



SpeechPainter: Text-conditioned Speech Inpainting

Zalán Borsos, Matt Sharifi, Marco Tagliasacchi

Google Research

{zborsos, mns, mtagliasacchi}.google.com

Abstract

We propose SpeechPainter, a model for filling in gaps of up to one second in speech samples by leveraging an auxiliary textual input. We demonstrate that the model performs speech inpainting with the appropriate content, while maintaining speaker identity, prosody and recording environment conditions, and generalizing to unseen speakers. Our approach significantly outperforms baselines constructed using adaptive TTS, as judged by human raters in side-by-side preference and MOS tests.

Index Terms: speech inpainting, multimodality, adversarial learning

1. Introduction

Applications involving recognition, enhancement, and recovery of signals in one modality can often benefit from input in other modalities. In the context of speech processing, there have been several efforts to separate, enhance or recover target speech sources based on visual and other sensory modalities. On the other hand, recovering speech based on text has received little attention, despite its wide range of use-cases.

In this work, we investigate text-conditioned speech inpainting, named by analogy with image inpainting, where a region of unknown pixels is inferred from the surrounding observed pixels [1]. Our task is to infer a segment of missing speech up to one second long, given two inputs: i) the surrounding observed speech in an utterance, and ii) a transcript of the entire utterance, including both the missing and observed speech. In order to sound natural, the infilled audio should preserve speaker identity, prosody, recording environment conditions (reverb, background noise, bandpass filtering and other artifacts due to the recording microphone) and its speech content should correspond to the missing part according to the transcript. The approach should generalize to unseen speakers and content without any additional context provided besides the original short speech sample with the gap and the transcript.

Text-conditioned speech inpainting can be useful in several use-cases. For example, it could serve as a tool for users with speech impairments or language learners for fixing mispronounced words in recorded audio. In a similar manner, the model could be used for correcting grammar and word choice mistakes, as well as for restoring parts of speech affected by packet losses or excessive background noise.

The problem of unconditional speech inpainting has been addressed in [2, 3], with the main focus on packet loss concealment. These methods are only practical for short gap sizes (less than 250 ms). Relying on another modality for conditioning enables the inpainting of longer gaps. When conditioning on text, the speech inpainting approach of [4] can operate with gaps up to 750 ms. However, the method requires a significant amount of parallel training data for each target speaker, which is impractical to collect in many settings. Equipped with a

duration predictor, the approach of [5] allows for insertion, replacement and deletion of segments of multiple seconds. However, the model relies on conditioning on speaker identity, which restricts generalization to unseen speakers; the evaluation in [5] is performed only on speakers seen during training.

Recent advances in text-to-speech (TTS) systems [6, 7, 8] demonstrate the capability of TTS to synthesize speech while transferring speaker identity, prosody and some recording environment conditions from a short target speech sample. In fact, such approaches could already fulfill our requirements, provided that an additional module is designed for detecting and replacing the gap with the synthesized speech. We evaluate such a baseline in Section 3. Also relevant, but not directly related to our work, are methods that enhance speech based on modalities other than text: using facial crops of the target speaker, denoising [9], inpainting speech [10] or separating a target speaker from a mixture [11, 12], and enhancing speech based on accelerometer data [13], just to name a few.

In this paper, we propose *SpeechPainter*, a model capable of handling our challenging setup of text-conditioned speech inpainting. We demonstrate that the model can synthesize the missing speech content based on an *unaligned* transcript, while maintaining speaker identity, prosody and recording environment conditions. We base our model on the Perceiver IO [14], a scalable architecture for multi-modal data that demonstrated excellent performance and versatility over a range of tasks and input / output modalities. Another essential component for generating artifact-free speech is adversarial training with feature matching losses [15], commonly used in high-fidelity speech synthesis [16, 17]. Our approach significantly outperforms baselines constructed using adaptive TTS, as measured by MOS of human raters. We encourage the reader to listen to the samples produced by our model for unseen speakers and content.¹

