Active Learning for Text-to-Speech Synthesis with Informative Sample Collection

Kentaro Seki*†, Shinnosuke Takamichi*†, Takaaki Saeki* and Hiroshi Saruwatari*
* The University of Tokyo, Japan. † Keio University, Japan.
Email: seki-kentaro922@g.ecc.u-tokyo.ac.jp

***Abstract*—The construction of high-quality datasets is a cornerstone of modern text-to-speech (TTS) systems. However, the increasing scale of available data poses significant challenges, including storage constraints. To address these issues, we propose a TTS corpus construction method based on active learning. Unlike traditional feed-forward and model-agnostic corpus construction approaches, our method iteratively alternates between data collection and model training, thereby focusing on acquiring data that is more informative for model improvement. This approach enables the construction of a data-efficient corpus. Experimental results demonstrate that the corpus constructed using our method enables higher-quality speech synthesis than corpora of the same size.**

I. Introduction

Recent advances in text-to-speech (TTS) synthesis have been strongly driven by the development of machine learning techniques that can leverage large-scale datasets [1], [2], [3], making the construction of such datasets increasingly important. As the data requirements for TTS systems continue to grow, the mainstream approach to dataset construction is shifting from costly studio recordings [4], [5], [6] to more scalable collection-based methods [7], [8], [9]. These methods typically involve collecting data from large-scale resources such as automatic speech recognition (ASR) corpus, audiobooks, and web data, and processing it into a format suitable for TTS model training. Owing to the high degree of automation in data acquisition and preprocessing, they offer a significant advantage in terms of scalability.

Existing collection-based speech corpus construction approaches [10], [11], [12] have primarily focused on collecting large quantities of speech data. However, in practice, storage capacity constraint inevitably limit the amount of data that can be effectively utilized. This makes it essential to consider how to efficiently select data that maximizes learning effectiveness within limited dataset size. Nevertheless, many large-scale data collection strategies have paid insufficient attention to the optimization of data efficiency, leaving a critical aspect of practical TTS training unaddressed.

In connection with this issue, core-set selection methods have been proposed for TTS tasks [13]. Core-set selection aims to extract a representative subset from a large corpus that ideally achieves an equivalent learning effect as the original entire dataset [14]. Prior work [13] improves data efficiency by maximizing diversity in the feature space and eliminating redundant samples; however, these methods are model-agnostic and do not take the learning dynamics of the TTS model into account. In the field of image recognition, a previous study [15] have demonstrated that one of the strongest baselines for core-set selection is active learning, which iteratively refines both model training and dataset construction [16]. This suggests that incorporating active learning into core-set selection may enable the construction of more data-efficient TTS corpus.

In this study, we propose a data-efficient corpus construction method for TTS based on active learning. Specifically, we focus on web-scale speech data, which is both large in volume and rich in diversity, and introduce a framework that selectively and incrementally utilizes such data. Our method begins by listing and partitioning candidate web data sources, then sequentially collects each segment and trains a TTS model on it. Based on the performance evaluation of the trained model, our method determines which parts of the next segment should be incorporated. Furthermore, since the proposed approach downloads data segments on demand, it eliminates the need to download the entire dataset in advance, improving storage efficiency. This feedback loop enables the model to focus on informative samples, thereby facilitating efficient and targeted corpus construction. Experimental results show that the proposed method achieves superior speaker coverage compared to baseline methods, when trained on corpora of the same size.

II. Proposed Method

In this study, we focus on multi-speaker TTS models that are conditioned on x-vectors [17], with the goal of enabling speech generation for a more diverse set of speakers. Specifically, we define a speaker as being "synthesizable" if their synthetic speech exceeds a predefined quality threshold, and our method aims to increase the number of such synthesizable speakers.

