Interspeech 2024
1-5 September 2024, Kos, Greece

Diffusion Synthesizer for Efficient Multilingual Speech to Speech Translation

Nameer Hirschkind, Xiao Yu, Mahesh Kumar Nandwana, Joseph Liu, Eloi DuBois, Dao Le, Nicolas Thiebaut, Colin Sinclair, Kyle Spence, Charles Shang, Zoe Abrams, Morgan McGuire

Roblox, United States
nhirschkind@roblox.com

Abstract

We introduce DiffuseST, a low-latency, direct speech-to-speech translation system capable of preserving the input speaker's voice zero-shot while translating from multiple source languages into English. We experiment with the synthesizer component of the architecture, comparing a Tacotron-based synthesizer to a novel diffusion-based synthesizer. We find the diffusion-based synthesizer to improve MOS and PESQ audio quality metrics by 23% each and speaker similarity by 5% while maintaining comparable BLEU scores. Despite having more than double the parameter count, the diffusion synthesizer has lower latency, allowing the entire model to run more than 5× faster than real-time.

Index Terms: direct speech-to-speech translation, zero-shot, voice cloning, style transfer, diffusion

1. Introduction

Speech-to-speech translation (S2ST) has the potential to transform the way we communicate with others who do not speak the same language. The simplest way to perform S2ST automatically is to chain automatic speech recognition (ASR) with text-to-text machine translation (MT) followed by text-to-speech synthesis (TTS), in what is called a "cascaded" system [1]. However, cascaded systems are slow and do not take full advantage of non-textual information imparted by the audio modality. For example, the tone of the input audio could help choose between several words with meanings close to "sorry" in the target language. In recent years, models that translate directly to speech in the target language and can be optimized end-to-end have outperformed cascaded systems [1, 2, 3, 4]. These works rely on intermediary representations of text in the target language, such as discrete acoustic units or phonemes [2, 3]. Systems like AudioPalm and VioLA use a single decoder-only transformer architecture [4, 5], while others like SeamlessM4T, UniTY, and Translatotron2 use separate encoder, decoder, and synthesizer components [1, 2, 3]. In this work, we study direct S2ST systems with separate encoder, decoder, and synthesizer modules.

To make translated communication more natural, much work has been done to enable S2ST systems to output speech in the same voice, emotion, and prosody as their input [2, 5, 6, 7, 8]. This is known as zero-shot voice cloning or speaker preservation. Some prior works add architectural components separately trained to capture speaker characteristics [6, 8]. Others attempt to capture these characteristics implicitly while training on pairs of utterances with the same speaking style and expression [2, 5, 7]. We take this approach but improve on prior works by pretraining the synthesizer on diverse voices to enable our model to learn speaker preservation with less data.

To make S2ST systems as close to real monolingual interaction as possible, it is critical to run at low latency in a streaming context (i.e. the system starts speaking before the input speaker is done talking). The rare S2ST systems addressing this challenge still operate with over 2.5 seconds of ending delay [6]. While we do not tackle streaming directly in this work, we believe an important step to reduce system delay is to reduce model latency via parameter-efficient, low-latency, direct S2ST.

Developments in direct S2ST research are heavily influenced by work on TTS systems [1, 2, 5]. Recently, diffusion models such as Voicebox and NaturalSpeech2 have been shown to work well for speech synthesis [9, 10]. Diffusion models are attractive because they can produce diverse audio, pretrain on unlabeled audio data, and run non-autoregressively [9, 10, 11, 12]. Despite their great potential, existing works have yet to use such audio diffusion models for S2ST.

We introduce DiffuseST, a direct S2ST system that translates from many languages into English with a novel diffusion-based synthesizer that can perform low-latency S2ST with speaker preservation given as few as 3 seconds of input audio. The main contributions of this work are as follows: 1) We are the first work we know of to use a diffusion synthesizer for S2ST. We show our synthesizer is capable of zero-shot speaker preservation using implicit extraction of speaker characteristics, improving audio quality metrics by 23% and speaker similarity by 5% over a baseline. 2) We show the feasibility of S2ST with a much smaller architecture than previous works, enabling 5× faster than real-time inference and facilitating future work on streaming. 3) We are one of the first S2ST works to rely only on public data while still training on over 1k hours of audio, making our work large-scale but more reproducible than prior research.

