Interactive Artistic Text-To-Voice: Tungnaá and Bla Blavatar vs Jaap Blonk

Victor Shepardson
Intelligent Instruments Lab
University of Iceland
Reykjavík, Iceland
victorshepardson@hi.is

Jonathan Chaim Reus
EMUTE Lab
University of Sussex
Brighton, UK
J.reus@sussex.ac.uk

Thor Magnusson
Intelligent Instruments Lab
University of Iceland
Reykjavík, Iceland
thormagnusson@hi.is

Abstract

Advances in deep learning have enabled speech synthesis to rival human speech in realism. While many artists have experimented with these technologies, real-time applications have been limited. We define a new task, interactive artistic text-to-voice (IATV), in order to bridge this gap. We also present a novel IATV system which achieves low-latency synthesis, interactivity, and controllability while allowing for exploration of unconventional vocal expressions. It leverages a character-level text encoder, Tacotron2-based streaming alignment, and a RAVE streaming vocoder. Tungnaá is our open source Python package implementing IATV training and real-time inference, plus a graphical interface for experimental music performance with IATV models. We report on strategies for low-resource training on artist-created datasets, and on an artistic application of Tungnaá in collaboration with sound poet Jaap Blonk.

1 Introduction

Deep learning-based voice synthesis ('Deep VS') architectures can reproduce much of the naturalness and variability of the human voice. Such work lies mostly within the text-to-speech (TTS) domain, but artistic applications outside of TTS are gaining interest. For example, singing voice conversion (SVC) has gained increased attention through deepfakes of popular artists [1], and community software efforts like SO-VITS and RVC [2, 3]. Singing voice synthesis (SVS) has also developed rapidly [4], but research narrowly focuses on popular singing styles rooted in Western harmony.

This research landscape has led to an abundance of software oriented toward voice acting and music production. However, options are fewer in areas like interactive art, sound design, or improvised music. We see strong desire to experiment with such systems in the electronic arts community, beginning with interest in the "WaveNet Aesthetic" of realistic babbling first heard in a 2016 demo [5], and found in the recorded work of musicians like Jennifer Walshe and DadaBots [6, 7], Holly Herndon [8], Mouse on Mars [9, 10], and in the live performances of Jonathan Reus [11] and Kelsey Cotton [12]. These latter examples emphasize specific needs of live performance, predominant within the New Instruments for Musical Expression (NIME) community [13], where an early iteration of this work was shown [14].

Artistically motivated tool creation can form an important contribution to speech research more broadly, as it cultivates playful and critical reflection on speech technologies while answering calls by AI and HCI researchers for interdisciplinary modes of research integrating scientific and cultural/critical forms of inquiry [15–17]. This motivates us to define a new research task for voice synthesis within the field of interactive art, which we call interactive artistic text-to-voice (IATV).

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

In this paper we propose requirements that define the IATV task. We also describe one possible implementation for an IATV system, using RAVE [18] as a streaming vocoder, a pretrained token-free text encoder [19] to allow the creation of flexible text-based conditioning systems, and a Tacotron2-family [20] attention model as a controllable streaming alignment generator. We also describe a real-time creative interface for IATV called Tungnaá, as well as the process of creating a bespoke IATV dataset and Tungnaá model in collaboration with Dutch sound poet Jaap Blonk [21].

2 Interactive Artistic Text-to-Voice

IATV prioritizes real-time, exploratory voice synthesis. IATV systems may be non-verbal and non-note-based, and should prioritize transparency of the underlying technology to allow control over its unique sonic artefacts, which are often the most interesting aspect for artists. We define IATV as a text-conditional audio generation task $P(x|t)$, with the following requirements:

1. Real-time performance on a CPU, with latency below 100 milliseconds, suitable for interactive use in a musical performance paradigm.
2. Interactivity, with human-in-the-loop manipulation of the synthesis process.
3. Controllability, exposing the underlying neural synthesis engine to allow nuanced explorations of effects such as alignment failures, glitches and babbling.
4. Flexibility, without limitation to speech or single singing style nor method of text transcription (though assuming some temporal notation with monotonic alignment between text and audio).
5. “Hi-Fi" audio, with frequencies up to 20 kHz and dynamic range suitable for music.
6. Openness, for users to train their own models, run them locally and integrate with other tools.
7. Customizability, with training methods amenable to small datasets or bespoke text notations which may not resemble common pretraining data.