A. Preparation

1) Creating a Video ID list: First, we create a list of video IDs D_{all} to sample from YouTube. This allows for fixing the distribution in subsequent random sampling, avoiding errors introduced by changes in distribution. Video IDs are considerably lightweight compared to text-audio pairs, making it feasible to collect IDs for a larger number of videos than the text-audio pair collection. In this study, we employ the method described in the previous study [9] to search for videos and create a list of video IDs. We then randomly shuffle the video ID list and divide it into K disjoint subsets $D_1, D_2, ..., D_K$,

979-8-3315-7206-8/25/$31.00

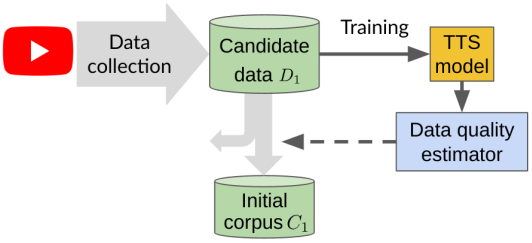
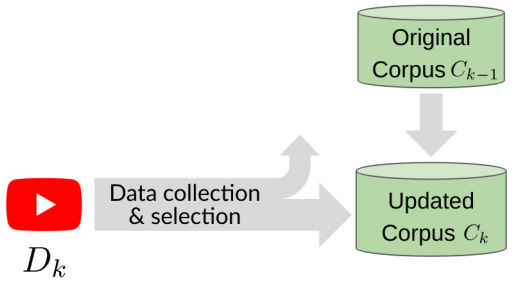
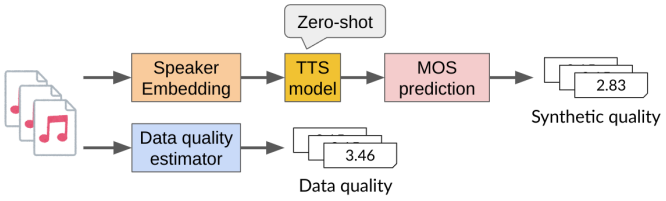


Fig. 1: Overall procedure of initial corpus construction.



(a) Additional data acquisition



(b) Data selection procedure

Fig. 2: Overall procedure of additional data acquisition.

each of which is used in a different iteration of data collection. The proportion of data used in the k-th iteration is denoted as $r_k := |D_k|/|D_{\text{all}}|$, which is a predefined parameter that determines the sampling ratio for each iteration.

2) Setting the target synthesis quality: As stated at the beginning of this section, our goal is to increase the number of speakers for whom speech can be successfully synthesized. To this end, we define a quality threshold that serves as the criterion for determining whether a speaker is considered "synthesizable." Specifically, we train a TTS model using an existing studio-recorded multi-speaker corpus and evaluate the synthesis quality for each speaker. We then define the minimum observed quality score among these speakers as the threshold θ_{hq}. Speakers whose synthetic speech exceeds θ_{hq} are regarded as achieving a synthesis quality comparable to that of the studio-recorded corpus, and thus are considered capable of generating high-quality speech.

B. Initial corpus construction

Fig. 1 shows the overall procedure of the initial corpus construction.

Following the previous study [9], we download audio-text pairs for the manually subtitled videos in L_1 and perform pre-screening based on two criteria: text-audio alignment accuracy and speaker compactness. Specifically, we compute the connectionist temporal classification (CTC) score [18] to evaluate how well the audio aligns with the corresponding text, and filter out utterances with low alignment scores. Additionally, we estimate the intra-video variance of x-vectors [17] to assess speaker consistency within each video, discarding groups with high variance. During the download process, we extract an x-vector for each video. These x-vectors are lightweight compared to audio or text data and are saved for later use during inference.

We further need to filter out low-quality data from the downloaded set. To this end, we employ the training-evaluation loop method introduced in our previous study [9], [19]. This method evaluates the quality of candidate training data by training a TTS model and then applying an automatic speech quality prediction model to the synthetic outputs. The evaluation model is trained to predict the synthesis quality of a TTS model that would be trained using the given data. Based on these predicted scores, we select utterances that are expected to exceed the threshold θ_{hq}, thereby constructing the initial corpus C_1.

C. Additional data acquisition

As illustrated in Fig. 2a, at the k-th step, we construct the corpus C_k by selecting necessary data from D_k and adding it to the existing corpus C_{k-1}.

First, we download audio-text data from D_k in the same manner as described in Section 2.B. Each data sample is then individually evaluated to determine whether it should be added to the corpus, based on two criteria: data quality and informativeness, as shown in Fig. 2b.