2. Related Work

In this section, we review prior S2ST and TTS research most relevant to our work. We draw the most architectural inspiration from Translatotron2 and Seamless, both direct, zero-shot voice cloning S2ST models [1, 2, 6]. Both train on artificially-generated speaker-preserving label audios generated by TTS systems, but Seamless also trains on automatically aligned utterances mined from a massive multilingual corpus using the SONAR algorithm [13]. While Seamless uses a separate expressivity module to capture speaker characteristics, Translatotron2 does so implicitly using an attention mechanism.

Other S2ST systems such as VioLA or AudioPaLM use LLM-style architectures to perform S2ST by predicting both speech and text tokens with a single autoregressive transformer [4, 5]. These models demonstrate good quality but tend to have high parameter counts and can only run auto-regressively, hurt-

10.21437/Interspeech.2024-616

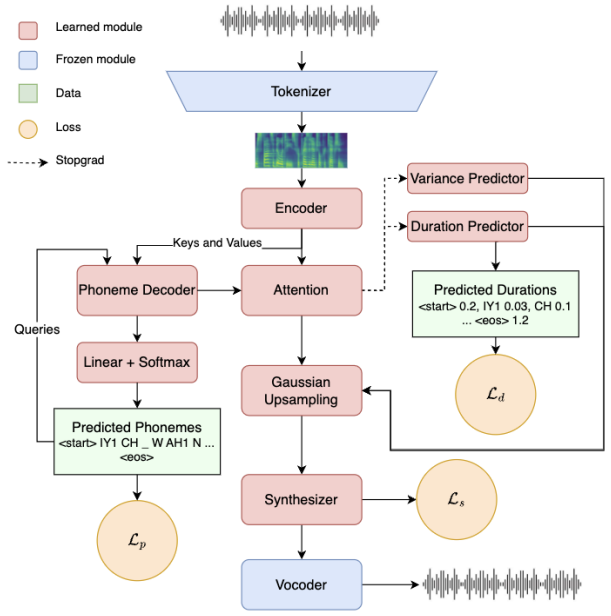

Figure 1: *Architecture Diagram of the proposed DiffuseST Model.*

ing latency.

Several recent works in TTS have used diffusion with great success. Voicebox is such a model that leverages a flow-matching objective to learn to generate Mel spectrograms given context audio and text [9, 14]. Recent work in using auto-encoders to learn discrete audio representations (e.g. EnCodec and SoundStream) enables performing audio diffusion in latent space [15, 16]. NaturalSpeech2 is such a diffusion-style TTS model that uses continuous embeddings of discrete acoustic tokens as targets, rather than Mel spectrograms [10]. Waveforms are then generated based on these latent features using the decoder of the same auto-encoder as a vocoder (module that produces waveforms given representations). SpeechFlow is a foundational model for speech generation tasks that uses Voicebox's flow-matching objective to generate EnCodec tokens [11]. After pretraining on a large corpus of unlabeled audio data, SpeechFlow can be fine-tuned to perform downstream tasks like audio quality enhancement or TTS.

3. System Description

In this section, we characterize our novel S2ST system, DiffuseST, delving into model architecture, training, and inference strategies. We give particular attention to the diffusion synthesizer as well as other parts of our architecture that differ significantly from prior work.

3.1. Model Architecture

As shown in Figure 1, DiffuseST comprises a tokenizer, acoustic encoder, phoneme decoder, upsampling and duration prediction module, synthesizer, and vocoder [1, 2]. To facilitate real-time use-cases later on, we target small parameter counts for all components.

