



SHEET: A Multi-purpose Open-source Speech Human Evaluation Estimation Toolkit

Wen-Chin Huang¹, Erica Cooper², Tomoki Toda³

¹Graduate School of Informatics, Nagoya University, Japan

²National Institute of Information and Communications Technology, Japan

³Information Technology Center, Nagoya University, Japan

wen.chinhuang@g.sp.m.is.nagoya-u.ac.jp

Abstract

We introduce SHEET, a multi-purpose open-source toolkit designed to accelerate subjective speech quality assessment (SSQA) research. SHEET stands for the Speech Human Evaluation Estimation Toolkit, which focuses on data-driven deep neural network-based models trained to predict human-labeled quality scores of speech samples. SHEET provides comprehensive training and evaluation scripts, multi-dataset and multi-model support, as well as pre-trained models accessible via Torch Hub and HuggingFace Spaces. To demonstrate its capabilities, we re-evaluated SSL-MOS, a speech self-supervised learning (SSL)-based SSQA model widely used in recent scientific papers, on an extensive list of speech SSL models. Experiments were conducted on two representative SSQA datasets named BVCC and NISQA, and we identified the optimal speech SSL model, whose performance surpassed the original SSL-MOS implementation and was comparable to state-of-the-art methods.

Index Terms: speech quality assessment, open-source

1. Introduction

Speech quality assessment (SQA) refers to the task of evaluating the quality of speech signals [1–3], and is an essential component to various applications, including telecommunications and speech generation tasks, from text-to-speech (TTS), voice conversion (VC) to speech enhancement. The gold standard for evaluating speech signals is to ask human listeners to assign subjective ratings based on perceived quality, using protocols like the mean opinion score (MOS) test. However, since conducting such evaluations can be expensive and time-consuming, objective metrics have been proposed. Among those, some metrics evaluate dimensions like clarity and intelligibility (like the short-time objective intelligibility measure (STOI) [4]) or rely on human-defined distances between hand-crafted features (like the Mel cepstral distortion (MCD) [5]).

We are particularly interested in the task of developing metrics that are **optimized directly using human preference data**, where one representative early attempt was the perceptual evaluation of speech quality (PESQ) metric [6]. We term such a task **subjective speech quality assessment (SSQA)**. Since such metrics are often data-driven and based on machine learning models, it is no exception for the field of SSQA to benefit from the rapid development of deep neural networks (DNNs) in recent decades [7–9]. A scientific competition series for promoting SSQA named the VoiceMOS Challenge (VMC) [10–12] was founded in 2022, and in that year, it was shown that the best-performing system, UTMOS [13], achieved a high correlation (0.959) with human ratings. Such results led to the adaptation of SSQA models as an objective measure for speech quality in TTS research and nourished increasing interest in SSQA as a

Table 1: Comparison of existing open-source speech quality assessment toolkits and SHEET.

Toolkit	Inference	Model training	Multi-model	Multi-dataset
[14–16], etc.	✓		✓	
[13, 17–19], etc.	✓	✓		
SHEET	✓	✓	✓	✓

critical research area.

In the modern era of artificial intelligence research, open-source activities play an important role in promoting and facilitating development. In the field of SSQA, existing open-source toolkits can be categorized into two types, as summarized in Table 1. The first type (row one) aims to provide an easy-to-use interface to multiple off-the-shelf metrics and pre-trained SQA models [14–16]. However, these toolkits often do not provide training recipes, making it difficult for researchers to develop custom models. The second type (row two) is often reproducible training recipes for their specific scientific papers [9, 13, 17–19], typically supporting only a limited range of datasets and models. To summarize, the lack of flexibility of existing toolkits poses challenges for researchers seeking to experiment with different datasets and architectures.

To address these limitations, we introduce **SHEET**¹, an open-source toolkit designed to facilitate SSQA research. SHEET stands for the **S**peech **H**uman **E**valuation **E**stimation **T**oolkit, which is designed with the following use cases in mind:

- Provide complete training and evaluation scripts for conducting SSQA experiments, making it accessible for newcomers to SSQA research.
- Enable researchers with existing SSQA models to evaluate their models on multiple testing sets using a standardized framework.
- Offer pre-trained SSQA models that can be easily accessed through `torch.hub.load` or HuggingFace.