For data quality, we apply the data quality estimator trained in Section 2.B and select samples that exceed the threshold θ_{hq}.

To assess informativeness, we train TTS model with the original corpus C_{k-1}. Then, we extract the x-vector from each audio sample and perform zero-shot synthesis using this TTS model. If the resulting synthetic speech exceeds the quality threshold θ_{hq}, the sample is deemed redundant and excluded, as high-quality synthesis is already achievable without it.

In summary, we construct C_k by adding to C_{k-1} the samples from D_k that satisfy both of the following: (1) data quality exceeds θ_{hq}, and (2) synthetic quality falls below θ_{hq}.

D. Inference process

The TTS system used in this study requires an x-vector as part of its input. During the corpus construction, we performed a preprocessing step in which we extracted x-vectors from all videos in D_{all}. As a result, we obtained and stored a set of speaker embeddings $\boldsymbol{x}_1, \ldots, \boldsymbol{x}N \in \mathbb{R}^d$, where d is the dimensionality of each x-vector and N is the number of unique speakers in Dall. In the following, we describe how these precomputed x-vectors are used to infer the appropriate speaker identity at test time.

1) Real speaker: A straightforward strategy is to sample from the set $\{\boldsymbol{x}i\}\, i = 1^N$. Since the corpus construction process aims to enable high-quality synthesis (above θ_{hq}) for as many speakers in D_{all} as possible, we expect that using these x-vectors will result in synthetic speech that meets or exceeds the threshold.

In cases where the synthesized speech for a speaker $\boldsymbol{x}i$ falls below θ_{hq}, a plausible cause is that the corresponding training data were excluded due to its low quality. While improving the quality of such data with data cleansing could enhance synthesis performance, this study does not address that issue.

2) Generated speaker: Since synthesis using real speaker x-vectors is inherently limited to the N speakers observed in the dataset, we consider an alternative approach using generated x-vectors. This task is known as speaker generation (SG), and SG using Gaussian mixture models (GMMs) conditioned on speaker attributes has been previously proposed [20].

In this study, the x-vectors are obtained by crawling speech data from the web and are not accompanied by explicit attribute annotations. Therefore, we adopt a data-driven approach to model the underlying structure of speaker embeddings using a diffusion model [21]. Diffusion models have been shown to effectively capture complex data distributions both theoretically [22] and empirically [23], making them a suitable choice for modeling the diverse and intricate distribution of YouTube speaker data.

However, given that the number of available x-vectors ($N = 2719$ in our experiment) is small relative to their dimensionality ($d = 512$), we draw inspiration from latent diffusion models. Specifically, we apply principal component analysis (PCA) to decompose the embedding space and model only the principal components using a diffusion model, while approximating the remaining dimensions with a Gaussian distribution.

Let $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{R} \in \mathbb{R}^{d\times d}$ denote the empirical mean and covariance matrix of the x-vectors:

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i, \qquad \boldsymbol{R} = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T. \tag{1}$$

Let $\lambda_1, \ldots, \lambda_d$ be the eigenvalues of $\boldsymbol{R}$ in descending order, and let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ be the corresponding orthonormal eigenvectors. We define $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d]$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_d)$. Then, we perform the following coordinate transformation to obtain a decomposition into principal and residual components:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{E}^T(\boldsymbol{x} - \boldsymbol{\mu}), \tag{2}$$

where $\boldsymbol{y} \in \mathbb{R}^{d'}$ and $\boldsymbol{z} \in \mathbb{R}^{d-d'}$ for a chosen integer d'. We model $\boldsymbol{y}$ using a diffusion model and approximate $\boldsymbol{z}$ using a Gaussian distribution. By construction, $\boldsymbol{z}$ has zero mean and unit covariance, and is therefore modeled as $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

III. Experimental evaluation

A. Experimental conditions

1) Data: The data download and pre-screening procedures followed the same methodology as the prior study [9]. As a result, $2,719$ YouTube videos were processed, yielding approximately 66 hours of candidate speech data, comprising around $60,000$ Japanese utterances. Note that downloading all of them simultaneously was not required. To calculate pseudo MOS, we used 100 phonetically balanced sentences from the JVS corpus [5]. For the final evaluation of the trained TTS models, we used 324 test sentences from the ITA corpus [24].