The tokenizer is a waveform to Mel Spectrogram converter that runs on GPU. For the acoustic encoder, we use the Whisper-Small encoder due to its high quality pretraining [17]. For the phoneme decoder, we use a transformer decoder with rotary embeddings and a softmax head to predict phonemes in the target language given context from the encoder [18, 19]. We apply a cross-entropy loss on the phoneme predictions $\mathcal{L}_p$. Inspired by Translatotron2, we add a single multi-headed cross-attention layer—with no masking or autoregression—that uses the decoder hidden states as queries and the encoder outputs as keys and values [2]. This gives the model a second chance to extract features from the input audio that may be useful for voice cloning in the synthesizer.

We use two small transformer encoders with output dimension 1 and softplus activation to predict the durations and variances of each phoneme given the outputs of the attention layer. We supervise the duration predictor directly with ground truth, per-phoneme durations via an L_2 loss $\mathcal{L}_d$. Variances are supervised implicitly by the final loss term from the synthesizer. We stop gradients before both the duration and variance predictors to improve training stability. We then use the Gaussian upsampling method from Non-Attentive Tacotron (NAT) to upsample the attention layer outputs to the frequency required by the synthesizer [20]. At train time we use ground truth durations for upsampling, but we always use the predicted variances, allowing the variance predictor to receive gradients from the synthesizer.

The upsampled representations and durations contain all the information necessary to speak the predicted phonemes in the same voice as the input. This makes our architecture highly adaptable in that different synthesizers can be swapped out on top of the upsampling layer. We experimented with several synthesizer options and found a diffusion-based synthesizer to yield the best audio quality and speaker preservation. However, as the diffusion synthesizer is too unstable to learn unless the rest of the S2ST network weights are frozen, we first train all network parameters with a NAT-based synthesizer like Translatotron2 before replacing it with the diffusion synthesizer [2, 20], freezing all other parameters, and fine-tuning the synthesizer at the end of the training process. We refer to the loss term used to train the synthesizer generically as $\mathcal{L}_s$.

3.2. Diffusion Synthesizer

We will now go into greater depth on our novel diffusion synthesizer (see Figure 2). We base our work on an open source implementation of Voicebox [9].[1] The aspect we are most interested in using from Voicebox is the conditional flow matching diffusion objective [14], as it is reported to require fewer diffu-

[1] Credit to GitHub user lucidrains for the implementation

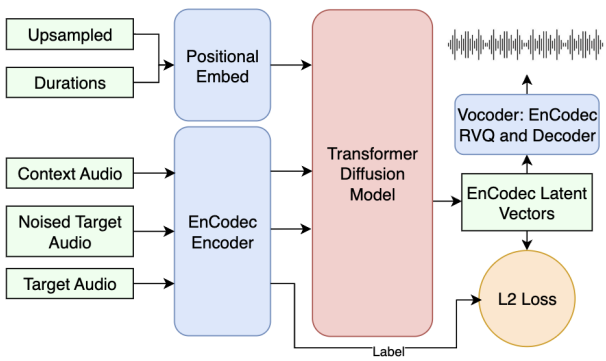

Figure 2: *Architecture of the Diffusion Synthesizer in the DiffuseST model.*

sion steps at inference time compared to other diffusion objectives [9]. We do not use Voicebox's duration predictor or text conditioning.

Similar to SpeechFlow, we first pretrain the diffusion model on unlabeled audio data [11]. We hypothesize this pretraining teaches the synthesizer how to speak in diverse voices, enabling it to learn the voice cloning task from minimal parallel S2ST data. For pretraining, we provide only masked context audio as conditioning. This makes our training task very similar to that of masked language modeling (MLM). We opt to have the diffusion synthesizer predict continuous EnCodec latent vectors [15], and then we use the EnCodec residual vector quantizer and decoder modules as a vocoder. At train time, we do not use the vocoder and compute an L_2 loss directly on the predicted EnCodec latents.

