INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Diffusion-based accent modelling in speech synthesis

Kamil Deja[1,*], *Georgi Tinchev*[2], *Marta Czarnowska*[2], *Marius Cotescu*[2], *Jasha Droppo*[2]

[1]Warsaw University of Technology
[2]Amazon

kamil.deja@pw.edu.pl

Abstract

In this work, we introduce a diffusion-based text-to-speech (TTS) system for accent modelling. TTS systems have become a natural part of our surroundings. Nevertheless, because of the complexity of accent modelling, recent state-of-the-art solutions mainly focus on the most common variants of each language. In this work, we propose to address this issue with a newly proposed diffusion generative model (DDGM). We first show how we can adapt DDGMs to the problem of accent modelling. We evaluate and compare this approach with a recent state-of-the-art solution, showing its superiority in modelling six different English accents. On top of our TTS system, we introduce a novel accent conversion method, where using the saliency map technique, we remove source accent-related features and replace them with the target ones through the diffusion process. We show that with this approach, we can perform accent conversion without a need for any additional speech information such as phonemes or text.

Index Terms: text-to-speech, accent modelling, diffusion generative models

1. Introduction

Text-to-speech (TTS) synthesis is a technology that allows computers to convert written text into spoken language. While the majority of recent systems focus on the most common versions of languages, such as American or British English, one of the key aspects that makes the usage of a synthesized voice more comfortable is the ability of the system to accurately capture the nuances of language such as accents and dialects.

Nevertheless, modelling accented speech with neural models remains a challenging problem, as they involve variations not only in pronunciation but also in rhythm and intonation. Hence, different state-of-the-art TTS systems rely on generative neural models that can create synthetic speech with different characteristics. Recent advances in this domain have led to the development of sophisticated models that can capture complex data distributions. Diffusion models [1], in particular, have shown to be effective in generating high-quality samples in variety of tasks, including the generation of images [2, 3, 4], text [5], and speech synthesis [6]. Inspired by those applications, in this work, we propose to adapt the diffusion-based generative model to handle complex accent data and generate high-quality accented speech.

In brief our contributions are as follows: 1) we introduce a new model for multi-speaker multi-accent speech synthesis based on the recently proposed GradTTS [6]; 2) we show that our method outperforms recent state-of-the-art solutions; 3) we propose a method for direct end-to-end accent conversion that does not need speech features such as phonemes or text.

2. Related Works

2.1. Accent modelling

Most speech systems that focus on accent modelling use accent conversion (AC) to transform a recorded voice into a different accent. This can be achieved in two ways. By combining spectral features via voice morphing [7, 8, 9] or through spectral and phoneme frame matching [10, 11, 12]. On top of those two approaches, there exists a line of work that relies on external features with frame-matching preceded by voice conversion [13, 14, 15, 16, 17, 18]. Several works focus on foreign accent conversion. In [19], authors propose to first to extract a phonetic posteriorgram of a non-native speaker and decode it with an acoustic model trained on the corpora of native speakers. The work is later extended in [20] with additional speaker and accent embedding models.

2.2. Diffusion models for speech synthesis

There are several methods that employ diffusion models for speech synthesis. In GradTTS [6] authors propose the adaptation of FlowTTS [21] method with diffusion decoder. In Guided-TTS [22, 23] authors combine this setup with classifier guidance [2] where diffusion model trained in unsupervised way is guided towards desired phonemes through external phoneme classifier. There are several works where diffusion models are employed to model multiple speakers [24, 25], or as a neural vocoder [26].

2.3. Data conversion with diffusion models

Outside of the speech domain diffusion models have been used for conversion tasks in image domain to alter the original data samples [27], change their domain [28], create counterfactual examples [29] or inpaint the missing parts [30]. In this work, we extend those studies to the problem of accented speech synthesis and accent conversion.