2) Text-to-speech model: We employed FastSpeech 2 [25] as our multi-speaker TTS model and used the UNIVERSAL_V1 configuration of the pre-trained HiFi-GAN vocoder [26], [27]. The model architecture and hyperparameters were adopted from the open-source implementation [28], with the exception of the speaker representation. Instead of the original one-hot encoding for speaker identity, we utilized a publicly available x-vector extractor [29]. A 512-dimensional x-vector was used to condition the TTS model, added to the encoder output via a 512-by-256 linear transformation. Each x-vector was computed by averaging over all utterances from the same speaker, resulting in one representative embedding per speaker.

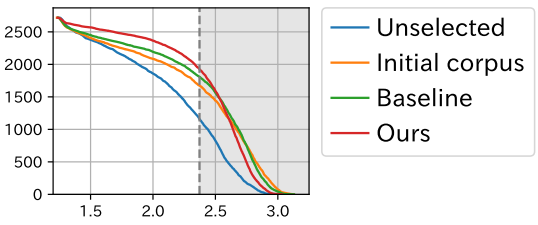
3) Speaker generation model: We adopted $d' = 28$ as the dimension of the diffusion model, determined by PCA to exceed a cumulative contribution rate of 99%. We concatenated the 28-dimensional input with a 16-dimensional embedding of t and passed it through linear layer, ReLU, linear layer, ReLU, and linear layer to output 28 dimensions. The hidden layer dimension was set to 56. We set time step T to be 200 and linearly increased β_t from $\beta_1 = 0.0001$ to $\beta_T = 0.05$. We split x-vectors from the 2719 speakers into train, validation, and test sets in a ratio of approximately 8:1:1, resulting in 2175, 272, and 272 speakers, respectively.

4) Compared methods: We compared the following data selection methods.

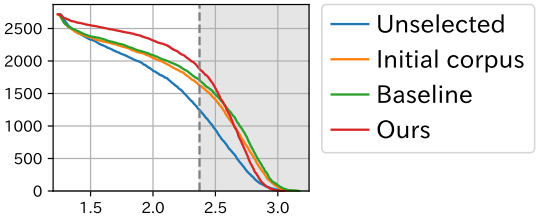
- **Unselected**: All the collected data was used for the TTS training; the training data size was approximately $60,000$ utterances.
- **Initial corpus**: To examine the improvement achieved by data addition, we investigate the initial corpus C_1.
- **Baseline**: We execute the evaluation-in-the loop data selection described in the previous study [9] with n as a parameter corresponding to the corpus size. Specifically, n was set to $3,943$, resulting in the same size as that of our proposed method.
- **Ours**: We construct the corpus using the proposed method. For the hyperparameter settings, we set $K = 2$, $r_1 = 0.1$ and $r_2 = 0.9$.

5) Evaluation: **High-quality speakers (real speakers):** We examine the number and distribution of speakers whose pseudo-MOS exceeded the threshold θ_{hq} and investigate whether the proposed method increases the number and distribution of high-quality speakers. We conducted this investigation for both the speaker list and the generated speaker dataset. The number of generated speakers is same as the speaker list, which is 2719.

High-quality speakers (generated speakers): We validate



(a) Evaluation on real speakers.



(b) Evaluation on generated speakers.

Fig. 3: Cumulative histograms of pseudo MOS. Y-axis value indicates number of speakers with higher score than x-axis value. The shaded area corresponds to high-quality speakers.

