Improving the expressiveness of neural vocoding with non-affine Normalizing Flows

Adam Gabryś, Yunlong Jiao, Viacheslav Klimkov, Daniel Korzekwa, Roberto Barra-Chicote

Alexa AI

{gabrysa, jyunlong, vklimkov, korzekwa, rchicote}@amazon.com

Abstract

This paper proposes a general enhancement to the Normalizing Flows (NF) used in neural vocoding. As a case study, we improve expressive speech vocoding with a revamped Parallel Wavenet (PW). Specifically, we propose to extend the affine transformation of PW to the more expressive invertible non-affine function. The greater expressiveness of the improved PW leads to better-perceived signal quality and naturalness in the waveform reconstruction and text-to-speech (TTS) tasks. We evaluate the model across different speaking styles on a multi-speaker, multi-lingual dataset. In the waveform reconstruction task, the proposed model closes the naturalness and signal quality gap from the original PW to recordings by 10%, and from other state-of-the-art neural vocoding systems by more than 60%. We also demonstrate improvements in objective metrics on the evaluation test set with L2 Spectral Distance and Cross-Entropy reduced by 3% and 6‰ comparing to the affine PW. Furthermore, we extend the probability density distillation procedure proposed by the original PW paper, so that it works with any non-affine invertible and differentiable function.[1]

Index Terms: Text To Speech, Neural vocoder, Normalizing Flows

1. Introduction

Text-to-speech (TTS) is a rapidly growing domain in artificial intelligence. TTS systems attract more attention every year as people use them for Voice Assistants, education, gaming, and much more. High quality and low latency systems are necessary to satisfy TTS customer needs. Most state-of-the-art Neural TTS systems address the problem of speech generation in two steps. The first step focuses on generating a low resolution intermediate speech representation [1, 2, 3]. The second step concentrates on transforming this acoustic representation into a high-fidelity high-quality acoustic waveform. Commonly, the model that refers to the second step is called the vocoder.

The state-of-the-art vocoders are generative Deep Neural Networks (DNN). The two most prominent classes of generative models are sequential and parallel architectures. Typically, sequential models achieve state-of-the-art results in audio, computer-vision, and textual domains [4]. In neural vocoding one of the best quality sequential architectures is WaveNet [5]. It generates waveforms autoregressively using a stack of dilated causal convolutions. However, due to a high number of computations per sample, and the high temporal resolution of the speech signal [6] it is not suited to real-time applications. More efficient sequential models were later proposed [7, 8] that are an order of magnitude faster than WaveNet. Nevertheless, these models are sequential, so computations cannot be easily parallelized to fully utilize modern Deep Learning ASICs, or GPUs.

The above limitation has driven most of the recent research in neural vocoding towards parallel models. The two widely used neural vocoding parallel architectures involve Generative Adversarial Networks (GAN) [9, 10, 11, 12] and Normalizing Flows (NF) [13, 14, 15, 16, 17, 18]. The generator part of a GAN can typically be any function appropriate for transforming some random inputs into synthetic outputs. GAN-based vocoders enjoy great flexibility of architectural design and hence fast parallel generation. The state-of-the-art adversarial-based vocoders produce high-quality, natural-sounding speech but suffer from occasional audio glitches. These artifacts can significantly reduce the subjective score of such models [19]. This is a common problem related to the performance of GANs in generalizing to unseen data [20, 21]. NF provides a general framework for defining probability distributions over continuous random variables. NF takes a base distribution and transforms it to the target probability density with sequential invertible and differentiable transformations. Normalizing Flows are compelling, in the context of vocoding, due to their efficient parallel synthesis procedure [13], and great generalization [19].