One of our main contributions is the mechanism by which we connect the diffusion synthesizer to the rest of the S2ST architecture without losing the benefits of pretraining, but still enabling zero-shot speaker preservation. We found a simple strategy to work well: we pass the upsampler's output through a small multilayer perceptron network, apply a positional encoding derived from the phoneme durations, and then add it to the context audio as conditioning. For the non-streaming S2ST task, context audio is just zeroes, as we are predicting an entire utterance from scratch. We note that the MLM-style training of the diffusion model facilitates further work on streaming S2ST, where we may want to continue speaking an already partially-spoken audio [6]. We train the synthesizer to make use of its new conditioning by training the entire DiffuseST network on parallel S2ST data but freezing all non-synthesizer parameters.

4. Experiments

We evaluate DiffuseST on the task of zero-shot voice cloning S2ST from multiple languages into English. We evaluate the quality of the translation, the quality of the output audio, the similarity between the input and output speakers, and the model latency with both the NAT and diffusion synthesizers.

4.1. Data

To improve reproducibility, we are one of the first large-data-scale works in S2ST to train entirely on public data. We pretrain on several public audio corpora to circumvent the lack of parallel S2ST data. We list all the datasets we use below:

- **The People's Speech**: We use the clean and dirty train segments of The People's Speech dataset [21], containing 52,000 hours of primarily English speech.
- **SpeechMatrix**: We take the en-en, es-en, fr-en, and de-en splits of the SpeechMatrix dataset [22], containing 14569 hours of source audio. We transcribe all English audios with Whisper Medium [17].
- **LibriSpeech Multilingual**: We use an MT model to translate the es, fr, and de splits of the LibriSpeech Multilingual dataset (3959 hours of audio) into English [23].
- **CVSS-T**: We use the high resource languages from the CVSS-T dataset (es-en, fr-en, de-en, and ca-en splits) [24], comprising 485 hours of source audio. CVSS-T has artificially synthesized speech labels that use a voice cloning TTS algorithm to preserve the voice of the input speaker in the output. We use the Montreal Forced Aligner to obtain duration labels for the target (English) phonemes [25].

Table 1: *Parameter counts of network components.*

Module	Parameter Count
Acoustic Encoder	88.2M
Phoneme Decoder	14.6M
Attention	1.0M
NAT Synthesizer +Duration and Upsampling	44.6M
Diffusion Synthesizer +Duration and Upsampling	102.1M

4.2. Training Protocol

We train our model in 4 stages. We focus on translating from Spanish, French, and German into English due to those languages' data availability. Our parameter counts are shown in Table 1. We use an AdamW optimizer with a reduce-on-plateau learning rate schedule [26]. When training on the smaller LibriSpeech and CVSS-T datasets, we apply SpecAugment and dropout to prevent overfitting [27].

Our training curriculum proceeds as follows: 1) Pretrain the diffusion synthesizer on unlabeled audio from The People's Speech. This takes about 7 days using 8 A100 GPUs. 2) Pretrain the encoder and decoder on a speech-to-text translation (S2TT) task with SpeechMatrix using loss $\mathcal{L} = \mathcal{L}_p$. This also takes 7 days on 8 A100 GPUs. 3) Train the entire model with the NAT synthesizer on both S2TT and S2ST tasks simultaneously on a mixture of the LibriSpeech and CVSS-T datasets. We find that this multi-task training helps prevent overfitting on the smaller CVSS dataset and enhances translation quality. This step uses all losses: $\mathcal{L} = \mathcal{L}_p + \mathcal{L}_d + \mathcal{L}_s$ and takes 2-3 days on 8 A100 GPUs. 4) Replace the NAT synthesizer with the diffusion synthesizer and freeze all parameters other than the synthesizer. Finetune the model on just the CVSS-T dataset with loss $\mathcal{L} = \mathcal{L}_d + \mathcal{L}_s$. This step takes about 3 days on 8 A100 GPUs.

We evaluate both the NAT synthesizer model produced by step 3 and the diffusion synthesizer model produced by step 4. Note that the encoder, decoder, attention, and duration modules of both models are the same because they are frozen in step 4.