TABLE I: Ratio of high-quality speakers. Bold indicates the highest ratio in each block.

n	Method	Real	Generated
58500	Unselected	42.6%	45.8%
2454	Initial corpus	61.5%	59.8%
3943	Baseline	66.6%	62.8%
	Ours	**71.0%**	**69.2%**

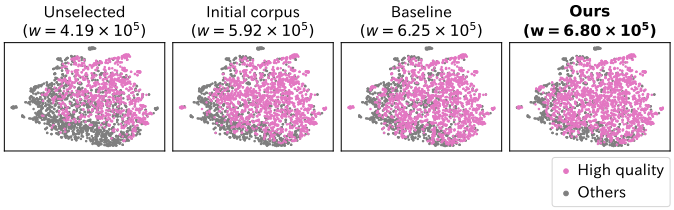


Fig. 4: Distributions of high quality speakers by each data selection method. Higher w indicates more diversity. Bold indicates the highest w among the methods.

the effectiveness of the speaker generation model. First, we demonstrate that the diffusion model captures complex structures that cannot be captured by a GMM in the x-vector space. To achieve this, we generate a speaker dataset of the same size as the test subset and calculate the Wasserstein 1-distance between the datasets. Since this value is a random variable that can vary depending on the generated dataset, we perform 30 samplings to calculate the mean and standard deviation. We adopt $M = 1, 2, ..., 10$ as the number of clusters for a GMM, and perform fitting for each.

Generated speaker distribtion: To verify whether the quality evaluation of generated speakers is valid as an assessment of the synthesis quality of unseen speakers, we calculate the Wasserstein 1-distance d_{RG} between real speakers and generated speakers in the latent variable space, serving as an index of deviation from real speakers. Since speaker characteristics are not always constant even for the same speaker, it is necessary to establish a reference value indicating that the generated speakers are different from real speakers. As a simple approach, one might consider calculating the Wasserstein-1distance d_{RR} between different real speakers in the x-vector space. However, the preprocessing does not distinguish whether speakers from different videos are the same, making d_{RR} inappropriate as a reference since it may inadvertently calculate the feature distances of different videos as if they were different speakers. Therefore, in this study, we calculate the Wasserstein 1-distance d_{GG} between different generated speakers as the reference. As each generated speaker is independently produced, we can expect them to be different speakers. If d_{RG} is of a similar magnitude to d_{GG}, we can consider the generated speakers as unseen speakers.

B. Results

1) High-quality speakers: Fig. 3a shows cumulative histograms of the pseudo MOSs, and Table I presents the ratio of high-quality speakers. "Ours" demonstrated a higher number of high-quality speakers compared to "Baseline" of the same size. In addition, Fig. 4 illustrates the distribution of high-quality speakers, where the spread is quantitatively represented by the value w. Here, w corresponds to the total length of the minimum spanning tree constructed from the x-vectors of the high-quality speakers, providing a measure of how widely the speakers are distributed in the embedding space. These results indicate that our proposed methods are data-efficient approaches for corpus construction. However, "Ours" demonstrated worse performance than "Baseline" in the range where pseudo MOS is 2.6 or higher. We can say that this is because our proposed method does not further enhance speakers whose pseudo-MOS exceeds θ_{hq}.

2) High-quality speakers from generated speakers: Fig. 3b shows cumulative histograms of the pseudo MOSs with generated speakers. The results are similar to the results with speaker list, and it can be said that "Ours" achieved more number of speakers whose synthetic quality is higher than θ_{hq}. This suggests that the TTS model trained in this experiment generalizes well to speakers and has the ability to synthesize even for speakers not included in the training data.

3) Investigation for speaker generation model: Fig. 5 shows the Wasserstein 1-distance between test speakers and generated speakers sampled from the speaker generation models. The error bars represent twice the standard deviation. In all cluster numbers, the diffusion model is more than twice the standard deviation as distant from the GMM, indicating that the diffusion model captures the speaker distribution more appropriately.

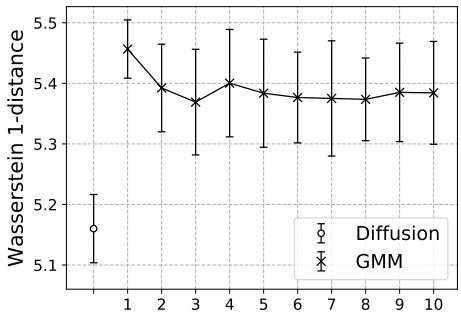


Fig. 5: Wasserstein 1-distance between the test dataset of speakers and generated speakers of the same size sampled from the speaker generation model. The Circle represents the diffusion model, and crosses correspond to GMM. The number of clusters for GMM is indicated on the x-axis. Error bars represent twice the standard deviation.