In this work, we focus on neural speech vocoding with Normalizing Flows (NF). The majority of NF used in neural vocoding implement a sequence of transformations as affine functions [13, 14, 15, 16]. This type of transformation is known to be limited in its density modeling power [22, 23, 24, 25]. In practice, we found that it adversely impacts perceived naturalness and signal quality, especially in vocoding scenarios involving highly expressive speech

The contributions of this work are: 1) We change the inexpressive affine flow transformation of PW [13] to a more expressive, non-affine function; 2) We extend the probability density distillation [26] procedure proposed by the original PW [13], so that it works with any non-affine invertible and differentiable function; and 3) We perform a detailed evaluation of the model across different speaking styles on a multi-speaker, multi-lingual dataset. We demonstrate that our network is qualitatively and quantitively preferred over the original PW in the waveform reconstruction and TTS tasks;

2. Normalizing Flows & Related Work

NF transforms some D-dimensional real vector of continuous random variables $\mathbf{u}$ into another D-dimensional real vector of continuous random variables $\mathbf{x}$. Usually $\mathbf{u}$ is sampled from a simple base distribution (for example Logistic) $p_u(\mathbf{u})$. In the vocoding task $\mathbf{x}$ corresponds to audio signal that follows a probability density $p_x(\mathbf{x})$. Conceptually, we can outline

[1] Audio samples will be made available on the amazon.science blog.

We would like to thank Alexis Moinet, Vatsal Aggarwal and Bartosz Putrycz for insightful research discussions. And David McHardy with Jaime Lorenzo Trueba for constructive criticism of the manuscript.

two blocks in the NF. One is the transformation function T, which has to be invertible and differentiable. The other is the conditioner neural network c that predicts the parametrization $\mathbf{h}$ for the transformation T.

$$\mathbf{x} = T(\mathbf{u}; \mathbf{h}) \qquad \mathbf{u} = T^{-1}(\mathbf{x}; \mathbf{h}) \qquad \mathbf{h} = c(\mathbf{u}) \tag{1}$$

Given the invertible and differentiable nature of T, the density of $\mathbf{x}$ is well-defined and can be obtainable by a change of variables:

$$p_x(\mathbf{x}) = p_u(\mathbf{u})|detJ_T(\mathbf{u})|^{-1} \tag{2}$$

The Jacobian $J_T(\mathbf{u})$ is D x D matrix of all partial derivatives of T over $\mathbf{u}$. In this section, we discuss the merits and limitations of different NF architectures and outline our model design.

There are two major paradigms of training NF. One paradigm is to fit NF to the data with Maximum Likelihood Estimation (MLE) [14, 16, 17, 18]. In practice, it means that the model computes T^{-1} during training and T during the synthesis. Another paradigm assumes that we can evaluate the target data density, and we aim to train a NF to minimize the divergence loss. Commonly this is done with knowledge distillation [26], where the data density is estimated through a teacher network [13, 15]. A notable example of this training in the context of vocoding is the use of a high-quality Wavenet [5] to train a NF-based PW [13]. This paradigm for training and synthesis requires only the forward transformation T. In both paradigms, to train the model, we have to compute the Jacobian determinant, which typically costs $O(D^3)$. However, in many practical applications, we can reduce this complexity. An autoregressive conditioner network has the Jacobian that is a lower triangular matrix with determinant computable in $O(D)$ [13, 14, 15, 16, 27, 28]. It is shown [24, 29] that under the assumption of enough capacity and data, an autoregressive conditioner with non-linear transformations can approximate any continuous distribution with any desired precision - a property called universal approximation. NF using such a conditioner can parallelize the forward transformation computation, but its inverse is sequential. This poses a challenge for the MLE paradigm training due to the high temporal resolution of speech data. Coupling layers are a common workaround for this problem [14, 28, 30]. Such an architecture allows efficient computation of both forward and inverse transformations. However, it may limit the expressivity of NF since a significant number of dimensions are left unchanged at each flow layer [31]. Because of the above argumentation, in this work, we decided to use ParallelWavenet [13] which is a fully-autoregressive model trained with knowledge distillation that does not require any computation of the transformation inverse.