3 Related Work

Per requirement (1), IATV shares many concerns with streaming TTS. Many streaming methods [22, 23] build on FastSpeech2 [24]. These rely on the text encoder to predict token durations, and then causal layers to decode the audio-aligned text to vocoder features. Modeling durations as conditionally independent can lead to poor performance on expressive speech datasets; recent work in the FastSpeech2 lineage [25] includes a more general duration model, but still requires durations estimated by an external alignment model, which may not be available for IATV per requirement (4).

Another family of streaming TTS methods based on Tacotron2 [20] learn text-audio alignment jointly with conditional generation at the utterance level. These models compute a distribution of attention over text tokens for each frame of audio, in general depending on all past audio frames, alignments, and the input text [26]. The major drawback of Tacotron2-family models is that they can be slow to train, as their recurrent alignment module is difficult to parallelize. Nevertheless, causal alignment is an advantage for IATV at inference time per requirement (3): if a user intervenes mid-generation, a model can immediately adjust timing.

Most streaming TTS methods rely on a separate vocoder. Streaming neural vocoders include WaveRNN [27] and derivatives [28, 29]. The WaveRNN family of vocoders are efficient, but generate audio in single samples or very small blocks, requiring bespoke low-level implementations for streaming inference. In contrast, block-level models can be implemented readily in high-level machine learning frameworks, as overhead is negligible when block size is large. For such vocoders, cached, causal convolutions or block-level RNNs support a generative model based on normalizing flows or GANs [30, 31]

RAVE [18] is a variational autoencoder (VAE) for raw audio which learns a continuous latent space of audio features. An adversarial term supplements the reconstruction loss, allowing RAVE models to highly compress inputs yet reconstruct high-fidelity audio with imputed detail. In this regard, RAVE is similar to neural codec models [32]. Unlike neural codecs, RAVE’s latent representations are continuous and relatively interpretable. RAVE is both high bandwidth (44.1-48 kHz) and streaming via cached causal convolutions [33]. Some work has been done to adapt RAVE specifically for speech, with voice conversion in mind [34, 35]

4 Proposed System

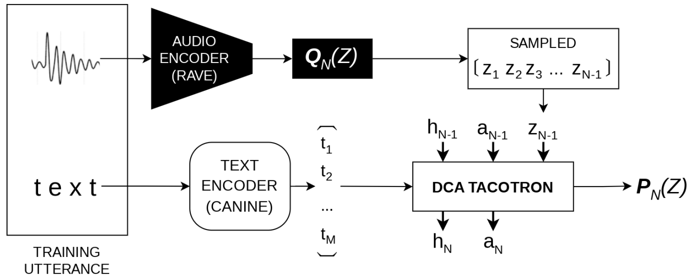

Figure 1: One audio frame of training. The RAVE Encoder is frozen, with latent $Q_n(Z)$ computed as a preprocessing step. The CANINE text encoder is fine-tuned during TTS training, except for embeddings. A Tacotron2-like module generates alignments a and audio feature distributions $P_N(z)$, conditioned on text encoding T and past frames z_{n-1} via hidden states h.

We present a streaming text-to-voice system which requires minimal text preprocessing (no tokenization, phonemization, or forced alignments), and provides further controllability via direct manipulation of text-audio alignments and vocoder parameters. Complete implementation details can be found in the code repository included in supplementary materials.

4.1 Streaming Alignment Model

Per requirements (4,7), we choose a Tacotron2-like architecture which can learn alignments from utterance-level text and audio pairs, avoiding a forced alignment step reliant on tools which might be out of domain for artist-created datasets. Specifically, we use Dynamic Convolutional Attention (DCA) [26], which mitigates the instability of purely content-based or location-sensitive attention but allows for creative (mis)use per (3).