4.3. Translation Quality Results

Table 2: *Translation metrics computed on ASR transcriptions of model output on the high resource languages of CVSS-T. For BLEU and ChrF, higher is better. For WER and TER, lower is better. According to [24], references were generated with a PnGNAT voice-cloning TTS system using ground truth text translations.*

Source Language	Method	Translation Metrics			
		BLEU	WER	ChrF	TER
French	NAT	23.7	0.61	49.3	59.7
	Diffusion	23.4	0.61	48.8	60.0
German	NAT	19.7	0.69	44.3	66.0
	Diffusion	19.4	0.69	44.0	66.4
Spanish	NAT	23.9	0.63	50.9	60.7
	Diffusion	23.1	0.65	50.3	62.8
Catalan	NAT	20.3	0.69	47.0	66.9
	Diffusion	20.1	0.69	46.9	67.1
Overall	NAT	21.9	0.65	47.9	63.3
	Diffusion	21.7	0.66	47.5	64.0
	Reference	90.8	0.06	95.5	5.5

We evaluate DiffuseST on the test split of CVSS-T. Note that all speakers in the test set are unseen in the train set [24]. In the decoder, we use a beam search with a size of 5. For the diffusion synthesizer model, we use 25 diffusion steps and a classifier-free guidance scale of 1.0 at inference time. For the NAT synthesizer model, we use a HiFiGAN vocoder on the output to convert mel spectrograms to waveforms [28]. To assess translation quality, we transcribe DiffuseST's output with Whisper-Large-V2 and compute several common translation evaluation metrics on the transcript by comparing to the text label in CVSS-T [17]. Results are shown in Table 2. Since the encoder and decoder are the same for the NAT and diffusion synthesizer models, all differences in translation quality are attributable to pronunciation.

We observe that the diffusion synthesizer lags in this area by 0.25 BLEU points across all languages, but this difference is not statistically significant (p = .35) and it potentially could be made up with further training on a larger dataset or hyperparameter tuning. Regardless, we believe the dramatic gains in audio quality more than make up for the small tradeoff in pronunciation.

As S2ST models in other works that are evaluated on CVSS use encoders in excess of 600M parameters, we lack direct comparisons in literature [1, 6, 29]. Unsurprisingly, we do not outperform these much larger works. For instance, SeamlessM4T-Large and AudioPalm achieve 36.5 and 32.5 BLEU respectively across all languages of CVSS-C. In future work, we plan on training versions of DiffuseST with higher parameter counts to create fair comparisons.

4.4. Audio Quality and Speaker Similarity Results

Next, we evaluate the audio quality of DiffuseST with both the NAT and diffusion synthesizers using the Torch Squim models (results in Table 3) [30]. We find the diffusion synthesizer to outperform the NAT synthesizer in both MOS and PESQ scores by about 23%. We believe that due to the diffusion model's extensive pretraining, it is better able to emulate the sounds of speech without producing excess artifacts or noise. Given the magnitude of the difference in MOS and PESQ scores, we think the diffusion synthesizer's gains in audio quality more than offset its issues with pronunciation, warranting further study of diffusion synthesizers for S2ST.

We also evaluate DiffuseST's voice cloning ability. We measure the cosine similarity between speaker embeddings of the input speech and model output. We use the style encoder from StyleTTS2 [31] to compute embeddings because of StyleTTS2's impressive voice cloning abilities. We observe that the diffusion synthesizer outperforms the NAT synthesizer with a 4.6% improvement in cosine similarity across all languages. Again, we believe that the diffusion synthesizer's pretraining on diverse voices allows it to learn the complex voice-cloning task from little data. We believe this could be further improved through monolingual voice cloning training, generating more parallel S2ST data with a higher quality TTS algorithm, and potentially even training with backtranslation like Translatotron3 [7]. We plan on exploring these methods in future work.