The NF transformation has to be invertible and differentiable. The most straightforward and common design choice is to implement the transformation as an affine function [13, 14, 15, 16, 27, 28]. Such a design is attractive because of its simplicity and analytically tractability. However, the drawback of such a transformation is its limited expressivity. Specifically, the output of NF belongs to the same distribution family as its base. In some cases, this might negatively affect the capture of multimodal target distributions [23, 24, 25]. To overcome this limitation, the transformation might be implement as a composition or the weighted sum of monotonically increasing activation functions [23, 24, 25], the integral of some positive function [29], or a spline of analytically invertible monotonic functions [22, 30, 32]. All above transformations are Universal Approximators [24, 29]. Another idea is to use constrained residual functions [33, 34]. Unfortunately, for these methods, we either cannot efficiently compute the determinant of the Jacobian or the function has limited expressivity [31]. Finally, we might also construct the flow by defining an ordinary differential equation (ODE) that describes the evolution of NF in time instead of a finite sequence of transformations [17, 18]. According to recent surveys [35], Normalizing Flows with finite composition of non-affine transformations outperform other flow-based methods. Considering the above pros and cons, we decide to enhance PW with a composition of monotonically increasing non-affine activation functions inspired by Flow++ [25].

3. Model description

3.1. Parallel Wavenet

The original Parallel Wavenet [13] uses conditional Inverse Autoregressive Flows [27] to shift and scale base a logistic distribution to model the probability density of audio waveforms. The procedure is as follows. First, to generate condtioning features $\mathbf{m}$, we pass the sequence of Mel-spectrograms[2] through transposed convolutions that upsample it to match the audio wavefrom of length D. Then, we sample a D-long sequence of noise from Logistic distribution $\mathbf{u} \sim \mathcal{L}(0, 1)$. We aim to model the audio waveform $\mathbf{x}$ with affine transformations that shift and scale the input noise $\mathbf{u}$. To predict the transformation scales $\boldsymbol{\alpha}$ and shifts $\boldsymbol{\beta}$ we use residual gated causal convolutions (RGCNN) [13, 36]. RGCNN takes as an input conditioning $\mathbf{m}$ and a sequence of the noise $\mathbf{u}$. For the t-th time step the predicted audio sample x_t is:

$$\begin{aligned} x_t &= \alpha_t \cdot u_t + \beta_t \\ \alpha_t, \beta_t &= \text{RGCNN}(\mathbf{u}_{<\mathbf{t}}, \mathbf{m}) \end{aligned} \tag{3}$$

Multiple instances of such Flows are stacked on top of each other to increase the expressivity of NF.

Parallel Wavenet [13] is trained with probability density distillation. It is defined as KLD loss D_{KL} between the teacher P_T given student predictions, and student P_S distributions. In general KLD can be defined as the difference between Cross Entropy $H(P_S, P_T)$ and Entropy $H(P_S)$ terms:

$$D_{KL}(P_S||P_T) = H(P_S, P_T) - H(P_S) \tag{4}$$

In the original Parallel Wavenet student distribution follows a Logistic function, and we can compute student Entropy analytically.

$$H(P_S) = \mathbb{E}_{u\sim\mathcal{L}(0,1)}\left[\sum_{t=1}^{D} \ln(\beta_t)\right] + 2D \tag{5}$$

The Cross Entropy term is computed via Monte Carlo approximation. For every sample x we draw from the student p_S, we compute all $p_T(x_t|x_{<t})$ with the teacher, and then evaluate $H(p_S(x_t|x_{<t}), p_T(x_t|x_{<t}))$.

$$H(P_S, P_T) = \sum_{t=1}^{D} \mathbb{E}_{p_S(\mathbf{x}_{<\mathbf{t}})} H(p_S(x_t|\mathbf{x}_{<\mathbf{t}}), p_T(x_t|\mathbf{x}_{<\mathbf{t}})) \tag{6}$$

Sampling from the student does not require passing noise through the NF. In a single forward pass, we cache parametrization for the Logistic Distribution, and in Monte Carlo sampling, we apply the reparametrization trick [37].

[2]Original PW uses linguistic features instead of acoustic signal representation such as Mel-spectrogram.