3. Background

3.1. Score-based generative models

In this work, we consider a unified method for denoising diffusion probabilistic models (DDPM) [1, 31] introduced in [32] with extension to score matching with Langevin dynamics (SMLD) by [33]. This approach is applied to text-to-speech (TTS) in the Grad-TTS [6] model, where Popov *et al.* describe the use of a diffusion process to convert any speech data distribution to the standard normal distribution:

$$dX_t = -\frac{1}{2}X_t\beta_t dt + \sqrt{\beta_t}dW_t, \quad t \in [0,T], \qquad (1)$$

[*] Work done while at Amazon.
Link to the generated samples: Amazon Science

10.21437/Interspeech.2023-154

where β_t is the pre-defined noise schedule $\beta_t = \beta_0 + (\beta_T - \beta_0)t$ and W_t is the Wiener process. Eq. 1 defines the forward diffusion process where original data sample X_0 is transformed into random noise X_T. To generate new samples a backward diffusion process is defined that reverses the forward one with trainable neural model. Following [32], Popov *et al.* use a single denoising model s_θ to predict the score function $s(X_t)$ defined as $\nabla_{X_t} \log p_t(X_t)$ [6]. Model s_θ is trained with a loss function:

$$L(\theta) = \mathbb{E}_{t,X_0,\epsilon_t}\left[\left\|s_\theta(X_t,t) + \lambda(t)^{-1}\epsilon_t\right\|_2^2\right], \qquad (2)$$

where $\lambda(t) = I - \mathrm{e}^{-\int_0^t \beta_s ds}$. New sample X_0 can be generated from random gaussian noise X_t through discretised version of the reverse stochastic differential equation:

$$X_{t-\frac{1}{N}} = X_t + \frac{\beta_t}{N}(\frac{1}{2}X_t + s(X_t)) + \sqrt{\frac{\beta_t}{N}}z_t, \qquad (3)$$

$$s(X_t) = \nabla_{X_t} \log p_t(X_t), \qquad (4)$$

where N is the number of steps in the discretized reverse process, z_t is the standard Gaussian noise and $\frac{1}{N}$ defines the size of one step (for $T = 1$).

3.2. Saliency maps

Saliency maps are used to find important information in the input from the perspective of a trained neural network. Different works use this technique to quantify and understand why machine learning models make certain decisions [34, 35, 36, 37, 38]. The general idea behind saliency maps is that the important area in the image for a given class should highly activate the convolutional features, while at the same time highly influencing the decision.

In particular, given an input X and a trained classifier q that predicts class y for this input, we can calculate the saliency map m using an internal layer of a classifier l. To that end, we first find the regions of the image that activate the convolutional filters, by calculating the activation values F_l at a given layer l. Those activations are then combined with a gradient δ_l that is backpropagated to the layer l after calculating loss wrt. target class y.

$$m(X, y) = \delta_l \mathcal{L}(q(X), y) \cdot F_l(X) \qquad (5)$$

The resulting saliency map $m(X, y)$ has the dimensionality of the layer l. If there is a pooling used for the calculation of the activations F_l, we have to upsample the resulting mask so that it reflects the shape of the input. The final map defines the importance of each input pixel from the classifier perspective. High saliency score, depict what the neural network *believes* is important to make a certain prediction. For example, if the neural network is an image classifier and the task is to predict the *dog* class, the saliency will reflect which areas of the input image are important to yield *dog* as the final prediction.

4. Method

In this work we propose a novel system for accent modelling with diffusion models. We first introduce a text-to-speech approach, followed by the implicit accent conversion method.

4.1. Text-to-speech with accent modelling

Our accented text-to-speech generation model is comprised of two parts - a text encoder module that learns the conditional average prediction for an utterance [6, 39] and a diffusion processes that refines this prediction into a final mel-spectrogram. An overview of our method is presented in Figure 1.

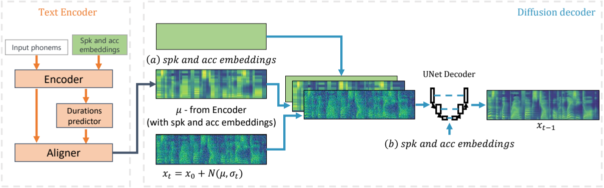


Figure 1: *Overview of our TTS model with two versions of accent conditioning (a) speaker and accent conditioning as third input channel, (b) speaker and accent conditioning as scaling of the latent UNet representations.*

4.1.1. *Text encoder*

In TTS systems, the input in the form of phonemes consists of L values $x_{1:L}$. The goal of the encoder is to roughly approximate the target mel-spectrogram $y_{1:F}$ where F is the number of acoustic frames. In our text-to-speech model, we follow the formulation from [6, 21] and train the encoder network to convert an input text sequence $x_{1:L}$ into a sequence of features $\hat{\mu}_{1:L}$. The feature sequence is then aligned to a target frame-wise features $\mu_{1:F}$ using durations predicted by a submodule of the encoder. The encoder is trained to minimize the distance between the predicted aligned output μ and the original mel-spectrogram y. We use a loss function that employs Monotonic Alignment Search and Mean Square Error in a logarithmic domain for duration prediction.