2. Method

Before presenting our model, we first describe the requirements that motivate our design choices. First, the model should operate in the learning setup described in Section 1, by filling in a gap of at most one second in a speech segment given the corresponding transcript. Second, in favor of simplicity and minimal effort for training data preparation, we assume that the transcript is not aligned with the corresponding speech sample. We allow for longer transcripts that have a subsequence covering the content of the speech sample with the gap. In this setup, the model should learn to discover the part of the transcript corresponding to the gap and synthesize its content accordingly. Training data preparation for this setup is trivial: take any speech dataset containing speech samples and transcripts (e.g., LibriTTS [18], VCTK [19]), randomly crop the speech samples to a few seconds, randomly create the gap and retain the whole original transcript. Third, for the sake of generality, we further require

¹<https://google-research.github.io/seanet/speechpainter/examples/>

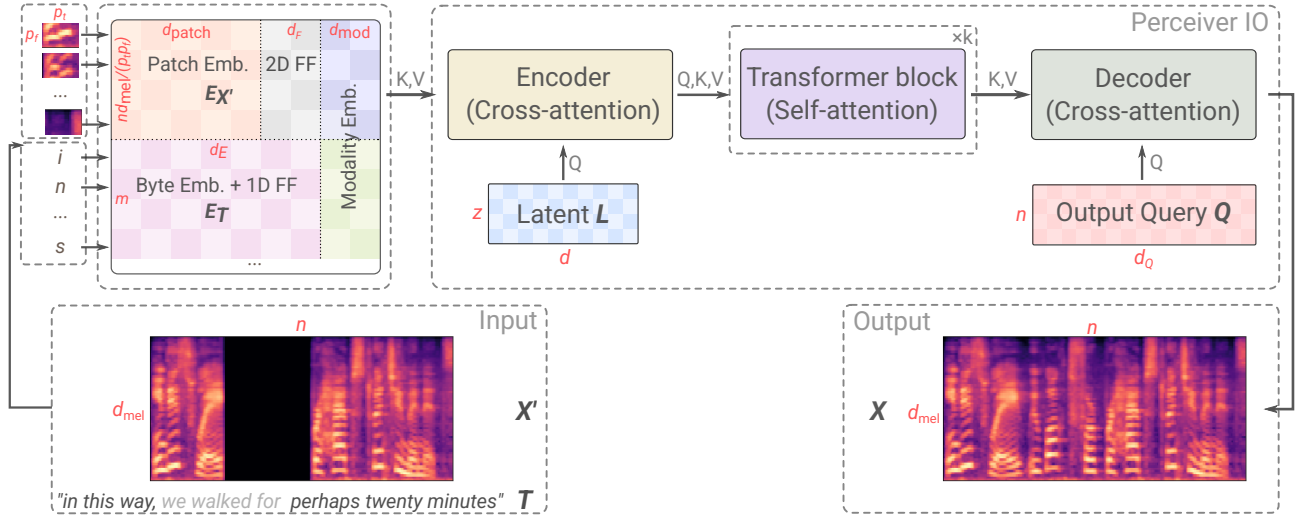


Figure 1: The architecture of SpeechPainter. The mel spectrogram of the audio input is split into patches, the text is split into characters. The concatenated embeddings are processed by Perceiver IO [14], which outputs inpainted mel spectrograms.

that the runtime and memory complexity of the model should scale well in terms of the length of the speech sample and transcripts, thus allowing the model to operate on longer sequences.

2.1. Model

We base our model architecture on Perceiver IO [14], which is capable of fulfilling the aforementioned requirements. In particular, Perceiver IO scales linearly in runtime and memory complexity with respect to the input and output sizes. This is achieved by processing the input using a cross-attention module with a learned latent query L to a fixed-size latent space, which is then processed by self-attending Transformer blocks [20]. Analogously, the output is constructed by cross-attending to the latent space using a learned output query Q . The choice of Perceiver IO as our base model, in contrast to autoregressive approaches, also allows us to use adversarial training (Section 2.2), which we found to be crucial for removing robotic artifacts from the inpainted sequence.

Instead of working with audio in the time domain, we choose the more compact log-scale mel spectrogram representation. This representation disregards phase information and emphasizes lower frequency details important for understanding speech, but also admits inversion algorithms capable of reproducing the audio in the time domain with high fidelity. In our experiments, we use a neural vocoder to perform the audio synthesis from log-mel spectrograms. The architecture of the vocoder is identical to MelGAN [16], but it is trained with a multi-scale reconstruction loss as well as adversarial losses from both wave- and STFT-based discriminators, following [21].