3.2. Non-affine transformation

In the original PW, a student can only output a uni-modal Logistic distribution per time step, and therefore is not able to reconstruct a multi-modal mixture of Logistics (MoL) of a Wavenet teacher [5]. To overcome this limitation, we propose to extend the affine transformation of the original PW to a non-affine function. Inspired by the Flow++ [25], we implement transformation T as a cumulative distribution function (CDF) for a mixture of N logistics (MoL) followed by an inverse sigmoid (logit) σ^{-1} and an affine transformation. Such transformation is invertible and differentiable. The MoL CDF domain is (0, 1), so a logit of it always exists. Also, both MoL CDF and logit functions are monotonically increasing, though invertible. Logistics are parameterized by shifts $\boldsymbol{\mu}$, and scales $\boldsymbol{s}$ that are combined with mixing proportions $\boldsymbol{\pi}$. The output of the logit is scaled by α and shifted by β. For the t-th time step, the predicted audio sample x_t is:

$$\begin{aligned} x_t &= \sigma^{-1}(\text{MoLCDF}(u_t; \boldsymbol{\pi_t}, \boldsymbol{\mu_t}, \mathbf{s_t})) \cdot \alpha_t + \beta_t \\ \alpha_t, \beta_t, \boldsymbol{\pi_t}, \boldsymbol{\mu_t}, \boldsymbol{s_t} &= \text{RGCNN}(\mathbf{u_{<t}}, \mathbf{m}) \end{aligned} \tag{7}$$

Comparing to the affine function (equation 3), such a transformation is non-affine and can induce multimodality [24]. The computation of the Jacobian of the transformation is straightforward since the derivative of MoLCDF is the MoL probability density function (PDF). We also know the derivatives of logit and affine functions:

$$\begin{aligned} \frac{\partial x_t}{\partial u_t} = \exp(&\alpha_t + \text{MoLPDF}(u_t, \boldsymbol{\pi_t}, \boldsymbol{\mu_t}, \mathbf{s_t}) \\ &- \ln(\text{MoLCDF}(u_t, \boldsymbol{\pi_t}, \boldsymbol{\mu_t}, \mathbf{s_t})) \\ &- \ln(1 - \text{MoLCDF}(u_t, \boldsymbol{\pi_t}, \boldsymbol{\mu_t}, \mathbf{s_t}))) \end{aligned} \tag{8}$$

MoLCDF and MoLPDF are defined as:

$$\begin{aligned} \text{MoLCDF}(u_t, \boldsymbol{\pi_t}, \boldsymbol{\mu_t}, \mathbf{s_t}) &\coloneqq \sum_{i=1}^{N} \pi_{ti} \cdot \sigma(z_{ti}) \\ \text{MoLPDF}(\cdots) &\coloneqq \sum_{i=1}^{N} \pi_{ti} \cdot (z_{ti} - \ln(s_{ti}) - 2\ln(1 + e^{z_{ti}})) \\ \text{where} \quad z_{ti} &= \frac{u_t - \mu_{ti}}{s_{ti}} \end{aligned} \tag{9}$$

3.3. Generic and efficient training procedure

The student distribution with non-affine transformation is no longer a uni-modal Logistic. Because of that, we have to adapt the KLD computation in the training procedure. As in the original PW, described in section 3.1, we use Monte Carlo approximation to estimate KLD. However, in addition to computing predicted samples $\mathbf{x}$ with the reparametrization trick [37], which is required to estimate Cross-Entropy with equation 6, we also compute a Jacobian determinant with equation 8. We do that with cached transformation parameters for every noise sample sequence. The Jacobian allows us to evaluate the final student density p_S with equation 2. We use this to estimate Entropy with:

$$\begin{aligned} H(P_S) &= \mathbb{E}_{u \sim \mathcal{L}(0,1)} \left[\sum_{t=1}^{D} -\ln(p_S(x_t | \mathbf{u_{<t}})) \right] \\ &= \mathbb{E}_{u \sim \mathcal{L}(0,1)} \left[\sum_{t=1}^{D} -\ln(p_u(u_t) | det J_T(u_t)|^{-1}) \right] \end{aligned} \tag{10}$$

4. Experiments

4.1. Experimental setup

4.1.1. Training & Evaluation datasets.

All models used in evaluations were trained on internal studio-quality recordings. The dataset used for training contains 22 male and 52 female voices speaking 27 languages and dialects in 10 different speaking styles. Data were balanced so that there are approximately 3000 utterances per speaker. The dataset we use has a diverse range of speech vocoding scenarios and is motivated by our assumption that the non-affine transformation improves modeling more expressive distributions.