The original Tacotron2 minimizes mean squared error between audio features; in the probabilistic view, it maximizes likelihood of each audio feature frame z_i given previous frames $z_{<i}$ and text t under a Gaussian density:

$$P_\theta^{\text{Tacotron2}}(z_i|z_{<i}, t) := \mathcal{N}(f(z_{<i}, t; \theta), \mathbb{1}) \tag{1}$$

where f is the neural network. We enhance the generative model using a mixture density head [36]:

$$P_\theta(z_i|z_{<i}, t) := \sum_{j=1}^{N} \pi_j \mathcal{N}(\mu_j, \operatorname{diag}(\sigma_j)) \qquad \pi, \mu, \sigma = f(z_{<i}, t; \theta) \tag{2}$$

where j indexes the mixture component, π is nonnegative and sums to 1. Compared to Tacotron2, this model quickly learns alignments and models diverse prosody without further conditioning. Our implementation derives from Coqui TTS [37] (Mozilla Public License 2.0).

4.2 Pre-trained, Tokenless Text Encoder

Considering requirement (6), we use a pre-trained CANINE language model [19] (Apache 2.0 licensed) as our text encoder. CANINE is a masked language model similar to BERT [38] which is trained on large-scale text data and can be built upon for downstream tasks. In contrast to other BERT-likes, CANINE represents text as unicode code-points, merely appending start and end of sequence tokens. CANINE's transformer layers give it complexity quadratic in the length of the text, but as authors of other streaming TTS systems [39, 23] have observed, the time to encode text remains small compared to TTS sampling and vocoding for short utterances. Our interactive setting favors short texts, so we accept a non-streaming text encoder, leaving a causally masked text encoder to future work.

We freeze the CANINE character embeddings, reasoning that preserving the pretrained embedding geometry will allow users to explore out-of-domain behavior of models in an intuitive, textually driven manner, in line with requirement (3). For the Jaap Blonk models described in section 6, we use only the embeddings, i.e. no transformer layers.

4.3 Real-time VAE Vocoder

RAVE [18] is used as a vocoder. Per requirement (5), it attains a high quality of sound via an adversarial objective and employment of signal processing layers. Per requirements (6,2), RAVE has both an open (CC-BY-NC-4.0) implementation of the training code, and streaming inference which is well-integrated with computer music workflows. Because audio features are learned via a VAE, almost any latent vector under the prior will decode to a data-like sound, making RAVE's latent space more suitable for exploratory manipulation than spectrogram-based vocoders, per requirements (2,3).

4.4 Low-resource Training Strategies

Because IATV is intended to support artists working with unusual voice sounds or notation systems, we explore low-resource training strategies without the benefit of substantial unpaired in-domain text and audio data as might be available for a low-resource language. For artistic purposes there are ethical and conceptual implications when involving pretraining datasets; in this research we explore how far we can push small datasets without generalization from large-scale data.

As noted in section 4.1, the standard normal likelihood is inadequate for expressive voice. However, we find that more expressive models are prone to overfit small datasets before meaningful text-audio alignments can develop: subjective quality still improves once validation increases, complicating evaluation. We mitigate this via techniques to delay overfitting and accelerate alignment.

Delaying Overfitting. Larger models (up to the the sizes allowed by the real-time constraint) often attained a similar validation loss more quickly (given higher GPU utilization), making them seem initially more appealing. However, smaller models proved better, because alignment is able to develop further without overfitting taking place.

Two data augmentation strategies also delay overfitting. First, source audio is cropped by up to half a RAVE block and speed is randomized by up to a quarter tone. Since the RAVE forward pass is expensive compared to the TTS model, we precompute this augmentation, creating 63 randomized versions of each utterance. Second, pairs of utterances are randomly concatenated to form training examples, which also teaches the model to switch styles mid-utterance when encountering a style-annotating character.

Accelerating Alignment. We observed that models trained on smaller datasets would utilize the text-audio alignments, but that they would not be sharp, usually spreading substantial weight across more than three tokens at a time, which is an obstacle to requirement (3). We designed two additional loss terms to encourage sharper alignments to develop quickly. An alignment *disperson* loss penalizes the entropy of attention distributions:

$$\mathcal{L}_d = \max(\gamma_d, \mathbb{E}_t[\mathrm{H}(a_t))] \tag{3}$$

Where a_t is the normalized vector of attention weights for each token at time step t. This term is clipped to a minimum value $\gamma_d = 1.5$ nats so that it is has zero gradient once alignments are sufficiently sharp. Despite the clipping, we find $\mathcal{L}_d$ can still cause pathological alignments which don't advance through the text, and so introduce a second *concentration* term which penalizes unused entropy in the marginal token weights:

$$\mathcal{L}_c = \max(\gamma_c, 1 - \frac{\mathrm{H}(\mathbb{E}_t[a_t])}{\ln T}) \tag{4}$$