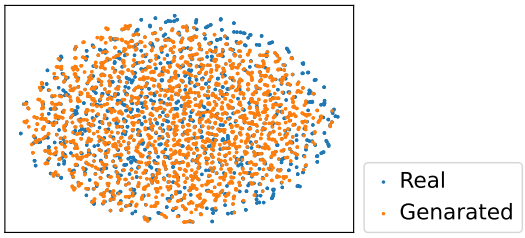


Fig. 6: The results of dimensionality reduction using t-SNE for the x-vectors of real speakers and generated speakers. It can be observed that the x-vectors of generated speakers are distributed over a wide range.

d_{RG}, d_{RR}, and d_{GG} took the following values, respectively.

$$d_{RR} = 3.492$$
$$d_{GG} = 270.187$$
$$d_{RG} = 268.382$$

d_{RR} is smaller than the other values, suggesting that the 2719 real speakers are actually counted as the same speaker with duplicates. Compared to d_{RR}, d_{RG} takes values similar to d_{GG}. Therefore, we can conclude that the generated speakers are different from real speakers.

To examine whether the generated speakers occupy the same embedding space as real speakers, we visualize the t-SNE projection of the combined x-vectors of both real and generated speakers, as shown in Fig. 6. The visualization indicates that the generated speaker embeddings are distributed within the same region as those of real speakers. Furthermore, the spread of the generated speaker embeddings demonstrates a diversity comparable to that of real speakers. These observations suggest that the generated speakers are unseen yet lie within the same speaker space as real speakers, supporting the validity of the generation process.

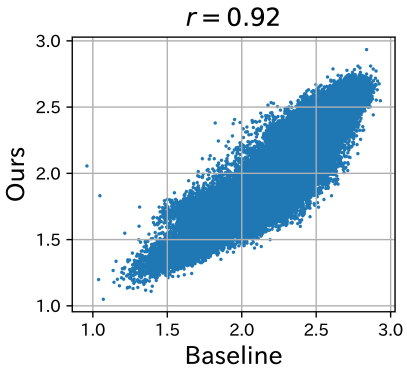


Fig. 7: Comparison of evaluation model.

4) Investigation for data quality estimator: Since our proposed method samples a subset of the candidate data for training the data quality estimator, it is important to investigate whether this modification affects the behavior of the estimator. Fig. 7 7 shows a comparison between the estimators trained under two different conditions: the "Baseline" method, which uses all candidate data for training, and our proposed method ("Ours"), which uses only 10% of that data.

The results indicate a strong correlation between the two estimators, suggesting that the proposed method maintains similar behavior despite the reduced training data. This finding implies that our approach effectively overcomes a critical limitation of the prior work—namely, the need to process the entire candidate dataset upfront to train the estimator.

IV. Conclusion

In this study, we proposed a novel corpus construction framework for multi-speaker text-to-speech (TTS) synthesis based on active learning. Our method incrementally selects informative samples from large-scale web data, enabling data-efficient training without requiring the full dataset to be downloaded in advance. By integrating a data quality estimator and evaluating informativeness via zero-shot synthesis, the proposed method efficiently increases the number of high-quality synthesizable speakers.

Experimental results demonstrated that our method outperforms conventional baseline approaches in terms of speaker coverage and synthetic speech quality, even with the same corpus size. We also showed that the TTS model generalizes well to unseen speakers, and the diffusion-based speaker generation model captures the complex distribution of speaker embeddings more effectively than conventional GMMs. Additionally, we confirmed that the data quality estimator remains effective even when trained on only a fraction of the candidate data.

These findings suggest that our approach is a promising direction for scalable and practical corpus construction for TTS systems.

Acknowledgment

This work was supported by JSPS KAKENHI 22H03639, 24KJ0860 and Moonshot R&D Grant Number JPMJPS2011.

REFERENCES

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. ICASSP*. IEEE, 2018, pp. 4779–4783.

[2] L.-W. Chen, S. Watanabe, and A. Rudnicky, "A vector quantized approach for text to speech synthesis on real-world spontaneous speech," in *Proc. AAAI*, 2023, pp. 12 644–12 652.