For evaluation, we extracted Mel-spectrograms from the original studio-quality recordings. The dataset contains 2700 recorded sentences covering 20 languages with 26 speakers in 10 different speaking styles. The dataset is balanced. There are at least 100 recordings per style and 50 per speaker. For the subset of 1950 utterances, we also generated Mel-spectrograms in a given style from the text with Tacotron-2 like systems. [2].

4.1.2. Evaluation setup

We run two types of evaluations. First is the subjective evaluation that compares the affine PW with the proposed non-affine model. We execute it as the preference tests of the TTS and waveform reconstruction tasks between the two systems. To quantify differences we run a MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) [38] evaluation of the waveform reconstruction task. Apart from the two PW systems it also includes original recordings and two other state-of-the-art neural vocoding models: WaveGlow [14] and ParallelWaveGAN [9]. Second is the objective evaluation, which compares affine and non-affine models in terms of Cross-Entropy between the teacher and student and L2 Spectral Distance of the reconstructed waveform.

For hypothesis testing with the objective metrics and MUSHRA we use a two-tailed t-test. For the preference test, we evaluate a one-tailed hypothesis with the Binomial test. We consider the difference between systems to be statistically significant if the $p - value$ is lower than 0.05. All subjective tests are executed on the Clickworker platform [39]. Each of the screens is assessed by 40 native listeners in the preference tests and 20 in the MUSHRA tests.

4.1.3. Training setup & Model details

All of the PW [13] models were distilled from a high-quality Wavenet teacher [5]. The teacher network uses 24 layers with 4 dilation doubling cycles, 128 RGCNN channels, kernel size 3, and output distribution of 10 MoLs. For the student architecture, we use 2 flows with 10 and 30 RGCNN layers with 128 channels and dilation reset every 10 layers[3]. Non-affine PW uses a mixture of 10 logistics in the transformation MoLCDF. Both models were trained on Mel-spectrogram conditioning corresponding to short audio clips, with the Adam optimizer [40] and a constant learning rate 10^{-4} for 4 million iterations with KLD and power loss [13]. The teacher uses a batch size of 64 and audio clips of 0.3625s duration, while the student uses 16 and 0.85s respectively. WaveGlow [14] and ParallelWaveGAN [9] models were trained using open-source implementations[4].

[3] Original PW [13] uses 4 flows with 10, 10, 10, 30 RGCNN layers utilizing 64 channels. We use different hyperparameters that in our case improve the student quality.

[4] github.com/NVIDIA/waveglow github.com/kan-bayashi/ParallelWaveGAN

4.2. Objective evaluation

To objectively evaluate the differences between the non-affine and original PW, we propose two benchmarking metrics. The first is the L2 Spectral Distance between the original waveform $\mathbf{x}$ and a waveform reconstructed from a Mel-spectrogram $\hat{\mathbf{x}}$. We transform the signal to spectrum using the short-time Fourier transform (STFT) with hop-size of 256 samples and 1024 bins. The metric is computed as $\||STFT(\mathbf{x})| - |STFT(\hat{\mathbf{x}})|\|_2$. The second is the Cross-Entropy between student and teacher under the student distribution. The latter can be interpreted as a negative log-likelihood, which is a common metric used to evaluate NF [35]. It is computed with a Monte Carlo approximation as outlined in equation 6. In Table 1, we present average results of objective metrics attained by affine and non-affine transformations across different speaking styles. The non-affine PW outperforms the original model on every style. The results are statistically significant for all styles except News Briefing. For more subjectively expressive styles, like Singing, we observe a bigger relative difference between affine and non-affine models than for less expressive ones, like Neutral.