Where T is the number of text tokens. Again we clip to a minimum value $\gamma_c = 0.5$ so the term only takes effect when the alignment is substantially concentrated on part of the text. Finally, we propose a simplified DCA alignment, where the dynamic convolution kernel is derived from just four parameters via a difference of Gaussians (DGDCA):

$$k(\alpha, \sigma_1, \sigma_2, \mu) = (1+\alpha)\mathcal{N}(\tau\mu, \sigma_1) - \alpha\mathcal{N}(\tau\mu, \sigma_1 + \sigma_2) \tag{5}$$

This kernel can only blur, sharpen, and move the alignment forward by a limited amount, preventing many kinds of alignment failures possible for DCA. The hyperparameter τ replaces the prior filter in the DCA. The four kernel parameters are taken from an affine transformation of the controller RNN state followed by a logistic sigmoid constraining them all to the interval $[0, 1]$.

5 Model Training

We train the proposed system using a new dataset created for the IATV task. This is a non-verbal voice dataset created in collaboration with the sound poet Jaap Blonk. Blonk's vocalizations are annotated using a system devised by Blonk called reduced phonetic alphabet (RPA) plus a set of special characters to denote transition into four moods: neutral, aggressive, happy, or worried. This dataset totals about 1 hour of audio and 20,000 characters.

Vocoder training follows that of a standard [1] causal RAVE v3 training procedure with two exceptions. One, the β warmup schedule is lengthened, resulting in a greater number of meaningful latent dimensions. Two, we modify the RAVE objective, cropping the KL-divergence term to remove dependence on zero-padding. This resolved a problem where the RAVE posterior sometimes collapses to the prior, leading to babbling in place of silence.

Next, the TTS model is trained. The text encoder is initialized from a pretrained CANINE-C model. We preprocess raw audio through the RAVE encoder, storing mean and variance of the RAVE posterior $Q(z|x)$. During training, we sample from Q, minimizing the KL-divergence of the RAVE posterior from the TTS model:

$$\min_{\theta} \mathop{\mathbb{E}}_{z \sim Q} [\log Q(z|x) - \log P_\theta(z|t)] \tag{6}$$

where $P(z|t)$ is the TTS model. Since Q is frozen, this is the same as maximizing likelihood of the TTS model (Eq. 2). We use AdamW optimization [40] with learning rate 3e-4, $\beta_1 = 0.9$, $\beta_2 = 0.998$, weight decay 1e-6, and gradient clipping to an L2 norm of 5.

6 Low-resource Ablations

In this experiment, We ablate the low-resource training strategies described in section 4.4: dispersion and concentration loss, DGDCA aligment, model size and data augmentation. Most parameters are in the main LSTM which has two layers and dimension 512 or 1024. All models are trained to 10,000 steps on the Jaap Blonk dataset using a batch size of 24. using an Nvidia A5000, alignment model training takes less than one GPU-day and RAVE training takes about one GPU-week.

In Figure 2, validation likelihood appears similar between DGDCA and DCA, but worse without $\mathcal{L}_d$ and $\mathcal{L}_c$. Overfitting is apparent with a larger LSTM or without the randomized speed and cropping data augmentation. In Figure 3, typical inference-time alignments are shown for the two alignment modules with and without the extra loss terms. Without $\mathcal{L}_d$ and $\mathcal{L}_c$, neither alignment becomes sharp on this dataset, particularly DCA. With the extra terms, both become sharp, but the DCA becomes more finely structured, sometimes backtracking.

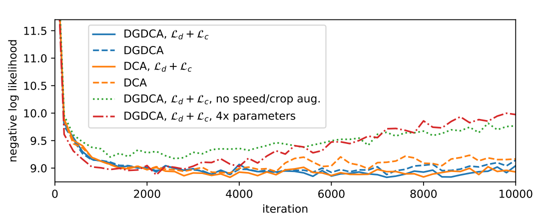

Figure 2: Validation negative log likelihood over training.