[3] Z. Ju, Y. Wang, K. Shen, X. Tan, D. Xin, D. Yang, Y. Liu, Y. Leng, K. Song, S. Tang *et al.*, "Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models," *arXiv preprint arXiv:2403.03100*, 2024.

[4] R. Sonobe, S. Takamichi, and H. Saruwatari, "JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis," *arXiv:1711.00354*, 2017.

[5] S. Takamichi, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji, and H. Saruwatari, "JVS corpus: free Japanese multi-speaker voice corpus," *arXiv:1908.06248*, 2019.

[6] O. Take, S. Takamichi, K. Seki, Y. Bando, and H. Saruwatari, "SaSLaW: Dialogue speech corpus with audio-visual egocentric information toward environment-adaptive dialogue speech synthesis," in *Proc. Interspeech*, 2024, pp. 1860–1864.

[7] 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 *Proc. Interspeech*, 2019, pp. 1526–1530.

[8] P. Puchtler, J. Wirth, and R. Peinl, "HUI-Audio-Corpus-German: A high quality TTS dataset," in *German Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer, 2021, pp. 204–216.

[9] K. Seki, S. Takamichi, T. Saeki, and H. Saruwatari, "Text-to-speech synthesis from dark data with evaluation-in-the-loop data selection," in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.

[10] G. Chen, S. Chai, G. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang *et al.*, "Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio," *arXiv preprint arXiv:2106.06909*, 2021.

[11] S. Takamichi, L. Kürzinger, T. Saeki, S. Shiota, and S. Watanabe, "JTubeSpeech: corpus of Japanese speech collected from YouTube for speech recognition and speaker verification," *arXiv:2112.09323*, 2021.

[12] W. Nakata, K. Seki, H. Yanaka, Y. Saito, S. Takamichi, and H. Saruwatari, "J-CHAT: Japanese large-scale spoken dialogue corpus for spoken dialogue language modeling," *arXiv preprint arXiv:2407.15828*, 2024.

[13] K. Seki, S. Takamichi, T. Saeki, and H. Saruwatari, "Diversity-based core-set selection for text-to-speech with linguistic and acoustic features," in *Proc. ICASSP*. IEEE, 2024, pp. 1–5.

[14] C. Guo, B. Zhao, and Y. Bai, "Deepcore: A comprehensive library for coreset selection in deep learning," in *International Conference on Database and Expert Systems Applications*. Springer, 2022, pp. 181–195.

[15] D. Park, D. Papailiopoulos, and K. Lee, "Active learning is a strong baseline for data subset selection," in *NeurIPS 2022 Workshop*, 2022.

[16] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[17] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*. IEEE, 2018, pp. 5329–5333.

[18] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "Ctc-segmentation of large corpora for german end-to-end speech recognition," in *Speech and Computer*, A. Karpov and R. Potapova, Eds. Cham: Springer International Publishing, 2020, pp. 267–278.

[19] K. Seki, S. Takamichi, T. Saeki, and H. Saruwatari, "TTSOps: A closed-loop corpus optimization framework for training multi-speaker TTS models from dark data," *arXiv preprint arXiv:2506.15614*, 2025.

[20] D. Stanton, M. Shannon, S. Mariooryad, R. Skerry-Ryan, E. Battenberg, T. Bagby, and D. Kao, "Speaker generation," in *Proc. ICASSP*. IEEE, 2022, pp. 7897–7901.

[21] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Proc. NeurIPS*, vol. 33, pp. 6840–6851, 2020.

[22] K. Oko, S. Akiyama, and T. Suzuki, "Diffusion models are minimax optimal distribution estimators," in *Pcor. ICML*. PMLR, 2023.

[23] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10 850–10 869, 2023.

[24] "ITA corpus," https://github.com/mmorise/ita-corpus.

[25] 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," *Proc. ICLR*, 2021.

[26] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *Proc. NeurIPS*, vol. 33, pp. 17 022–17 033, 2020.

[27] "HiFi-GAN," https://github.com/jik876/hifi-gan.

[28] "FastSpeech 2-JSUT," https://github.com/Wataru-Nakata/FastSpeech2-JSUT.

[29] "x-vector," https://github.com/sarulab-speech/xvector_jtubespeech.