To demonstrate the capabilities of SHEET, we conducted an extensive experiment using SSL-MOS [18], a representative SSQA model, to re-evaluate the effectiveness of existing self-supervised learning (SSL) models in predicting human judgments. Experiments were conducted on two datasets: BVCC [20], which consists of speech samples generated by a total of 187 TTS and VC systems, and NISQA [17], covering simulated and real-world speech samples in a wide range of distorted conditions (noise, packet-loss, warping, low-bitrate, etc). We identified the optimal SSL model for each dataset, whose

¹<https://github.com/unilight/sheet>

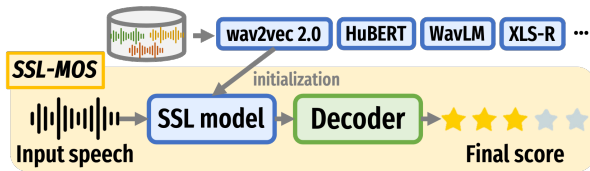


Figure 1: Illustration of SSL-MOS.

result surpassed the original SSL-MOS model and is comparable with the state-of-the-art method. Such results underscore the usefulness of the proposed toolkit and its adaptability to different datasets.

2. SHEET design and supported models

2.1. Overview

The structure of SHEET follows the design of popular speech processing toolkits like Kaldi [21] and ESPnet [22]. Specifically, in such a design, there are two core parts: the *library* implements the model architectures, loss computation, and training logic, which are based on Python. On the other hand, a *recipe* is a collection of bash or python scripts that provides easy-to-understand instructions to complete an experiment, from data preprocessing, model training to benchmark evaluation. In practice, each training dataset has one recipe, and there are many configuration files to choose from, each representing a set of hyper-parameters of the model and the training procedure.

2.2. Supported datasets

SHEET currently provides training recipes for a total of seven training datasets and twelve testing sets, demonstrating a diverse collection of datasets with different properties. The domains span from samples synthesized by speech generation systems like TTS and VC to speech that underwent a variety of distortions, including artificially added and real noise, reverberation, VoIP, transmission, and replay. A total of six languages were covered, and the sampling frequencies also ranged from 8000 Hz to 48000 Hz. Finally, some training datasets provide listener-wise scores, allowing for listener modeling techniques.

Due to space limits, we omit the introduction to each training and testing dataset. Interested readers and users can refer to individual papers, and access the recipes as they are already made publicly available.

2.3. Supported models and features

Early DNN-based SSQA models were mainly built upon convolutional and recurrent layers and trained from scratch using a certain SSQA dataset [7–9, 17, 23]. As the SSL paradigm revolutionized almost all fields in speech processing [24], SSQA was no exception [18, 25]. Thus, all models supported in SHEET are based on SSL. Below, we introduce one of the most representative SSL-based SSQA models.

SSL-MOS. SSL-MOS has been widely used as an objective measure of speech quality in scientific papers and even challenges like the Singing Voice Conversion Challenge [26], due to its simplicity and strong performance. Figure 1 is an illustration of SSL-MOS. Given an input speech x , the SSL model $\text{SSL}(\cdot)$ outputs a sequence of frame-wise hidden representations which is further sent into a decoder $\text{DEC}(\cdot)$ to generate a frame-wise score sequence. The training objective \mathcal{L} is an L1 loss between

the ground truth score y and the predicted score \hat{y} obtained by applying time pooling over the frame-wise score sequence:

$$\mathcal{L} = \|y - \hat{y}\|_1 = \|y - \text{TimePooling}(\text{DEC}(\text{SSL}(x)))\|_1. \quad (1)$$

During training, $\text{SSL}(\cdot)$ and $\text{DEC}(\cdot)$ are jointly optimized. During inference, the process to obtain the predicted score \hat{y} is the same as that during training.

Other features. SHEET implements features and components that were reported useful in recent SSQA papers, including:

- Range clipping, repetitive padding, and the clipped L1 loss, as proposed in [27].
- Listener modeling, as proposed in [27, 28].
- Contrastive loss, as proposed in [13].
- Multiple dataset training, as proposed in [29].
- Retrieval augmentation based on k-nearest neighbors, as proposed in [19]².

2.4. Training recipe

A typical training recipe in SHEET involves the following stages.

Downloading stage. We provide automatic downloading scripts or manual instructions to obtain the dataset. To ensure reproducibility, we also provide pre-trained model checkpoints for each dataset, which are hosted on HuggingFace.