The architecture of SpeechPainter is shown in Figure 1. The input and output of the model are (X', T) and X , respectively, where $X \in \mathbb{R}^{n \times d_{\text{mel}}}$ is the n -frame, d_{mel} -band log-mel spectrogram of the target sample (without gap), T is the transcript, $X' := \text{mask}(X)$ and $\text{mask}(\cdot)$ is an augmentation that masks consecutive frames of the spectrogram, creating a gap of random length covering at most one second, starting at a random frame. This is achieved by setting the corresponding subsequence of the target signal in the time domain to zero.

The input keys l values to Perceiver IO’s encoder (the

first cross-attention) are constructed as follows. The log-mel spectrogram X' is split into non-overlapping patches of size $p_t \times p_f$, and the resulting patches are embedded to d_{patch} -dimensional representations through a learned linear layer after flattening. Then, the sequence of patches is concatenated with 2D Fourier positional encodings (FF, Fourier features) [20, 22] of dimensions d_{FF} , resulting in the spectrogram embedding $E_{X'} \in \mathbb{R}^{(n \cdot d_{\text{mel}} / (p_t \cdot p_f)) \times d_E}$, where $d_E = d_{\text{patch}} + d_{FF}$. For embedding the transcript, T is first padded with whitespaces to length m (the maximum number of characters accepted by the model) in order to support batching, and, following [14], it is UTF-8 byte-embedded with embedding dimension of d_E and is summed with Fourier positional encodings to obtain the text embedding $E_T \in \mathbb{R}^{m \times d_E}$. Finally, the learnable modality embeddings $e_X, e_T \in \mathbb{R}^{d_{\text{mod}}}$ are appended to $E_{X'}$ and E_T , respectively, the concatenation of which is the key l value to the encoder’s cross-attention.

The learned latent $L \in \mathbb{R}^{z \times d}$ serves as the query to the encoder, the output of which is attended to by k Transformer self-attention blocks. The task of the decoder is to produce the target log-mel spectrograms. To achieve this, the decoder’s cross-attention receives as query a learned output query $Q \in \mathbb{R}^{n \times d_Q}$ and as key l value the output of the final self-attention layer. Finally, the output of the decoder is projected along the last dimension to match the dimension of the target spectrogram.

2.2. Two-phase Training

We train the model in two consecutive phases in order to achieve both high signal reconstruction fidelity and high perceptual quality. In the first phase, we train the model with the L_1 reconstruction loss on log-mel spectrograms,

$$\mathcal{L}_{\mathcal{G}}^{\text{rec}} = \frac{1}{nd_{\text{mel}}} \mathbb{E}_{(X, T)} [\|X - \mathcal{G}(X', T)\|_1],$$

where \mathcal{G} denotes the model. With this choice of loss function, the model learns to correctly identify and inpaint the content of the gap based on the transcript, maintaining speaker identity, prosody and recording environment conditions. However, the results contain robotic artifacts (see ablation in the accompanying samples).

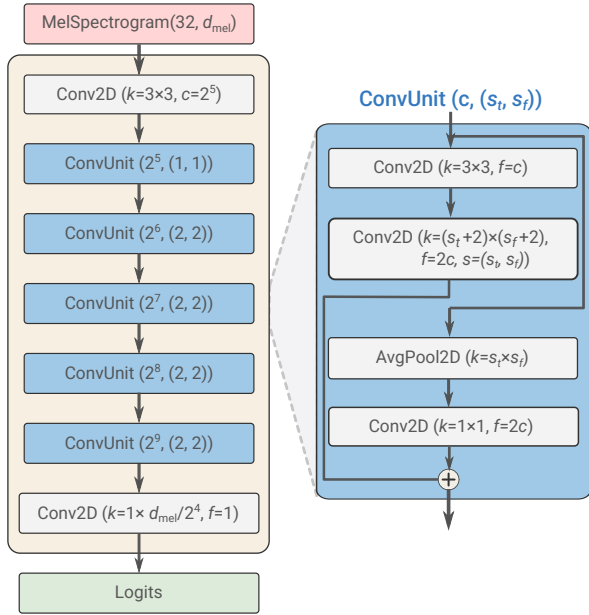


Figure 2: The architecture of the discriminator operating on log-mel spectrogram chunks. The convolution parameters k , f , s denote the kernel size, number of filters and strides, respectively. The first two convolutions in each block are preceded by ELU activation, and all convolution layers except the last are using layer normalization [23].