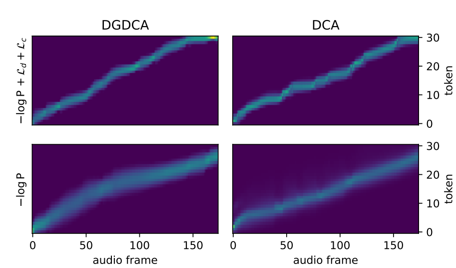

Figure 3: Alignments using DCA (right) and DGDCA (left); with dispersion and concentration loss (top) and without (bottom).

[1]`https://github.com/acids-ircam/RAVE`

7 Creative Applications

Tungnaá is a Python package[2] which incorporates training code for our models, real-time inference, and a front-end musical instrument interface. A video demonstration is online[3]. The inference engine is built in Python using PyTorch and TorchScript [41] , enabling low latency streaming inference for audio. The Tungnaá GUI runs in a separate Python process using Qt. Functions of the GUI can be controlled remotely using Open Sound Control [42] for integration with common computer music software. Tungnaá's GUI consists of three areas: text entry, alignment, and vocoder (Figure 4).

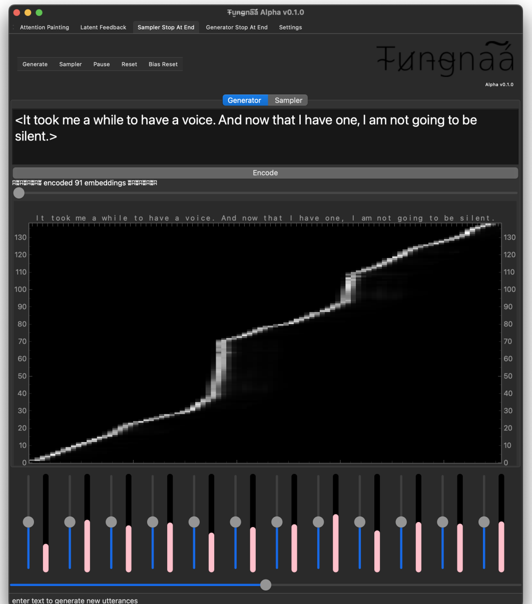

Figure 4: The Tungnaá GUI.

In the text entry field, a performer can provide text for Tungnaá to vocalise. The text can be sent on command to the text encoder, resetting the alignments.

A scrolling alignment graph depicts progress through the text over time. Once encoded, an input text appears along the horizontal axis of the graph, while time is on the vertical axis, with the present time at the top. Light pixels denote the portion of the text being used at a given time.

If *attention painting* mode is engaged, the *paint bar* allows the performer to directly manipulate alignment. If not, the text is read through according to the learned alignment model. Clicking into the graph instantly moves the alignment.

In the vocoder section, each RAVE latent dimension is displayed by a level meter, while manipulating its slider applies a bias. A temperature control affects how variable they are.. The *latent feedback* switch selects whether the biased latents are fed back into the alignment model to affect timing.

7.1 Bla Blavatar vs Jaap Blonk

Tungnaá was used in a collaboration with the Dutch sound poet Jaap Blonk for the live performance *Bla Blavatar vs. Jaap Blonk*[4]. Sound poetry has origins in the European Futurist and Dada art movements, blurring the lines between poetry and music performance. It involves exploration of abstract voice sounds and the invention of new voice notation systems. Sound poetry is also meant to be experienced live, making it an ideal case study for IATV.

In collaboration with Blonk, we created an initial IATV dataset consisting of approximately two hours of voice audio (section 5). In each performance, Blonk performs a new algorithmically generated, phonetically balanced RPA score. Throughout the performance he "battles" the Bla Blavatar, an artificial sound poet controlled by a second performer using Tungnaá. All of Blonk's vocalisations are recorded live and matched to the RPA score, adding new utterances to the dataset with each performance, and contributing to the next model in an iterative process.

8 Conclusion

We proposed interactive artistic text-to-voice, a Deep VS task meeting requirements for real-time artistic exploration and performance. Our Tacotron2-like architecture combines a token-free text encoder and stronger acoustic model with a streaming RAVE vocoder and novel strategies for low-resource training on an artist-created dataset. Our qualitative results, open-source software and creative applications demonstrate real-time and human-in-the-loop control during inference.