Data preparation stage. We provide scripts to integrate all necessary information for training, inference, and evaluation into files in `.csv` format. Information includes the audio `.wav` path, sample ID, system ID (if available), and human score.

Model training stage. Given a configuration file (in `yaml` format, the SSQA model training takes place in this stage. We provide convenient features, including automatic early stopping based on a pre-defined criterion on the validation set, and inference result visualization for inspection during training.

Evaluation stage. We provide an integrated script to evaluate the trained model on all the supported test sets.

2.5. Inference usages

SHEET adopts the `torch.hub` function to support an easy deployment of our provided pre-trained model in Python. The following example code demonstrates how to estimate the quality score given an audio sample path.

```
import torch

# load model.
# <anonymized> will be updated upon acceptance
predictor = torch.hub.load(
    "<anonymized>/sheet:v0.1.0",
    "default"
)

# prediction. example output: 3.6066928
score = predictor.predict(
    wav_path="/path/to/wav/file.wav"
)
```

For researchers who wish to benchmark the SSQA model they developed, SHEET provides instructions to download the testing sets. After predictions are generated, they may utilize the provided calculation script to assess the performance of the SSQA model.

²We also attempted to reproduce the full RAMP model [19], which is the current state-of-the-art on BVCC and the top system in VMC 2023 and 2024. However, the official implementation did not release training scripts, and as of the time of the submission, we could not fully reproduce the results as shown in the paper.

3. Experiments

To demonstrate the capabilities of SHEET, in this section, we conduct an extensive experiment to re-evaluate the effectiveness of existing SSL models in the SSL-MOS framework.

3.1. Experimental setting

3.1.1. Dataset

The following two datasets are used in the experiments.

- The **BVCC** dataset [20] was used in the main track of VMC 2022. It contains English speech samples in 16 kHz and their MOS ratings from 187 different TTS and VC systems, which mainly come from past years of the Blizzard Challenges (BC) and Voice Conversion Challenges (VCC), as well as published samples from ESPnet-TTS [30]. Each sample was rated by 8 distinct listeners.
- The **NISQA** dataset contains several subsets, each containing speech samples with a sampling frequency of 48 kHz. The `NISQA TRAIN SIM` and `NISQA VAL SIM` sets contained English speech samples with simulated distortions and background noises, while the `NISQA TRAIN LIVE` and `NISQA VAL LIVE` sets contained English live Skype and phone recordings, with real distortions created during recording. Each sample in these four sets was rated by five listeners. We combined the `NISQA TRAIN SIM` and `NISQA TRAIN LIVE` sets as the training set, and combined the `NISQA VAL SIM` and `NISQA VAL LIVE` sets as the validation set, resulting in 11020/2700 samples, respectively. For the test sets, we used the following sets: **P501** and **FOR** sets each included 240 English samples with simulated distortions and live VoIP calls where speech samples were played back directly from the laptop. Each sample in these two sets was rated by thirty listeners. The **LIVETALK** set consists of 232 real German phone call recordings from different backgrounds and distortions. Each sample in these two sets was rated by 24 listeners.

3.1.2. Model and training settings

We mainly experimented with the SSL-MOS model, and we utilized S3PRL [31], an open-source toolkit that provides a convenient interface to access a large collection of speech and audio SSL models. Using the supported models in S3PRL, our experiment spanned 22 SSL models. Due to space limits, we refer authors to the S3PRL codebase for details of each SSL model³. The output from the last layer of the SSL model was used by default. Since most SSL models take 16 kHz waveforms as input, all input speech samples were resampled to 16 kHz. The training batch size was set to 16, and the SGD optimizer with an initial learning rate of 0.001 and a momentum of 0.9 was used. All training runs were allowed to execute for a maximum of 100,000 steps, and if the 5 best checkpoints had not been updated for 2000 steps, the training automatically halted. For more detailed settings, readers can refer to `egs/bvcc/conf/ssl-mos-wav2vec2.yaml`, which is a representative configuration file.

3.1.3. Comparing systems

As the main purpose of SHEET is to promote open-source, reproducible SSQA research, our experiments focus on evaluating publicly available model checkpoints of existing systems, in-

³https://s3prl.github.io/s3prl/tutorial/upstream_collection.html

stead of just reporting numbers in previous papers. Therefore, we chose five models and used the checkpoints from their official implementation codebases as the comparing systems. The **SSL-MOS** [18], **UTMOS** [13], and **RAMP+** were trained with the BVCC training set. SSL-MOS and UTMOS were strong-performing models in VMC 2022. RAMP+ is an improved version of the previous RAMP model [19], which was the top-performing system in VMC 2023⁴. The **NISQA** model [17] was trained on a large collection of datasets including the NISQA training sets, and **DNSMOS P808** [9] was trained using an internal crowdsourcing SSQA dataset, not including the NISQA training sets. The choice of the last two models was driven by their common use in evaluating noisy and distorted speech. Note that the last two models are not based on SSL.