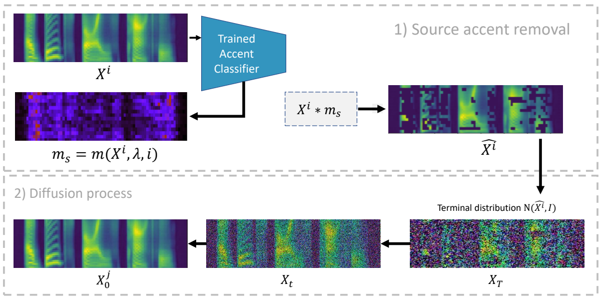
Since accent influences the duration of individual phonemes [40, 41, 42], we postulate to include the accent and speaker information already in the encoder, changing $\mu = E_\varphi(x_{1:L})$ to $\mu = E_\varphi(x_{1:L}, spk_{emb}, acc_{emb})$, where φ are the parameters of the encoder E.

4.1.2. *Diffusion decoder*

With accented speech encoded into aligned features $\mu = E_\varphi(x_{1:L}, spk_{emb}, acc_{emb})$, we use a diffusion-based generative model to refine the final mel-spectrogram [6]. To that end, we create a forward diffusion process (Equation 1), that takes original data samples and gradually noise them towards random Gaussian noise. However, to take into account the encoded features, as a terminal distribution, we use the Gaussian noise $N(\mu, I)$, where μ are aligned features from the text encoder.

For the training of the diffusion decoder, we use the loss function based on the score function as presented in Equation 2. For a diffusion model, a single denoising model s_θ is trained with a set of parameters θ to perform denoising at each diffusion timestep. To simplify the training, a timestep encoding t is used to condition the output of the decoder. Additionally, [6] propose to condition the diffusion model with the unchanged output of the text encoder, μ. In particular, for the UNet [43] architecture most commonly used as a diffusion denoiser, the timestep embeddings are used to scale the latent activations outputted by the UNet encoder, while additional conditioning with μ is added as a second channel to the input mel-spectrogram as presented in Figure 1.

In this work, we additionally propose to condition the denoising decoder model on speaker and accent embeddings. We evaluate two approaches where embeddings similarly to μ are added as a third input channel *(input conditioning)* or combined with the timestep embeddings t are used to scale the latent activations *(latent conditioning)*. In the first approach, the UNet encoder encodes the mel-spectrogram while being aware of the



X^i
Trained Accent Classifier
1) Source accent removal
$m_s = m(X^i, \lambda, i)$
$X^i * m_s$
$\widehat{X^i}$
2) Diffusion process
Terminal distribution $N(\widehat{X^i}, I)$
X_0^j
X_t
X_T

Figure 2: *Overview of our accent conversion model.*

target accent and speaker, while in the second approach, this information is provided only to the decoder.

4.2. Accent conversion

On top of the text-to-speech synthesis model with accent and speaker modelling, we introduce a diffusion-based end-to-end accent conversion method that does not require any external speech features such as text or phonemes. In particular, we propose a two-step approach where in the first step, we use saliency maps extracted from externally trained classifier to remove accent-related features from the original input. Then, we propose to recreate removed features replacing them with a target accent with a diffusion decoder. The overview of this method is presented in Figure 2.

