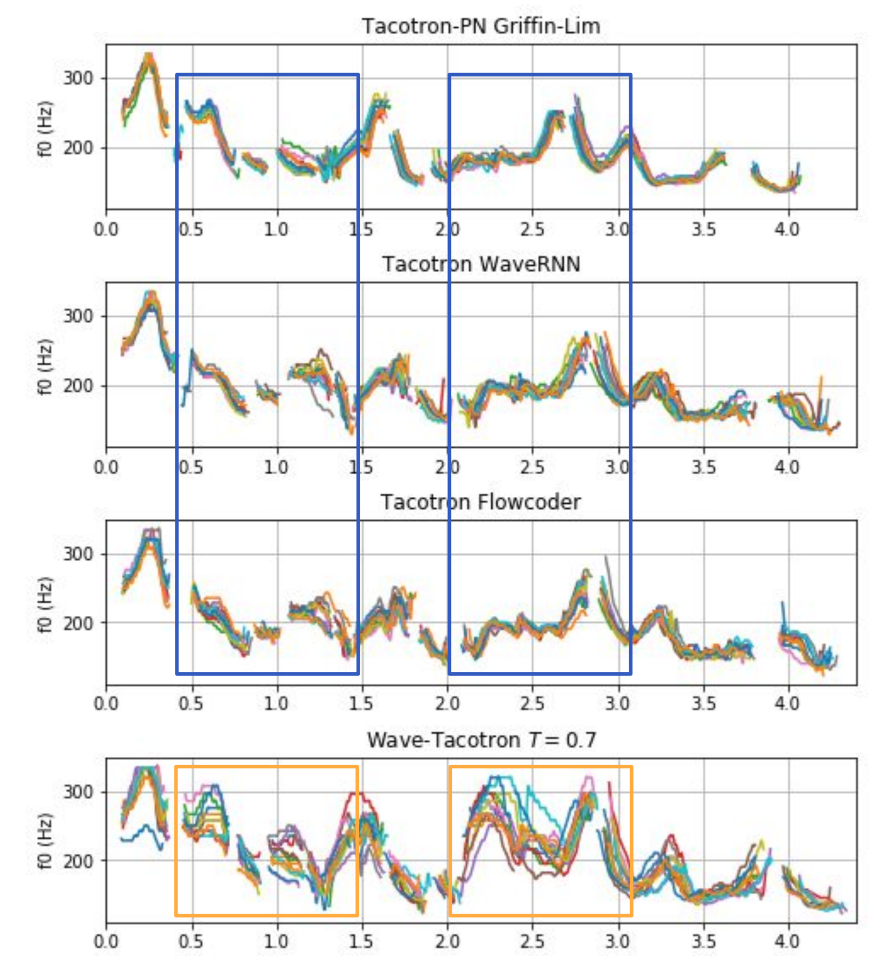
Model	R	M	N	MOS
Base $T = 0.8$	3	5	12	4.01 ± 0.06
$T = 0.6$	3	5	12	4.12 ± 0.06
$T = 0.7$	3	5	12	4.16 ± 0.06
$T = 0.9$	3	5	12	3.77 ± 0.07
128 flow channels	3	5	12	3.31 ± 0.07
30 steps, 5 stages	3	5	6	3.11 ± 0.07
60 steps, 4 stages	3	4	15	3.50 ± 0.07
60 steps, 3 stages	3	3	20	2.44 ± 0.07
$K = 320$ (13.33 ms)	1	5	12	4.05 ± 0.06
$K = 640$ (26.67 ms)	2	5	12	4.06 ± 0.06
$K = 1280$ (53.3 ms)	4	5	12	3.55 ± 0.07

Sound examples:

https://google.github.io/tacotron/publications/wave-tacotron

Sample variation

- Generate 12 samples from the same input text
- Baselines generate very consistent samples, across vocoders
  - same prosody every time
- Wave-Tacotron has high variance
  - captures multimodal training distribution?
    - Tacotron regression loss collapses to single *prosody mode*?
  - similar pattern in Flowtron [8]
  - useful for ASR data augmentation?



References

[1] Wang, et al., Tacotron: Towards End-to-End Speech Synthesis. Interspeech 2017.
[2] Shen, et al., Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. ICASSP 2018.
[3] Kalchbrenner, et al., Efficient Neural Audio Synthesis. ICML 2018.
[4] Kim, et al., FloWaveNet : A Generative Flow for Raw Audio. ICML 2019.
[5] Prenger, et al., WaveGlow: A Flow-based Generative Network for Speech Synthesis. ICASSP 2019.
[6] Dinh, et al., NICE: Non-linear independent components estimation. ICLR 2015.
[7] Dinh and Bengio, Density estimation using Real NVP. ICLR 2017.
[8] Valle, et al., Flowtron: an Autoregressive Flow-based Generative Network for Text-to-Speech Synthesis . ICLR 2021.