3.1.4. Evaluation metrics

For BVCC, we reported the following two metrics: system-level mean squared error (Sys MSE) and Spearman’s rank correlation coefficient (Sys SRCC). For NISQA, following the original paper [17], we reported the following two metrics: utterance-level MSE (Utt MSE) and utterance-level linear correlation coefficient (Utt LCC). Readers may refer to `sheet/evaluation/metrics.py` for the implementation of the calculation.

3.2. Re-evaluating SSL models in SSL-MOS

Table 2 shows the experimental results over the 22 SSL models we investigated. On the BVCC test set, the `data2vec large` model [32] and the `HuBERT large` model achieved the best Sys MSE and Sys SRCC scores, respectively. On the NISQA dataset, on average, the `WavLM large` model [33] and the `XLS-R 1b` model [34] achieved the best and second best scores on the Sys MSE and Sys SRCC metrics, respectively. The fact that the optimal SSL model for NISQA is different from that of BVCC demonstrates that SSL-MOS is sensitive to the choice of the SSL model.

We also observed some other interesting results. On the BVCC test sets, large SSL models perform better than their base variants. Such a tendency also holds on the NISQA test sets, except for the `HuBERT` and `data2vec` model. On the NISQA test `LIVETALK` set, `XLS-R 1b` was the best among all SSL models, which is likely because `XLS-R 1b` was trained in 128 languages, while many other SSL models were trained using English data only. This highlights the importance of pre-training the SSL model with datasets in multiple languages in multilingual SSQA. Finally, we included results from three SSL models trained using not only speech but also general audio. Although we expected such a model can learn better representations, especially for NISQA which consists of mostly distorted speech, none of them outperformed other SSL models on both BVCC and NISQA datasets. Nonetheless, we still remain optimistic on this direction.

3.3. Results with comparing models

We then compare the best models in the previous section to existing publicly available models. On the BVCC testing set, the SSL-MOS models with `HuBERT large` and `data2vec large` outperformed not only NISQA and `DNSMOS P808`, which were not trained on the BVCC training set but also SSL-MOS and

⁴The technical paper of RAMP+ has not yet been published as of the submission date of this paper, and the model checkpoints of RAMP were not provided. We therefore only reported RAMP+ results.

Table 2: Experimental results. For MSE, the smaller the better, and for LCC and SRCC, the larger the better. Boldface and underline indicate the best and second-best score in each column, respectively. †: results calculated using official pre-trained model checkpoints.