[2] `https://pypi.org/project/tungnaa`
[3] `https://www.goethe.de/ins/pt/en/kul/sup/web/pws.html#lightbox-10878284`
[4] `https://jonathanreus.com/portfolio/bla-blavatar-vs-jaap-blonk/`

References

[1] J. Coscarelli, "An a.i. hit of fake 'drake' and 'the weeknd' rattles the music world," *The New York Times*, Apr. 2023. [Online]. Available: https://www.nytimes.com/2023/04/19/arts/music/ai-drake-the-weeknd-fake.html

[2] "svc-develop-team/so-vits-svc," Feb. 2025, original-date: 2023-03-10. [Online]. Available: https://github.com/svc-develop-team/so-vits-svc

[3] "RVC-Project/Retrieval-based-Voice-Conversion-WebUI," Feb. 2025, original-date: 2023-03-27. [Online]. Available: https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI

[4] Y.-P. Cho, F.-R. Yang, Y.-C. Chang, C.-T. Cheng, X.-H. Wang, and Y.-W. Liu, "A survey on recent deep learning-driven singing voice synthesis systems," in *Proc. AIVR 2021*, Nov 2021, p. 319–323.

[5] A. v. d. Oord *et al.*, "Wavenet: A generative model for raw audio," Sep 2016, arXiv:1609.03499 [cs].

[6] Z. Zukowski and C. J. Carr, "Generating black metal and math rock: Beyond bach, beethoven, and beatles," Nov 2018, arXiv:1811.06639 [cs, eess].

[7] G. K. Haggett, "Jennifer walshe, a late anthology of early music vol. 1: Ancient to renaissance. bandcamp." *Tempo*, vol. 75, no. 295, p. 112–115, Jan. 2021.

[8] H. Herndon, *PROTO*. Stanford University, 2019, dMA thesis. [Online]. Available: https://purl.stanford.edu/fh292ky0538

[9] *AAI / Anarchic Artificial Intelligence / With Mouse on Mars and Birds on Mars*, Mar. 2021. [Online]. Available: https://www.youtube.com/watch?v=G8yykZjJhFU

[10] *Krach software*. Birds on Mars, 2021. [Online]. Available: https://www.krach.ai

[11] J. C. Reus, "iː goʊ weɪ," *AIMC 2023*, Aug 2023. [Online]. Available: https://aimc2023.pubpub.org/pub/hpy32yre/release/4

[12] K. Cotton and K. Tatar, "Sounding out extra-normal AI voice: Non-normative musical engagements with normative AI voice and speech technologies," *AIMC 2024*, Aug. 2024. [Online]. Available: https://aimc2024.pubpub.org/pub/extranormal-aivoice/release/1

[13] R. Kleinberger, N. Singh, X. Xiao, and A. v. Troyer, "Voice at nime: a taxonomy of new interfaces for vocal musical expression," in *Proc. New Interfaces for Musical Expression*, Jun 2022. [Online]. Available: https://nime.pubpub.org/pub/180al5zt/release/1

[14] V. Shepardson, J. Reus, and T. Magnusson, "Tungnaá: a Hyper-realistic Voice Synthesis Instrument for Real-Time Exploration of Extended Vocal Expressions," Oct. 2024, pages: 536–540 Publication Title: Proceedings of the International Conference on New Interfaces for Musical Expression Publisher: Zenodo. [Online]. Available: https://zenodo.org/records/13904943

[15] P. E. Agre, "Toward a critical technical practice: Lessons learned in trying to reform ai," in *Social science, technical systems, and cooperative work*. Psychology Press, 1998, pp. 131–157.

[16] L. Morrison and A. McPherson, "Entangling Entanglement: A Diffractive Dialogue on HCI and Musical Interactions," 2024.

[17] A.-K. Kaila and B. L. T. Sturm, "Agonistic Dialogue on the Value and Impact of AI Music Applications," *AIMC 2024*, Aug. 2024. [Online]. Available: https://aimc2024.pubpub.org/pub/p3rww87r/release/1

[18] A. Caillon and P. Esling, "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," *arXiv:2111.05011 [cs, eess]*, Nov. 2021, arXiv: 2111.05011 version: 1.