Therefore, we introduce a second training phase, where we perform adversarial training in order to achieve higher perceptual quality. We instantiate a discriminator \mathcal{D} with the objective to distinguish between synthesized and real mel spectrogram chunks of 32 frames. At the same time, we optimize \mathcal{G} to produce outputs that are indistinguishable from the ground truth by matching the feature representations in all layers of \mathcal{D} [15, 16]. \mathcal{D} is a convolutional network closely resembling the single-scale STFT discriminator of [21], shown in Figure 2 — for further details about the architecture, we refer the reader to [21]. The discriminator is trained with the hinge loss,

$$\mathcal{L}_{\mathcal{D}} = \mathbb{E}_{(X, T)} \left[\sum_t \max(0, 1 - \mathcal{D}_t(X)) \right] + \mathbb{E}_{(X, T)} \left[\sum_t \max(0, 1 + \mathcal{D}_t(\mathcal{G}(X', T))) \right],$$

where \mathcal{D}_t is the t -th logit of the discriminator along the time axis. The feature matching loss for \mathcal{G} is defined as

$$\mathcal{L}_{\mathcal{G}}^{\text{feat}} = \mathbb{E}_{(X, T)} \left[\frac{1}{\ell} \sum_{i=1}^{\ell} \frac{1}{d_i} \left\| \mathcal{D}^i(X) - \mathcal{D}^i(\mathcal{G}(X', T)) \right\|_1 \right]$$

where ℓ is the number of layers of \mathcal{D} , \mathcal{D}^i denotes the discriminator features at layer i and d_i is the dimension of \mathcal{D}^i . In the second phase of the training, \mathcal{G} is optimized using the loss $\mathcal{L}_{\mathcal{G}}^{\text{rec}} + \lambda_{\text{feat}} \mathcal{L}_{\mathcal{G}}^{\text{feat}}$ with $\lambda_{\text{feat}} = 10$.

3. Experiments

We train our model from scratch on LibriTTS at 24kHz, without relying on transfer learning or reusing pre-trained

embeddings. We choose the target audio length equal to three seconds, randomly cropped from the target utterance, which, in the case of LibriTTS, corresponds to an entire sentence. We feed the transcript of the whole target utterance to the model, and require the model to learn to determine the content of the gap based on the input audio containing the gap, thus minimizing the data preparation effort. Random amplitude scaling is used as the sole data augmentation. At inference time, the model is expected to generalize to unseen speakers, content and environmental recording conditions without fine-tuning.

We operate on log-scale mel spectrograms with hop length of 12.5 ms, frame length of 40 ms and $d_{\text{mel}} = 64$ channels, resulting in $n = 240$ dimensional log-mel spectrogram X along the time axis. We divide the spectrogram into non-overlapping patches of $p_t = 8$ by $p_f = 4$. The spectrogram patches are mapped to $d_{\text{patch}} = 768$ dimensions after flattening, and are concatenated with 2D Fourier positional encodings with $d_{FF} = 256$ to obtain $d_E = 1024$ -dimensional embeddings for each patch. We pad the text to $m = 500$ characters and use modality embeddings of $d_{\text{mod}} = 16$ dimensions. The learned latent dimensions are determined by $z = 512$ and $d = 1024$, while for the learned decoder query we have $n = 240$ and $d_Q = 512$. We use $k = 16$ self-attention layers, while both the cross- and self-attention layers have $h = 16$ heads. The query, key and value dimensions for all attention layers are set to $d/h = 64$. In the attention blocks, we use two-layer feed forward networks with hidden layer dimension of $d = 1024$. These parameter choices lead to a model with 112M parameters. In the first training phase, we train the model with Adam [24] with batch size of 256 for 3M steps, using learning rate of 10^{-4} decayed by a factor of 5 after every 800K steps. The second phase consists of 100K steps, where both the model and the discriminator are trained with Adam using a learning rate of $5 \cdot 10^{-5}$.