Model	SSL model	BVCC test		NISQA test FOR		NISQA test LIVETALK		NISQA test P501		NISQA average	
		Sys MSE	Sys SRCC	Utt MSE	Utt LCC	Utt MSE	Utt LCC	Utt MSE	Utt LCC	Utt MSE	Utt LCC
SSL-MOS	CPC	0.186	0.873	0.367	0.723	0.801	0.499	0.440	0.790	0.536	0.671
	APC	0.194	0.845	0.355	0.737	0.381	0.761	0.457	0.817	0.398	0.772
	VQ-APC	0.199	0.841	0.402	0.691	0.419	0.728	0.508	0.789	0.443	0.736
	NPC	0.321	0.802	0.720	0.524	0.516	0.673	0.778	0.603	0.671	0.600
	DeCoAR 2.0	0.255	0.917	0.148	0.903	0.306	0.824	0.455	0.901	0.303	0.876
	wav2vec	0.200	0.870	0.261	0.824	0.344	0.841	0.416	0.865	0.340	0.843
	vq-wav2vec	0.166	0.870	0.270	0.806	0.469	0.747	0.568	0.841	0.436	0.798
	wav2vec 2.0 base	0.146	0.928	0.128	0.918	0.444	0.806	0.363	0.928	0.312	0.884
	wav2vec 2.0 large	0.100	0.929	0.139	0.903	0.235	0.863	0.182	0.918	0.185	0.895
	HuBERT base	0.175	0.916	0.117	0.919	0.432	0.777	0.249	0.918	0.266	0.871
	HuBERT large	0.109	0.936	0.209	0.862	0.482	0.791	0.323	0.879	0.338	0.844
	data2vec base	0.131	0.905	0.202	0.860	0.280	0.832	0.436	0.908	0.306	0.867
	data2vec large	0.098	0.919	0.300	0.888	0.224	0.884	0.608	0.891	0.377	0.888
	WavLM base	0.131	0.920	0.154	0.906	0.331	0.809	0.305	0.919	0.263	0.878
	WavLM large	0.119	0.928	0.098	0.933	0.177	0.912	0.140	0.945	0.138	0.930
	UniSpeech-SAT base	0.118	0.917	0.131	0.923	0.341	0.794	0.394	0.908	0.289	0.875
	UniSpeech-SAT large	0.118	0.925	0.150	0.895	0.217	0.888	0.230	0.918	0.199	0.900
	XLSR	0.129	0.916	0.146	0.900	0.219	0.886	0.232	0.916	0.199	0.901
	XLS-R 1b	0.124	0.922	0.121	0.920	0.170	0.914	0.154	0.943	0.148	0.926
	BYOL-A 2024-dim	0.496	0.720	0.437	0.673	0.647	0.519	0.471	0.813	0.518	0.668
SSAST frame	0.132	0.913	0.353	0.797	0.564	0.633	0.430	0.856	0.449	0.762	
MAE-ASR frame	0.162	0.921	0.123	0.921	0.322	0.812	0.284	0.926	0.243	0.886	
SSL-MOS [18]†		0.113	0.923	0.633	0.776	1.496	0.731	0.468	0.852	0.866	0.786
UTMOS [13]†		0.148	0.925	0.364	0.802	1.044	0.772	0.400	0.855	0.603	0.810
RAMP+ †		0.086	0.939	0.525	0.800	1.238	0.772	0.882	0.885	0.882	0.819
NISQA [17]†		0.978	0.713	0.207	0.875	0.395	0.771	0.328	0.900	0.310	0.849
DNSMOS P808 [9]†		0.724	0.773	1.295	0.534	0.819	0.713	1.368	0.671	1.161	0.639

UTMOS, which used the wav2vec 2.0 base model [35]. This result could be because of our hyperparameter choices, as well as the use of the features described in Section 2.3. However, our best results are comparable but still fall behind RAMP+, the state-of-the-art method on the BVCC test set. Reproducing RAMP+ will be an important future work.

On the NISQA testing sets, the SSL-MOS models with WavLM large and XLS-R 1b surpassed all five comparing systems, setting new strong baseline results on the NISQA testing sets for future research. Notably, the results of the NISQA model and DNSMOS fell behind those of many SSL-MOS models explored in the previous section. This again highlights the importance of using SSL models in SSQA.

4. Conclusions

In this work, we introduced SHEET, an open-source toolkit designed to facilitate research in SSQA. By addressing the limitations of existing toolkits, SHEET provides a standardized and flexible framework for conducting SSQA experiments, evaluating existing models across multiple datasets, or simply employing the provided pre-trained models. To demonstrate its effectiveness, we conducted extensive experiments using SSL-MOS and re-evaluated SSL models on the BVCC and NISQA datasets. Using SHEET, we were able to easily identify optimal SSL models for different datasets, achieving performance comparable to or better than comparing methods. This highlights the versatility and practical impact of SHEET in advancing SSQA research.

SHEET is an ongoing effort, and we aim to continually maintain the toolkit by implementing emerging state-of-the-art methods. Moving forward, we also plan to include methods to evaluate other dimensions in speech, including speaker similarity prediction [36] and descriptive evaluation using natural language [37, 38].

We also noticed that in Table 2, when models trained solely

on BVCC (including SSL-MOS, UTMOS, and RAMP+) were tested on NISQA, the performance was worse than that of the NISQA model, and the NISQA model was not even using SSL. This highlights the insufficient out-of-domain generalization ability of current SSQA models. In the search for a general-purpose SSQA model, investigating and improving the out-of-domain generalization ability of SSQA models will be another important future direction.

5. Acknowledgements

This work was partly supported by JSPS KAKENHI Grant Number 25K00143 and JST AIP Acceleration Research JP-MJCR25U5, Japan.