4.2.1. Accent features removal with saliency maps

In order to remove accent-related features from a mel-spectrogram, we propose a novel method based on the saliency map technique that allows to identify regions of interest that highly influence the decision of the classifier. To that end, we first prepare a convolutional classifier that we train directly on mel-spectrograms to distinguish between all available accents. Then, before converting a mel-spectrogram from one accent to another, we follow the method described in section 3.2 and calculate saliency values for each mel-spectrogram with respect to the source accent. We then take the first quantile and mask that information out. In other words, we take the top 25% most important regions according to the classifier and replace them with silence. In the baseline scenario, we remove features by replacing them with zero values. However, from experiments, we learnt that this approach might lead to confusion between removed features and silence regions. Therefore, we also experimented with other approaches, such as replacing accent features with mean or random values.

4.2.2. Accent conversion

With source accent features removed from the source mel-spectrogram, we propose to train a diffusion model to convert examples from one accent to another. We once more refer to the procedure described in Section 3. This time, however, we implicitly create a forward diffusion process between the same utterances but different accents. In particular, we train the model with pairs of parallel examples X^i and X^j, with the same utterance but two different accents i and j.

For i as a source accent and j as a target one, we first remove the accent features from the source mel-spectrogram $\hat{X}^i = X^i \times m_b$, where m_b is a binary mask calculated as $m_b = m(X^i, i) > \lambda$ with saliency map for input X wrt. accent i and threshold λ. Then, we create a forward diffusion process following equation Eq. 1 between $\hat{X}^i$ and gaussian noise $N(X^j, I)$ as a terminal distribution. This approach is similar to the one proposed in [6], where terminal distribution is shifted from Normal distribution by the output of the text encoder. In our case, however, we shift it towards an altered source mel-spectrogram.

The final accent conversion method consists of two parts. First, we take the input mel-spectrogram X^i with source accent i, we remove the source accent features with a saliency map technique using a trained accent classifier $\hat{X}^i = X^i \times m(X^i, i) > \lambda$. Second, we sample the random noise to the preprocessed mel-spectrogram $x_T^i \sim N(\hat{X}^i, I)$, which we process with a diffusion decoder while conditioning it on the target accent j. We additionally condition the model on the original preprocessed source mel-spectrogram $\hat{X}^i$[6]. Our experiments show that while accent is converted, the source mel-spectrogram conditioning is necessary since the model levereges it as a vague representation of phonemes.

5. Dataset

We follow the experiments schema introduced in [42], where authors model accents with the Flow model. Therefore, we use the same training data with 3173 speakers distributed across six different English accents: American, Australian, British, Canadian, Indian, and Welsh. The dataset ranges in terms of sound quality and number of utterances per speaker (from 100 to 20k) and per accent (from 18k to 300k).

For the training of the accent conversion method, we create an artificial dataset of parallel examples by generating them with our TTS model. To that end, we randomly select 10000 train samples from each accent that we synthesize with the original speaker embeddings and all six target accents. This gives us a training dataset of 10000 tuples of 6 utterances with the same input text but different accented speech.

6. Experiments

In our experiments we validate both of our approaches on all of the available accents comparing it to state-of-the-art approach. We first evaluate our text-to-speech method, followed by experiments on accent conversion. Finally, we offer brief ablation study of our accent conversion using different saliency masks. Synthesised samples are available on our website[1].

6.1. Experimental details

For all of our experiments we use input phonemes processed by a single front-end designed for British English. We train accent embeddings jointly with the encoder and the decoder, while for speaker embeddings we use externally trained GE2E [45] model. For saliency extraction, we use the convolutional classifier that achieves average accuracy of 73% distinguishing between the six different accents. We use perceptual evaluation with a MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) tests. Each test was performed with 100 samples evaluated by at least 24 native listeners from each locale. The listeners rated the audio clips between 0 and 100 on three axis the quality of generated speech, the quality of the accent and the speaker similarity. For that end we ask evaluators from six different locales to answer

[1] Anonymized due to the double-blind policy. Please check the html file attached in the supplementary material.