We evaluate our method for unseen speakers and content using samples from LibriTTS test-clean and test-other splits, as well samples from VCTK. We first trim the leading and trailing silence, and retain samples that are at least three seconds long after trimming. From these, we randomly choose 20 samples from each split (60 samples in total) from different speakers and randomly crop them to three seconds. We mask out a sequence of length between 750 ms and 1000 ms starting at a random location, while ensuring that the masked-out part has at least 300 ms left and right context.

As we are unaware of existing methods for text-conditioned speech inpainting suitable for our learning and inference setup described in Section 1, we design one based on adaptive TTS. For this, we synthesize the target transcript using a TTS system capable of transferring the speech style from the short target sample (three seconds, without masking), identify the part corresponding to the masked-out sequence, and perform the inpainting by inserting the identified part using a fade-in / fade-out technique. We rely on Meta-StyleSpeech [8] for adaptive TTS, using the pretrained model provided by the authors² and a vocoder matching Meta-StyleSpeech’s frontend trained in the same manner as SpeechPainter’s vocoder (Section 2.1).

We conduct side-by-side preference tests and Mean Opinion Score (MOS) tests with 6 human raters for assessing the naturalness of the inpainted samples. The raters were provided with the definition of a natural speech sample as a sample containing intelligible speech with no breaks in speaker identity, prosody and recording conditions. In the preference

²<https://github.com/KevinMIN95/StyleSpeech>, accessed on 21.11.2021

Table 1: Side-by-side preference and MOS tests (mean and standard error of the scores). In both tests, SpeechPainter outperforms adaptive TTS. According to aggregated MOS, human raters can detect the artifacts of inpainting with SpeechPainter.

	Preference	MOS		
	Speechpainter vs. TTS	TTS	SpeechPainter	Original (Target)
LibriTTS clean	0.72 ± 0.10	3.06 ± 0.10	3.60 ± 0.09	3.68 ± 0.09
LibriTTS other	0.55 ± 0.12	2.77 ± 0.12	3.12 ± 0.10	3.30 ± 0.12
VCTK	0.80 ± 0.10	3.38 ± 0.10	3.70 ± 0.09	3.98 ± 0.08
Aggregated	0.69 ± 0.06	3.07 ± 0.06	3.48 ± 0.06	3.66 ± 0.06

test, the raters are provided with the 60 pairs of samples corresponding to inpainting with TTS and SpeechPainter. The raters are asked to provide a score in $\{-2, -1, 0, 1, 2\}$, where a positive score indicates that the first sample is more natural than the second one. For MOS, we ask the raters for the naturalness on a five-point Likert scale (1-Bad, 2-Poor, 3-Fair, 4-Good, 5-Excellent) of the $60 \cdot 3$ samples corresponding to the target samples, inpainting via TTS and SpeechPainter, in a random order. The results in Table 1 show the scores over splits, as well as the aggregated scores. Although not designed for our use-case, the approach based on adaptive TTS represents a strong baseline according to the MOS tests. The superiority of SpeechPainter over adaptive TTS is expressed both in side-by-side preference test and in MOS scores on individual splits, as well as in the aggregated scores. According to aggregated MOS, human raters can also detect the artifacts of inpainting, and express a statistically significant preference of real samples over the ones inpainted with SpeechPainter.

Limitations. We experimented with real recordings in order to identify the limitations of the current model. We found that the model struggles in generalizing to recording environment conditions that significantly deviate from the ones encountered during training. The model fails to produce intelligible speech for highly reverberant, noisy samples, or when the recording microphone introduces noticeable distortions. Similarly, the model fails to generalize to unseen accents and speech tempos. There is also a degradation in speaker identity preservation for longer gap sizes, as well as a tendency towards producing a monotone prosody. We demonstrate the failure cases in our accompanying samples.

4. Conclusion

We formulated speech inpainting in the setup where only a short speech sample containing a gap and its corresponding transcript are available. We presented SpeechPainter, a model capable of synthesizing the missing speech content based on its unaligned transcript. We demonstrated that the model is capable of generalizing to unseen speakers and content, while maintaining speaker identity, prosody and recording environment conditions.