Table 1: *Average objective metrics with confidence interval of 95% computed on the test-set. Lower (better) numbers that are different with statistical signficance ($p - val < 0.05$, two-tailed t-test) are in bold. Results are sorted by relative difference (RD) between affine and non-affine transformation on L2 Spectral Distance.*

Style	L2 Spectral Distance			Cross Entropy		
	affine	non-affine	RD	affine	non-affine	RD
News briefing	0.083±0.007	**0.078**±0.001	-6.3%	4.96±0.07	**4.92**±0.07	-8.7‰
Singing	0.073±0.005	**0.068**±0.005	-6.1%	4.97±0.07	**4.95**±0.07	-4.3‰
Spelling	0.051±0.003	**0.049**±0.003	-4.2%	4.35±0.07	**4.32**±0.07	-6.2‰
Disc Jockey	0.058±0.003	**0.055**±0.003	-4.0%	4.71±0.06	**4.68**±0.06	-7.2‰
Jokes	0.055±0.003	**0.053**±0.003	-3.5%	4.27±0.06	**4.25**±0.05	-5.7‰
Long Form	0.056±0.004	**0.055**±0.004	-3.1%	4.67±0.08	**4.64**±0.08	-6.9‰
Emotional	0.072±0.005	**0.070**±0.005	-3.0%	4.88±0.04	**4.86**±0.04	-5.6‰
Whispering	0.314±0.008	**0.305**±0.007	-2.9%	5.56±0.06	**5.55**±0.06	-2.4‰
Conversational	0.078±0.005	**0.077**±0.005	-2.1%	5.02±0.05	**4.98**±0.05	-7.4‰
Neutral	0.066±0.003	**0.065**±0.003	-2.0%	4.62±0.02	**4.60**±0.02	-6.6‰
Overall	0.089±0.003	**0.086**±0.003	-2.8%	4.76±0.02	**4.73**±0.02	-6.1‰

4.3. Subjective evaluation

To understand the subjective preference of naive listeners between the proposed non-affine model, the original PW, and other state-of-the-art neural vocoding systems WaveGlow [14] and ParallelWaveGAN [9], we run three perceptual evaluations.

To quantify differences between all of the systems we evaluate the waveform reconstruction task with MUSHRA test. We ask listeners to rate the voices in terms of their naturalness, paying attention to the quality of the audio signal and articulation clarity. 100 means the most natural and highest audio quality speech; 0 means the least natural and lowest audio quality speech. Overall the non-affine transformation outperforms affine ParallelWavenet with statistical significance $p - val < 0.05$. The non-affine transformation closes the gap from the original PW to recordings by 10%, and from other state-of-the-art neural vocoding systems by more than 60%. Non-affine PW achieves 95.35% Relative MUSHRA, when compared to recordings, while affine PW has 94.83%. Detailed results are reported in Table 2.

To get a more sensitive subjective preference between affine and non-affine systems, we evaluate the waveform reconstruction task with a simple preference test. Listeners can select either preference towards one of the systems or no-preference. In case of no-preference, we split the votes equally between both systems. Overall preference towards the non-affine model is confirmed with statistical significance $p-val < 0.05$. Results across different speaking styles are statistically significant for Spelling, Singing, News briefing, and Jokes. For these, the non-affine PW is preferred for all except News briefing. Results are presented in Table 3.

Table 2: *The MUSHRA evaluation of naturalness and signal-quality. Systems are: affine (A-PW), non-affine (NA-PW) Parallel Wavenet, WaveGlow (WG), ParallelWaveGAN (PWG), and recordings (Rec.). The highest (best) scores are in bold. '*' means that the difference between A-PW and NA-PW is statistically significant ($p - val < 0.05$, two-tailed t-test). Results are sorted by score of recordings.*