Table 3: *Mean audio quality metrics computed on model output using Torch Squim models. For both MOS and PESQ, higher is better. Dataset is high-resource languages of CVSS-T.*

Source Language	Synth	Audio Quality Metrics	
		MOS	PESQ
French	NAT	3.21	1.81
	Diffusion	**3.98**	**2.28**
German	NAT	3.20	1.83
	Diffusion	**3.98**	**2.29**
Spanish	NAT	3.39	1.85
	Diffusion	**3.91**	**2.26**
Catalan	NAT	3.12	1.93
	Diffusion	**3.99**	**2.24**
Overall	NAT	3.23	1.85
	Diffusion	**3.97**	**2.27**
	Reference	4.02	3.11

Table 4: *Speaker similarity scores between input speech and generated speech as measured by cosine similarity between embeddings computed with the StyleTTS2 [31] style embedding network. Higher is better.*

Method	Source Language				
	French	German	Spanish	Catalan	All
NAT	.61	.62	.61	.57	.60
Diffusion	**.64**	**.64**	**.62**	**.63**	**.63**
Reference	.67	.67	.66	.66	.67

4.5. Performance Results

Lastly, we evaluate the speed of our model at inference time with both the NAT and diffusion synthesizers. We measure the mean inference time using a subset of data from the CVSS-T test set, running on a single A100 GPU using fp16 precision. We use flash attention [32] in the whisper encoder and diffusion synthesizer. The input audios used for performance evaluation have a mean duration of 5.61 seconds. We find the mean inference time of the model with the diffusion synthesizer to be 0.99 seconds while the NAT model takes 1.04 seconds, making both models over 5× faster than real-time. Furthermore, despite having more than 2× more parameters, the diffusion synthesizer is faster than NAT one. In both cases, the most time-intensive step of inference is the decoder's beam search because it requires doing many forward passes, each of which must attend to the entire high-dimensional encoder output. We believe this low latency sets us up to adapt DiffuseST to work in a streaming setting.

5. Conclusion

We propose DiffuseST, a multilingual S2ST model featuring a non-autoregressive diffusion synthesizer capable of zero-shot voice cloning. In our experiments, we show our diffusion synthesizer improves audio quality and speaker preservation with minimal costs to pronunciation. We are one of the first large-data-scale S2ST works to train exclusively on public datasets, paving the way for more democratized S2ST work. DiffuseST is also one of the first direct, zero-shot speaker-preserving S2ST systems with fewer than 600M parameters, enabling over 5× faster than real-time processing. In the future, we intend to make DiffuseST work in a streaming setting by further optimizing the model latency while adding predictive capabilities to the phoneme decoder. We plan to continue to improve translation and audio quality via artificial dataset generation and backtranslation.

6. References

[1] L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, P.-A. Duquenne, H. Elsahar, H. Gong, K. Heffernan, J. Hoffman *et al.*, "Seamlessm4t: Massively multilingual and multimodal machine translation," 2023.

[2] Y. Jia, M. T. Ramanovich, T. Remez, and R. Pomerantz, "Translatotron 2: Robust direct speech-to-speech translation," *CoRR*, vol. abs/2107.08661, 2021. [Online]. Available: https://arxiv.org/abs/2107.08661

[3] H. Inaguma, S. Popuri, I. Kulikov, P.-J. Chen, C. Wang, Y.-A. Chung, Y. Tang, A. Lee, S. Watanabe, and J. Pino, "Unity: Two-pass direct speech-to-speech translation with discrete units," *arXiv preprint arXiv:2212.08055*, 2022.

[4] P. K. Rubenstein, C. Asawaroengchai, D. D. Nguyen, A. Bapna, Z. Borsos, F. de Chaumont Quitry, P. Chen, D. E. Badawy, W. Han, E. Kharitonov *et al.*, "Audiopalm: A large language model that can speak and listen," 2023.

[5] T. Wang, L. Zhou, Z. Zhang, Y. Wu, S. Liu, Y. Gaur, Z. Chen, J. Li, and F. Wei, "Viola: Unified codec language models for speech recognition, synthesis, and translation," 2023.