	Naturalness						Accent Similarity						Speaker Similarity					
Locale	US	CA	GB	AU	IN	WLS	US	CA	GB	AU	IN	WLS	US	CA	GB	AU	IN	WLS
Lower anchor	-	-	-	-	-	-	65.84	36.34	44.61	30.81	31.61	38.73	56.95	34.86	43.02	31.26	43.14	37.73
Upper anchor	63.82	69.74	63.35	72.48	59.88	63.40	66.98	65.94	65.09	63.55	57.20	62.31	67.06	72.47	61.28	68.75	62.61	61.20
Flow-TTS AC [42]	63.06	60.24	57.11	65.78	57.59	59.40	70.06	66.87	51.88	**66.88**	56.78	59.18	65.90	48.70	54.13	50.19	51.20	**54.38**
Flow-TTS VC [44]	63.90	58.71	56.50	66.87	56.91	59.00	69.43	**67.14**	51.42	66.38	57.08	59.33	**66.34**	48.78	55.81	50.53	**52.10**	53.52
Ours (input conditioning)	64.08	60.61	56.19	**68.97**	59.66	59.13	69.81	64.94	**66.71**	64.40	60.04	58.62	64.81	**49.74**	62.49	50.61	46.89	51.79
Ours (latent conditioning)	**64.60**	**61.35**	**57.37**	68.52	**61.09**	**60.32**	**70.58**	**66.86**	66.31	65.85	**62.66**	**59.72**	65.32	48.53	**64.69**	**51.32**	46.69	52.80

Table 1: *Results of the MUSHRA evaluation of our proposed TTS approach when compared to recent state-of-the-art in terms of naturalness, accent, and speaker similarity. Our approach significantly outperforms SOTA in majority of cases especially in accent quality where it reaches the quality of upper anchor.*

three questions:

- *Please rate the audio samples in terms of their naturalness.*
- *Please rate how well this audio sample resemble X (e.g. Indian / British / American / Australian / Welsh) accent?*
- *Please listen to the speaker in the reference sample first. Then rate how similar the speakers in each system sound compared to the reference speaker.*

As part of MUSHRA evaluation, the reference and the hidden upper anchor were recordings of the corresponding locale, while the lower anchor was a recording from a different locale or speaker. We performed a paired t-test with Holm-Bonferroni correction ($p \leq 0.05$) to ensure results are statistically significant (marked in bold).

6.2. Text-to-speech approach

In Table 1, we show the full comparison of our method with two Flow-TTS-based baselines [42, 44] in all 6 different locales. We present MUSHRA scores averaged over 100 listeners. In order to spot cheating listeners, we added to the evaluation artificial samples containing white noise. Listeners that provided high scores for those samples were fully excluded from the score. In all of the evaluations our method performs similarly or better than Flow-TTS, while for 5 evaluations the difference between those two approaches is statistically significant. The highest gain in terms of performance is observed for accent quality, where for several locales (US, IN, CA) our approach reaches the quality of upper anchor (randomly selected native speaker from a given locale). The proposed method performs similarly to the baselines on speaker similarity evaluations. We can observe that latent conditioning performs better than the input one.

6.3. Accent conversion

Next, we evaluated the performance of our method applied to the task of accent conversion. This is much harder task, since by definition we do not have access to the phonemes at hand. We forced the model to convert from a source accent to a target accent in the backward and forward diffusion respectively. By applying saliency to the *important* regions of the mel-spectrogram we were able to convert the accent. Figure 3 presents the results of both using the text encoder and not using phonemes at all - both with and without masked saliency. While the big degradation without using phonemes is expected we see that masking using saliency significantly improves the results.

6.4. Ablation studies

For ablation studies, we run an experiment to validate the performance of accent removal with saliency maps. For that end, randomly select 100 training examples data in all 6 locales. Then, we follow procedure described in Section 4.2.2 to remove their original accent, replacing features described by saliency map as the most important ones with silence. With this alteration in place, we ask evaluators from all 6 locales to guess what was the original accent of the sample. As presented in 2 where we report the accuracy of human evaluators, we observe that they are easily confused and score very low in guessing the right accent.