Style	Rec.	NA-PW	A-PW	PWG	WG
Singing	73.59	**64.41**	63.93	49.04	50.79
Spelling	72.39	**69.34***	68.00	63.48	59.30
Jokes	71.10	**68.89***	67.56	64.18	55.04
Neutral	70.73	**67.63***	66.89	61.71	53.52
News briefing	70.00	68.04	**68.17**	64.81	61.97
Disc Jockey	68.64	**66.91**	66.59	63.56	61.85
Conversational	66.96	**67.00**	66.81	63.34	61.38
Emotional	66.78	65.56	**66.34***	62.81	61.46
Long Form	66.42	65.90	**66.49**	63.57	62.84
Whispering	61.97	**54.07**	54.00	34.64	43.60
Overall	68.99	**65.78***	65.42	59.01	55.38

To understand if non-affine improvements hold for Mel-spectrograms synthesized with a Tacotron-2 like NTTS system we run an additional preference test. The non-affine system overall outperforms the affine one with statistical significance $p - val < 0.05$. Results for specific styles are mostly statistically insignificant, except Whisper which is better with the non-affine model. Results are reported in the Table 3

Table 3: *The preference tests between affine and non-affine PW in the task of waveform reconstruction and TTS. Numbers correspond to the percent of votes towards the given system. No-preference votes are split between the two systems equally. The preferred system scores are in bold. '*' means that results are statistically significant ($p - val < 0.05$, one-tailed binomial test). Results are sorted by preference in reconstruction task.*

Style	Reconstruction		TTS	
	non-affine	affine	non-affine	affine
Spelling	**51.92***	48.08	**50.87**	49.13
Singing	**51.81***	48.19	-	-
Jokes	**51.65***	48.35	49.45	**50.55**
Conversational	**50.42**	49.58	**50.16**	49.84
Emotional	**50.35**	49.65	**50.33**	49.67
Whispering	**50.28**	49.72	**51.16***	48.84
Long Form	**50.16**	49.84	**50.03**	49.97
Neutral	**50.12**	49.88	**50.42**	49.58
Disc Jockey	49.08	**50.92**	49.77	**50.23**
News briefing	49.01	**50.99**	**50.45**	49.55
Overall	**50.29***	49.71	**50.42***	49.58

5. Conclusions & Future Work

In this work, we presented a general improvement to the probability density modeling power of Normalizing Flows (NF) used in neural vocoding. We enhanced Parallel Wavenet (PW) with a monotonically increasing non-affine activation function. The proposed model closed the naturalness and signal quality gap from the original PW to recordings by 10%, and from other state-of-the-art neural vocoding systems by more than 60%. It also reduced the L2 Spectral Distance and the Cross-Entropy computed on the multi-speaker, multi-lingual test set by 3% and 6‰ compared to the affine PW. For more expressive styles like Singing, we observe more improvements than for less expressive ones.

This work motivates several possible directions for further research. 1) Non-affine NF might significantly reduce the memory footprint and the number of floating-point operations in the neural vocoding. It is reported in other domains [23, 24] that non-affine models achieve the same or better quality than the affine ones with a much lower number of layers. 2) The non-affine transformation might simplify the complex teacher selection process for knowledge distillation models. 3) Finally, it might help to improve vocoding quality of other NF-based neural vocoding architectures.

6. References

[1] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly *et al.*, "Tacotron: A fully end-to-end text-to-speech synthesis model," *CoRR*, vol. abs/1703.10135, 2017.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang *et al.*, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *ICASSP*, 2018, pp. 4779–4783.

[3] S. Vasquez and M. Lewis, "Melnet: A generative model for audio in the frequency domain," *arXiv preprint arXiv:1906.01083*, 2019.

[4] H. Jun, R. Child, M. Chen, J. Schulman, A. Ramesh, A. Radford *et al.*, "Distribution augmentation for generative modeling," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 5006–5019.

[5] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves *et al.*, "Wavenet: A generative model for raw audio," in *The 9th ISCA Speech Synthesis Workshop*, 2016, p. 125.

[6] R. V. Cox, S. F. D. C. Neto, C. Lamblin, and M. H. Sherif, "Itu-t coders for wideband, superwideband, and fullband speech communication [series editorial]," *IEEE Communications Magazine*, vol. 47, no. 10, pp. 106–109, 2009.

[7] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart *et al.*, "Efficient neural audio synthesis," in *Proceedings of the 35th International Conference on Machine Learning ICML*, 2018, pp. 2415–2424.