6. References

- [1] P. C. Loizou, “Speech quality assessment,” in *Multimedia Analysis, Processing and Communications*, W. Lin, D. Tao, J. Kacprzyk, Z. Li, E. Izquierdo, and H. Wang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 623–654.
- [2] S. Miller, W.-Y. Chan, N. Ct, T. H. Falk, A. Raake, and M. Wltermann, “Speech Quality Estimation: Models and Trends,” *IEEE Signal Processing Magazine*, vol. 28, no. 6, pp. 18–28, 2011.
- [3] E. Cooper, W.-C. Huang, Y. Tsao, H.-M. Wang, T. Toda, and J. Yamagishi, “A review on subjective and objective evaluation of synthetic speech,” *Acoustical Science and Technology*, vol. 45, no. 4, pp. 161–183, 2024.
- [4] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “An Algorithm for Intelligibility Prediction of TimeFrequency Weighted Noisy Speech,” *IEEE/ACM TASLP*, vol. 19, no. 7, pp. 2125–2136, 2011.
- [5] R. Kubichek, “Mel-cepstral distance measure for objective speech quality assessment,” in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1, 1993, pp. 125–128 vol.1.
- [6] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, “Perceptual evaluation of speech quality (PESQ)-a new method

- for speech quality assessment of telephone networks and codecs,” in *Proc. ICASSP*, vol. 2, 2001, pp. 749–752.
- [7] B. Patton, Y. Agiomyrgiannakis, M. Terry, K. Wilson, R. A. Saurous, and D. Sculley, “AutoMOS: Learning a non-intrusive assessor of naturalness-of-speech,” *arXiv preprint arXiv:1611.09207*, 2016.
 - [8] C.-C. Lo, S.-W. Fu, W.-C. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H.-M. Wang, “MOSNet: Deep Learning-Based Objective Assessment for Voice Conversion,” in *Proc. Interspeech*, 2019, pp. 1541–1545.
 - [9] C. K. A. Reddy, V. Gopal, and R. Cutler, “DNSMOS: A Non-Intrusive Perceptual Objective Speech Quality Metric to Evaluate Noise Suppressors,” in *Proc. ICASSP*, 2021, pp. 6493–6497.
 - [10] W.-C. Huang, E. Cooper, Y. Tsao, H.-M. Wang, T. Toda, and J. Yamagishi, “The VoiceMOS Challenge 2022,” in *Proc. Interspeech*, 2022, pp. 4536–4540.
 - [11] E. Cooper, W.-C. Huang, Y. Tsao, H.-M. Wang, T. Toda, and J. Yamagishi, “The Voicemos Challenge 2023: Zero-Shot Subjective Speech Quality Prediction for Multiple Domains,” in *Proc. ASRU*, 2023, pp. 1–7.
 - [12] W.-C. Huang, S.-W. Fu, E. Cooper, R. Zezario, T. Toda, H.-M. Wang, J. Yamagishi, and Y. Tsao, “The Voicemos Challenge 2024: Beyond Speech Quality Prediction,” in *Proc. SLT*, 2024.
 - [13] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, “UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022,” in *Proc. Interspeech*, 2022, pp. 4521–4525.
 - [14] 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,” in *Proc. ICASSP*, 2023, pp. 1–5.
 - [15] T. Saeki, S. Maiti, S. Takamichi, S. Watanabe, and H. Saruwatari, “SpeechBERTScore: Reference-Aware Automatic Evaluation of Speech Generation Leveraging NLP Evaluation Metrics,” in *Proc. Interspeech*, 2024, pp. 4943–4947.
 - [16] J. Shi, H. jin Shim, J. Tian, S. Arora, H. Wu, D. Petermann, J. Q. Yip, Y. Zhang, Y. Tang, W. Zhang, D. S. Alharthi, Y. Huang, K. Saito, J. Han, Y. Zhao, C. Donahue, and S. Watanabe, “Versa: A versatile evaluation toolkit for speech, audio, and music,” 2024. [Online]. Available: <https://arxiv.org/abs/2412.17667>
 - [17] G. Mittag, B. Naderi, A. Chehadi, and S. Miller, “NISQA: A Deep CNN-Self-Attention Model for Multidimensional Speech Quality Prediction with Crowdsourced Datasets,” in *Proc. Interspeech*, 2021, pp. 2127–2131.
 - [18] E. Cooper, W.-C. Huang, T. Toda, and J. Yamagishi, “Generalization ability of MOS prediction networks,” in *Proc. ICASSP*, 2022, pp. 8442–8446.