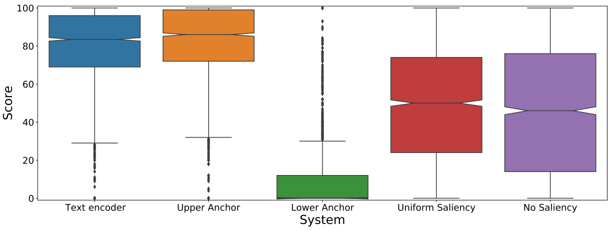


Figure 3: *Comparison of our accent conversion method for accent similarity.*

Accent	GB	US	CA	AU	IN	WLS
Original	48%	34%	23%	25%	30%	21%
Removed accent	43%	28%	21%	18%	26%	18%

Table 2: *Accuracy of naive listeners asked to classify the accent of samples with accent features removed through our saliency-based approach.*

7. Discussion & Inclusiveness

In this work, we introduce a method that models accented speech in a problem of speech synthesis. In our opinion it is an important goal, to expand the recent TTS system beyond the most common and most represented locales. The ability of TTS systems to recognise and synthesise linguistic nuances such as accents and dialects not only promotes inclusion and understanding, but also helps to preserve linguistic diversity. This is particularly important in an increasingly globalised world where the use of technology has the potential to homogenise linguistic expression and undermine linguistic diversity.

Although, our method presented in this work is limited to different variants of English, we believe that it can be easily adapted to numerous accents and dialects in different languages.

8. Conclusions

In this work, we propose a novel method for accented speech modelling based on diffusion generative models. We first propose an accented TTS system and show that our approach outperforms recent state-of-the-art, achieving upper bound performance in accent modelling for several locales. On top of this approach we introduce a method for accent conversion, where we remove source accent features using the saliency maps, in order to replace them with a target features using a diffusion model without the need to explicitly condition on phonemes.

9. References

[1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *ICML*. PMLR, 2015, pp. 2256–2265.

[2] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," *NeurIPS*, vol. 34, 2021.

[3] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation," *JMLR*, vol. 23, no. 47, pp. 1–33, 2022.

[4] D. P. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," in *NeurIPS*, 2021.

[5] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto, "Diffusion-lm improves controllable text generation," *arXiv preprint arXiv: Arxiv-2205.14217*, 2022.

[6] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-TTS: A diffusion probabilistic model for text-to-speech," in *ICML*. PMLR, 2021, pp. 8599–8608.

[7] M. Huckvale and K. Yanagisawa, "Spoken language conversion with accent morphing," 2007.

[8] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Foreign accent conversion in computer assisted pronunciation training," *Speech Comm.*, 2009.

[9] S. Aryal, D. Felps, and R. Gutierrez-Osuna, "Foreign accent conversion through voice morphing." in *Interspeech*, 2013.

[10] S. Aryal and R. Gutierrez-Osuna, "Can voice conversion be used to reduce non-native accents?" in *ICASSP*. IEEE, 2014.

[11] S. Liu, D. Wang, Y. Cao, L. Sun, X. Wu, S. Kang, Z. Wu, X. Liu, D. Su, D. Yu *et al.*, "End-to-end accent conversion without using native utterances," in *ICASSP*. IEEE, 2020.

[12] Z. Wang, W. Ge, X. Wang, S. Yang, W. Gan, H. Chen, H. Li, L. Xie, and X. Li, "Accent and speaker disentanglement in many-to-many voice conversion," in *ISCSLP*. IEEE, 2021, pp. 1–5.

[13] G. Zhao, S. Sonsaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent conversion using phonetic posteriorgrams," in *ICASSP*. IEEE, 2018.

[14] G. Zhao, S. Ding, and R. Gutierrez-Osuna, "Foreign accent conversion by synthesizing speech from phonetic posteriorgrams," in *Interspeech*, 2019.

[15] G. Zhao and R. Gutierrez-Osuna, "Using phonetic posteriorgram based frame pairing for segmental accent conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.

[16] S. Ding, G. Zhao, and R. Gutierrez-Osuna, "Accentron: Foreign accent conversion to arbitrary non-native speakers using zero-shot learning," *Computer Speech & Language*, vol. 72, p. 101302, 2022.

[17] S. Ding, C. Liberatore, S. Sonsaat, I. Lučić, A. Silpachai, G. Zhao, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "Golden speaker builder–an interactive tool for pronunciation training," *Speech Comm.*, vol. 115, pp. 51–66, 2019.

[18] C. Liberatore and R. Gutierrez-Osuna, "An exemplar selection algorithm for native-nonnative voice conversion." in *Interspeech*, 2021, pp. 841–845.