[19] J. H. Clark, D. Garrette, I. Turc, and J. Wieting, "CANINE: Pre-training an Efficient Tokenization-Free Encoder for Language Representation," *Trans. ACL*, vol. 10, pp. 73–91, Jan. 2022, arXiv:2103.06874 [cs].

[20] J. Shen *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," Feb 2018, arXiv:1712.05884 [cs].

[21] V. Shepardson, J. Reus, and T. Magnusson, "Tungnaá in live performance: An implementation of interactive artistic text-to-voice," in *Proc. Interspeech*, 2025.

[22] C. Wu, Z. Xiu, Y. Shi, O. Kalinli, C. Fuegen, T. Koehler, and Q. He, "Transformer-Based Acoustic Modeling for Streaming Speech Synthesis," in *Proc. Interspeech*. ISCA, Aug. 2021, pp. 146–150. [Online]. Available: https://www.isca-speech.org/archive/interspeech_2021/wu21b_interspeech.html

[23] G. Shopov, S. Gerdjikov, and S. Mihov, "StreamSpeech: Low-Latency Neural Architecture for High-Quality on-Device Speech Synthesis," in *Proc. ICASSP*, Jun. 2023, pp. 1–5.

[24] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and High-Quality End-to-End Text to Speech," *arXiv:2006.04558 [cs, eess]*, Mar. 2021, arXiv: 2006.04558.

[25] A. Abbas, T. Merritt, A. Moinet, S. Karlapati, E. Muszynska, S. Slangen, E. Gatti, and T. Drugman, "Expressive, variable, and controllable duration modelling in tts," Jun 2022, arXiv:2206.14165 [cs, eess].

[26] E. Battenberg, R. J. Skerry-Ryan, S. Mariooryad, D. Stanton, D. Kao, M. Shannon, and T. Bagby, "Location-relative attention mechanisms for robust long-form speech synthesis," Apr 2020, arXiv:1910.10288 [cs, eess].

[27] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient Neural Audio Synthesis," Feb. 2018, arXiv: 1802.08435.

[28] J.-M. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *Proc. ICASSP*. IEEE, May 2019, p. 5891–5895.

[29] K. Matsubara, T. Okamoto, R. Takashima, T. Takiguchi, T. Toda, Y. Shiga, and H. Kawai, "Full-band lpcnet: A real-time neural vocoder for 48 khz audio with a cpu," *IEEE Access*, vol. 9, p. 94923–94933, 2021.

[30] B. Wu, Q. He, P. Zhang, T. Koehler, K. Keutzer, and P. Vajda, "Fbwave: Efficient and scalable neural vocoders for streaming text-to-speech on the edge," Nov 2020, arXiv:2011.12985.

[31] A. Mustafa, J.-M. Valin, J. Büthe, P. Smaragdis, and M. Goodwin, "Framewise wavegan: High speed adversarial vocoder in time domain with very low computational complexity," Mar 2023, arXiv:2212.04532 [cs, eess].

[32] Y. Wu, I. D. Gebru, D. Marković, and A. Richard, "Audiodec: An open-source streaming high-fidelity neural audio codec," in *Proc. ICASSP*, Jun 2023, p. 1–5.

[33] A. Caillon and P. Esling, "Streamable Neural Audio Synthesis With Non-Causal Convolutions," *arXiv:2204.07064 [cs, eess, stat]*, Apr. 2022, arXiv: 2204.07064.

[34] A. R. Bargum, S. Lajboschitz, and C. Erkut, "RAVE for Speech: Efficient Voice Conversion at High Sampling Rates," Aug. 2024, arXiv:2408.16546 [cs].

[35] S. Nercessian, "P-RAVE: Improving RAVE through pitch conditioning and more with application to singing voice conversion," 2023.

[36] C. M. Bishop, "Mixture density networks," Monograph, 1994, num Pages: 26 Place: Birmingham Publisher: Aston University. [Online]. Available: https://publications.aston.ac.uk/id/eprint/373/

[37] G. Eren and The Coqui TTS Team, "Coqui TTS," Jan. 2021. [Online]. Available: https://github.com/coqui-ai/TTS

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018, arXiv: 1810.04805.