5. Broader Impact

In Section 1, we discussed several socially beneficial use-cases of text-conditioned speech inpainting. The main assumption was that the target speaker corresponds to the user of the model and the speech inpainting process happens with the consent of the speaker. However, since the model generalizes to unseen speakers and content, it raises the potential of third-party abuse of altering content and possibly the message of the speech of an unknown speaker without their consent; this could be used to

spread misinformation or for phishing attacks. Therefore, crucial components that should accompany such models are protocols for ensuring the speaker’s consent and the disclosure of the performed modifications, as well as systems for detecting inpainted speech sequences.

From the perspective of fairness, the model reflects the biases present in its training data. The most obvious manifestation of such a bias when training on LibriTTS (where native EN-US speakers outnumber other groups) is that accents might not be preserved: for example, the accent in the inpainted sequence for a non-native speaker is likely to be altered to an EN-US accent. A possible mitigation of this issue is better representation of a wide range of accents in the training data.

6. Acknowledgements

We would like to thank John Hershey and Johnny Soraker for their insightful feedback on this work.

7. References

- [1] J. Hays and A. A. Efros, “Scene completion using millions of photographs,” *ACM Transactions on Graphics (ToG)*, vol. 26, no. 3, 2007.
- [2] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, “Audio Inpainting,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 922–932, 2012.
- [3] Y. Bahat, Y. Y. Schechner, and M. Elad, “Self-content-based audio inpainting,” *Signal Processing*, vol. 111, pp. 61–72, 2015.
- [4] P. Prablanc, A. Ozerov, N. Q. K. Duong, and P. Pérez, “Text-informed speech inpainting via voice conversion,” in *2016 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 878–882.
- [5] D. Tan, L. Deng, Y. T. Yeung, X. Jiang, X. Chen, and T. Lee, “Editspeech: A text based speech editing system using partial inference and bidirectional fusion,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 626–633.
- [6] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, “Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron,” in *International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 4693–4702.
- [7] W.-N. Hsu, Y. Zhang, R. Weiss, H. Zen, Y. Wu, Y. Cao, and Y. Wang, “Hierarchical Generative Modeling for Controllable Speech Synthesis,” in *International Conference on Learning Representations (ICLR)*, 2019. [Online]. Available: <https://openreview.net/forum?id=rygkk305YQ>
- [8] D. Min, D. B. Lee, E. Yang, and S. J. Hwang, “Meta-StyleSpeech: Multi-Speaker Adaptive Text-to-Speech Generation,” in *International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 7748–7759.

- [9] J. Hou, S. Wang, Y. Lai, Y. Tsao, H. Chang, and H. Wang, "Audio-Visual Speech Enhancement Using Multimodal Deep Convolutional Neural Networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 117–128, 2018.
- [10] G. Morrone, D. Michelsanti, Z.-H. Tan, and J. Jensen, "Audio-Visual Speech Inpainting with Deep Learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6653–6657.
- [11] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 37, 2018.
- [12] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking," in *Interspeech*, 2019, pp. 2728–2732.
- [13] D. Roblek, K. Misiunas, M. Tagliasacchi, and P. Li, "SEANet: A Multi-modal Speech Enhancement Network," in *Interspeech*, 2020.
- [14] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, O. Hénaff, M. M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver IO: A General Architecture for Structured Inputs & Outputs," 2021.
- [15] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International Conference on Machine Learning (ICML)*, vol. 48. PMLR, 20–22 Jun 2016, pp. 1558–1566. [Online]. Available: <https://proceedings.mlr.press/v48/larsen16.html>
- [16] K. Kumar, R. Kumar, T. de Boissiere, L. Gestein, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019. [Online]. Available: <https://proceedings.neurips.cc/paper/2019/file/6804c9bca0a615bdb9374d00a9fcb59-Paper.pdf>
- [17] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 17 022–17 033.
- [18] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech," in *Interspeech*, 2019, pp. 1526–1530. [Online]. Available: <https://doi.org/10.21437/Interspeech.2019-2441>
- [19] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92)," 2019.
- [20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
- [21] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "SoundStream: An End-to-End Neural Audio Codec," 2021.
- [22] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, "Perceiver: General Perception with Iterative Attention," in *International Conference on Machine Learning (ICML)*, vol. 139. PMLR, 2021, pp. 4651–4664. [Online]. Available: <https://proceedings.mlr.press/v139/jaegle21a.html>
- [23] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," 2016.
- [24] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, 2015.