[19] G. Zhao, S. Ding, and R. Gutierrez-Osuna, "Foreign Accent Conversion by Synthesizing Speech from Phonetic Posteriorgrams," in *Proc. Interspeech 2019*, 2019, pp. 2843–2847.

[20] S. Ding, G. Zhao, and R. Gutierrez-Osuna, "Accentron: Foreign accent conversion to arbitrary non-native speakers using zero-shot learning," *Computer Speech and Language*, vol. 72, p. 101302, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230821001029

[21] C. Miao, S. Liang, M. Chen, J. Ma, S. Wang, and J. Xiao, "Flow-TTS: A non-autoregressive network for text to speech based on flow," in *ICASSP*. IEEE, 2020.

[22] H. Kim, S. Kim, and S. Yoon, "Guided-tts: A diffusion model for text-to-speech via classifier guidance," *arXiv preprint arXiv: Arxiv-2111.11755*, 2021.

[23] S. Kim, H. Kim, and S. Yoon, "Guided-TTS 2: A diffusion model for high-quality adaptive text-to-speech with untranscribed data," *arXiv preprint arXiv:2205.15370*, 2022.

[24] A. Levkovitch, E. Nachmani, and L. Wolf, "Zero-Shot Voice Conditioning for Denoising Diffusion TTS Models," in *Proc. Interspeech 2022*, 2022, pp. 2983–2987.

[25] T. Sadekova, V. Gogoryan, I. Vovk, V. Popov, M. Kudinov, and J. Wei, "A Unified System for Voice Cloning and Voice Conversion through Diffusion Probabilistic Modeling," in *Proc. Interspeech 2022*, 2022, pp. 3003–3007.

[26] Y. Koizumi, H. Zen, K. Yatabe, N. Chen, and M. Bacchiani, "Specgrad: Diffusion probabilistic model based neural vocoder with adaptive noise spectral shaping," *INTERSPEECH*, 2022.

[27] O. Avrahami, D. Lischinski, and O. Fried, "Blended diffusion for text-driven editing of natural images," in *CVPR*, June 2022.

[28] K. Song, L. Han, B. Liu, D. Metaxas, and A. Elgammal, "Diffusion guided domain adaptation of image generators," *arXiv preprint https://arxiv.org/abs/2212.04473*, 2022.

[29] M. Augustin, V. Boreiko, F. Croce, and M. Hein, "Diffusion visual counterfactual explanations," *arXiv preprint arXiv: Arxiv-2210.11841*, 2022.

[30] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *CVPR*, 2022, pp. 11 461–11 471.

[31] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[32] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *NeurIPS*, vol. 32, 2019.

[33] ——, "Improved techniques for training score-based generative models," in *NeurIPS*, 2020, pp. 12 438–12 448.

[34] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *ICLRW*, 2014.

[35] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017, pp. 618–626.

[36] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *NeurIPS*, vol. 31, 2018.

[37] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *JMLR*, vol. 11, pp. 1803–1831, 2010.

[38] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "Pointcloud saliency maps," in *ICCV*, 2019, pp. 1598–1606.

[39] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search," *NeurIPS*, vol. 33, pp. 8067–8077, 2020.

[40] Q. Yan, S. Vaseghi, D. Rentzos, and C.-H. Ho, "Analysis by synthesis of acoustic correlates of british, australian and american accents," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2004, pp. I–637.

[41] M. Huckvale and K. Yanagisawa, "Spoken language conversion with accent morphing," 2007.

[42] A. Ezzerg, T. Merritt, K. Yanagisawa, P. Bilinski, M. Proszewska, K. Pokora, R. Korzeniowski, R. Barra-Chicote, and D. Korzekwa, "Remap, warp and attend: Non-parallel many-to-many accent conversion with normalizing flows," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 984–990.

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[44] P. Bilinski, T. Merritt, A. Ezzerg, K. Pokora, S. Cygert, K. Yanagisawa, R. Barra-Chicote, and D. Korzekwa, "Creating New Voices using Normalizing Flows," in *Interspeech*, 2022, pp. 2958–2962.

[45] L. Wan, Q. Wang, A. Papir, and I. Lopez-Moreno, "Generalized end-to-end loss for speaker verification," *ICASSP*, 2017.