[8] J.-M. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5891–5895.

[9] R. Yamamoto, E. Song, and J. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP*, 2020, pp. 6199–6203.

[10] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo *et al.*, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 14 881–14 892.

[11] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *arXiv preprint arXiv:2010.05646*, 2020.

[12] A. Mustafa, N. Pia, and G. Fuchs, "Stylemelgan: An efficient high-fidelity adversarial vocoder with temporal adaptive normalization," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6034–6038.

[13] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," in *Proceedings of the 35th International Conference on Machine Learning ICML*, 2018, pp. 3915–3923.

[14] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP*, 2019, pp. 3617–3621.

[15] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," in *International Conference on Learning Representations (ICLR)*, 2019.

[16] S. Kim, S.-G. Lee, J. Song, J. Kim, and S. Yoon, "Flowavenet: A generative flow for raw audio," *arXiv preprint arXiv:1811.02155*, 2018.

[17] N. Wu and Z. Ling, "Waveffjord: Ffjord-based vocoder for statistical parametric speech synthesis," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7214–7218.

[18] H. Kim, H. Lee, W. H. Kang, S. J. Cheon, B. J. Choi, and N. S. Kim, "Wavenode: A continuous normalizing flow for speech synthesis," *arXiv preprint arXiv:2006.04598*, 2020.

[19] Y. Jiao, A. Gabryś, G. Tinchev, B. Putrycz, D. Korzekwa, and V. Klimkov, "Universal neural vocoding with parallel wavenet," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6044–6048.

[20] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (gans)," in *International Conference on Machine Learning*. PMLR, 2017, pp. 224–232.

[21] B. Wu, S. Zhao, C. Chen, H. Xu, L. Wang, X. Zhang *et al.*, "Generalization in generative adversarial networks: A novel perspective from privacy protection," *arXiv preprint arXiv:1908.07882*, 2019.

[22] T. Müller, B. McWilliams, F. Rousselle, M. Gross, and J. Novák, "Neural importance sampling," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, pp. 1–19, 2019.

[23] N. De Cao, W. Aziz, and I. Titov, "Block neural autoregressive flow," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 1263–1273.

[24] C. Huang, D. Krueger, A. Lacoste, and A. C. Courville, "Neural autoregressive flows," *CoRR*, vol. abs/1804.00779, 2018.

[25] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, "Flow++: Improving flow-based generative models with variational dequantization and architecture design," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2722–2730.

[26] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[27] D. P. Kingma, T. Salimans, and M. Welling, "Improving variational inference with inverse autoregressive flow," *CoRR*, vol. abs/1606.04934, 2016.

[28] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *arXiv preprint arXiv:1807.03039*, 2018.

[29] P. Jaini, K. A. Selby, and Y. Yu, "Sum-of-squares polynomial flow," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3009–3018.

[30] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, "Neural spline flows," *arXiv preprint arXiv:1906.04032*, 2019.

[31] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *arXiv preprint arXiv:1912.02762*, 2019.

[32] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, "Cubic-spline flows," *arXiv preprint arXiv:1906.02145*, 2019.

[33] J. Behrmann, W. Grathwohl, R. T. Chen, D. Duvenaud, and J.-H. Jacobsen, "Invertible residual networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 573–582.

[34] R. T. Chen, J. Behrmann, D. Duvenaud, and J.-H. Jacobsen, "Residual flows for invertible generative modeling," *arXiv preprint arXiv:1906.02735*, 2019.

[35] I. Kobyzev, S. Prince, and M. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2020.

[36] A. v. d. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with pixelcnn decoders," *arXiv preprint arXiv:1606.05328*, 2016.

[37] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[38] I. Recommendation, "BS. 1534-1. method for the subjective assessment of intermediate sound quality (MUSHRA)," *International Telecommunications Union, Geneva*, 2001.

[39] Clickworker, "https://www.clickworker.com/machine-learning-ai-artificial-intelligence/," August, 2020.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations ICLR*, 2015.