 - [19] H. Wang, S. Zhao, X. Zheng, and Y. Qin, “RAMP: Retrieval-Augmented MOS Prediction via Confidence-based Dynamic Weighting,” in *Proc. Interspeech*, 2023, pp. 1095–1099.
 - [20] E. Cooper and J. Yamagishi, “How do voices from past speech synthesis challenges compare today?” in *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*, 2021, pp. 183–188.
 - [21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The Kaldi Speech Recognition Toolkit,” in *Proc. ASRU*, 2011.
 - [22] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, “ESPnet: End-to-End Speech Processing Toolkit,” in *Proc. Interspeech*, 2018, pp. 2207–2211.
 - [23] Z. Zhang, P. Vyas, X. Dong, and D. S. Williamson, “An End-To-End Non-Intrusive Model for Subjective and Objective Real-World Speech Assessment Using a Multi-Task Framework,” in *Proc. ICASSP*, 2021, pp. 316–320.
 - [24] A. Mohamed, H.-y. Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaløe *et al.*, “Self-supervised speech representation learning: A review,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1179–1210, 2022.
 - [25] W.-C. Tseng, C. Yu Huang, W.-T. Kao, Y. Y. Lin, and H. Yi Lee, “Utilizing Self-Supervised Representations for MOS Prediction,” in *Proc. Interspeech*, 2021, pp. 2781–2785.
 - [26] W.-C. Huang, L. P. Violeta, S. Liu, J. Shi, and T. Toda, “The Singing Voice Conversion Challenge 2023,” in *Proc. ASRU*, 2023, pp. 1–8.
 - [27] Y. Leng, X. Tan, S. Zhao, F. Soong, X.-Y. Li, and T. Qin, “MB-NET: MOS Prediction for Synthesized Speech with Mean-Bias Network,” in *Proc. ICASSP*, 2021, pp. 391–395.
 - [28] W.-C. Huang, E. Cooper, J. Yamagishi, and T. Toda, “LDNet: unified listener dependent modeling in MOS prediction for synthetic speech,” in *Proc. ICASSP*, 2022, pp. 896–900.
 - [29] J. Pieper and S. Vorn, “Alignnet: Learning dataset score alignment functions to enable better training of speech quality estimators,” in *Proc. Interspeech*, 2024, pp. 82–86.
 - [30] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, “Espnet-TTS: Unified, Reproducible, and Integratable Open Source End-to-End Text-to-Speech Toolkit,” in *Proc. ICASSP*, 2020, pp. 7654–7658.
 - [31] S.-w. Yang, H.-J. Chang, Z. Huang, A. T. Liu, C.-I. Lai, H. Wu, J. Shi, X. Chang, H.-S. Tsai, W.-C. Huang, T.-h. Feng, P.-H. Chi, Y. Y. Lin, Y.-S. Chuang, T.-H. Huang, W.-C. Tseng, K. Lakhota, S.-W. Li, A. Mohamed, S. Watanabe, and H.-y. Lee, “A Large-Scale Evaluation of Speech Foundation Models,” *IEEE/ACM TASLP*, vol. 32, pp. 2884–2899, 2024.
 - [32] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, “data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language,” in *Proc. ICML*, 2022, pp. 1298–1312.
 - [33] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, “WavLM: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
 - [34] A. Babu, C. Wang, A. Tjandra, K. Lakhota, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, “XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale,” in *Proc. Interspeech*, 2022, pp. 2278–2282.
 - [35] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Proc. NeruIPS*, 2020.
 - [36] J. Ahn, Y. Kim, Y. Choi, D. Kwak, J.-H. Kim, S. Mun, and J. S. Chung, “VoxSim: A perceptual voice similarity dataset,” in *Proc. Interspeech*, 2024, pp. 2580–2584.
 - [37] S. Deshmukh, D. Alharthi, B. Elizalde, H. Gamper, M. Al Ismail, R. Singh, B. Raj, and H. Wang, “PAM: Prompting Audio-Language Models for Audio Quality Assessment,” in *Proc. Interspeech*, 2024, pp. 3320–3324.
 - [38] C. Chen, Y. Hu, S. Wang, H. Wang, Z. Chen, C. Zhang, C.-H. H. Yang, and E. Chng, “Audio Large Language Models Can Be Descriptive Speech Quality Evaluators,” in *Proc. ICLR*, 2025.