[6] L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, M. Duppenthaler, P.-A. Duquenne, B. Ellis, H. Elsahar, J. Haaheim *et al.*, "Seamless: Multilingual expressive and streaming speech translation," 2023.

[7] E. Nachmani, A. Levkovitch, Y. Ding, C. Asawaroengchai, H. Zen, and M. T. Ramanovich, "Translatotron 3: Speech to speech translation with monolingual data," 2024.

[8] K. Song, Y. Ren, Y. Lei, C. Wang, K. Wei, L. Xie, X. Yin, and Z. Ma, "StyleS2ST: Zero-shot Style Transfer for Direct Speech-to-speech Translation," in *Proc. INTERSPEECH 2023*, 2023, pp. 42–46.

[9] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, J. Mahadeokar, and W.-N. Hsu, "Voicebox: Text-guided multilingual universal speech generation at scale," 2023.

[10] K. Shen, Z. Ju, X. Tan, Y. Liu, Y. Leng, L. He, T. Qin, S. Zhao, and J. Bian, "Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers," 2023.

[11] A. H. Liu, M. Le, A. Vyas, B. Shi, A. Tjandra, and W.-N. Hsu, "Generative pre-training for speech with flow matching," 2023.

[12] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, "Audioldm 2: Learning holistic audio generation with self-supervised pretraining," 2023.

[13] P.-A. Duquenne, H. Schwenk, and B. Sagot, "Sonar: Sentence-level multimodal and language-agnostic representations," 2023.

[14] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," 2023.

[15] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," 2022.

[16] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," 2021.

[17] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," 2022.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[19] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *CoRR*, vol. abs/2104.09864, 2021. [Online]. Available: https://arxiv.org/abs/2104.09864

[20] J. Shen, Y. Jia, M. Chrzanowski, Y. Zhang, I. Elias, H. Zen, and Y. Wu, "Non-attentive tacotron: Robust and controllable neural TTS synthesis including unsupervised duration modeling," *CoRR*, vol. abs/2010.04301, 2020. [Online]. Available: https://arxiv.org/abs/2010.04301

[21] D. Galvez, G. Diamos, J. Ciro, J. F. Cerón, K. Achorn, A. Gopi, D. Kanter, M. Lam, M. Mazumder, and V. J. Reddi, "The people's speech: A large-scale diverse english speech recognition dataset for commercial usage," 2021.

[22] P.-A. Duquenne, H. Gong, N. Dong, J. Du, A. Lee, V. Goswani, C. Wang, J. Pino, B. Sagot, and H. Schwenk, "Speechmatrix: A large-scale mined corpus of multilingual speech-to-speech translations," 2022.

[23] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," in *Interspeech 2020*, ser. interspeech_2020. ISCA, Oct. 2020. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2826

[24] Y. Jia, M. T. Ramanovich, Q. Wang, and H. Zen, "Cvss corpus and massively multilingual speech-to-speech translation," 2022.

[25] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi," in *Interspeech*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:12418404

[26] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017. [Online]. Available: http://arxiv.org/abs/1711.05101

[27] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*, ser. interspeech_2019. ISCA, Sep. 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[28] 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*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 17 022–17 033. [Online]. Available: https://proceedings.neurips.cc/paper{_}files/paper/2020/file/c5d736809766d46260d816d8dbc9eb44-Paper.pdf

[29] Y. Jia, Y. Ding, A. Bapna, C. Cherry, Y. Zhang, A. Conneau, and N. Morioka, "Leveraging unsupervised and weakly-supervised data to improve direct speech-to-speech translation," in *Proc. Interspeech 2022*, 2022, pp. 1721–1725.

[30] A. Kumar, K. Tan, Z. Ni, P. Manocha, X. Zhang, E. Henderson, and B. Xu, "Torchaudio-squim: Reference-less speech quality and intelligibility measures in torchaudio," 2023.

[31] Y. A. Li, C. Han, V. S. Raghavan, G. Mischler, and N. Mesgarani, "StyleTTS 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=m0RbqrUM26

[32] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16 344–16 359, 2022.