[39] N. Ellinas *et al.*, "High quality streaming speech synthesis with low, sentence-length-independent latency," in *Proc. Interspeech*. ISCA, 2020. [Online]. Available: https://www.isca-speech.org/archive/interspeech_2020/ellinas20_interspeech.html

[40] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," Jan. 2019, arXiv: 1711.05101.

[41] J. Ansel *et al.*, "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation," in *Proc. 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. Association for Computing Machinery, 2024.

[42] M. Wright, "OpenSoundControl Specification 1.0," Mar. 2002. [Online]. Available: https://opensoundcontrol.stanford.edu/spec-1_0.html

A Hyperparameters

For the experiments in section 6, the following base commands are used (relative to the `tungnaa` package version 0.1.1):

```
tungnaa prep --datasets '{kind:"dadaset_v2", path:$DATA_PATH}' \
--rave-path $RAVE_PATH --out-path $PREP_STORAGE --aug_n 63

tungnaa trainer --experiment $NAME --device $DEVICE \
--model-dir $CHECKPOINT_STORAGE --log-dir $LOG_STORAGE \
--manifest $PREP_STORAGE/manifest.json --rave-model $RAVE_TS_PATH \
--batch-max-frames 768 --batch-size 24 --epoch-size 200 --valid-size 64 \
--model '{attention_type:gauss,tokens_per_frame:0.25,
text_encoder_type:canine_embedding,prenet_dropout:0,dropout:0.2,
likelihood_type:mixture,rnn_layers:2,rnn_size:512,decoder_type:None,
prenet_layers:1,text_encoder:{bottleneck:32,use_positions:False},
prenet_wn:True,proj_wn:True,attn_wn:True}' \
--freeze-embeddings True \
--concentration-norm-scale 0.1 --concentration-cutoff 0.5 \
--dispersion-scale 0.1 --dispersion-cutoff 1.5 \
--style-annotate 1 --concat-speakers 2 \
train
```

i.e., the above hyperparameters correspond to the solid blue line in figure 2 (DGDCA, $\mathcal{L}_d+\mathcal{L}_c$), and the others vary `aug_n`, `attention_type`, `concentration_norm_scale`, `concentration_cutoff`, `dispersion_scale`, `dispersion_cutoff` and/or `rnn_size`.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our contributions are stated clearly in the abstract and elaborated in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [NA]

   Justification: The paper's main contribution is to formalize a new symbol-to-voice task for artistic use, and to offer one possible implementation of that task in the form of a ready-to-use piece of open source software, but we do not claim that this is the most optimal or only way to address the requirements of the task. Other implementations are certainly possible, and encouraged, but this is left to future research.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: There are no theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We offer all the necessary code and training hyperparameters (appendix A) for people to replicate the results, the Jaap Blonk dataset is not made public at this time, but the experiments can be repeated using any open, high-fidelity TTS training dataset, such as VCTK or HifiTTS.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide inference and training code in a public github repository, which allows for reproducing the results of training on the VCTK and HifiTTS/John van Stan datasets. The dataset of Jaap Blonk's vocalizations is an ongoing artistic project, as part of continued performances of Bla Blavatar vs. Jaap Blonk, it is our intent to release this growing dataset to the public in the future as an artistic gesture.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyperparameters, optimizer, etc are specified in section 5 and appendix A, which together with the open source codebase specify the details of each model trained.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We were unable to produce statistical evaluations due to computational and time constraints, but we note that the results reported in section 6 are consistent with what we observed during development.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: compute hardware is mentioned in section 5

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the code of ethics. We do not anticipate systemic risks or risks to individuals in connection with our released models or use of datasets. We do not use human subjects, other than Blonk, who has full agency over his participation and whose consent is explicit. Our methods are low-resource and will not contribute meaningfully to resource or energy use on a global scale.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the positive impacts in the form of contributions to critical discourse and artistic practice. We do not anticipate negative impacts from this work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do no believe there is substantial risk of misuse for our models, as they are less realistic than many existing solutions, instead prioritizing low-latency creative uses.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Software used (RAVE, Coqui TTS, CANINE) is cited and license stated

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The new assets described (the Jaap Blonk dataset) are not yet released.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Human subjects were not used in evaluations.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

Justification: We do not use human subjects, other than Blonk, who has full agency over his participation and whose consent is explicit.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLMs are not used.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.