DNSMOS Pro: A Reduced-Size DNN for Probabilistic MOS of Speech

Fredrik Cumlin[1], *Xinyu Liang*[1], *Victor Ungureanu*[2], *Chandan K. A. Reddy*[2], *Christian Schüldt*[2], *Saikat Chatterjee*[3]

[1] Codemill AB, Umeå, Sweden [2] Google LLC
[3] Digital Futures and KTH Royal Institute of Technology, Stockholm, Sweden

fcumlin@gmail.com, hope9954@icloud.com, ungureanu@google.com, chandanka@google.com, cschuldt@google.com, sach@kth.se

Abstract

We propose a deep neural network-based architecture and training design for objective non-intrusive speech quality assessment. The proposed method builds on DNSMOS, and we call the proposed model DNSMOS Pro. DNSMOS Pro has a reduced-size architecture suitable for VoIP, a relatively simple training design using only the mean opinion score (MOS) as the target label, and predicts the posterior distribution of MOS given an input speech clip. This means DNSMOS Pro can be trained when only the MOS is reported on a subjectively rated dataset. Furthermore, we implement several non-intrusive speech quality methods and compare them to DNSMOS Pro when training and testing on different subjectively rated datasets. DNSMOS Pro has significantly better performance on these benchmark datasets compared to similar DNN-based non-intrusive speech quality methods, and competitive results to methods assuming auxiliary information in the datasets.
Index Terms: Speech quality assessment, deep neural network, maximum-likelihood, voice conversion challenge.

1. Introduction

Non-intrusive speech quality assessment (SQA) is the task of giving a speech quality score on a distorted signal without a clean reference. The subjective listening test is the golden standard for assigning quality labels to speech clips but is both costly and time-consuming. Thus, objective non-intrusive speech quality measures are of interest, which are cheaper and save time. Recently, several deep neural network (DNN) based methods have been proposed, which are becoming more and more complex concerning the underlying dataset. For the interest of real-time processing in VoIP, we aim for a model with reduced complexity and simple design to fit general application scenarios while maintaining high performance.

Starting from MOSNet [1], many end-to-end trained DNN-based non-intrusive SQA methods have been proposed. They generally follow a CNN-BLSTM architecture that maps the speech clip features to mean-opinion-score (MOS) labels. To fully utilize some training datasets that have individual scores from each rater, many models, such as MBNet, LDNet, LaMOSNet, MOSLight [2, 3, 4, 5], have achieved better performance with advanced model design while increasing the model size at the same time.

Parallel to the development of the abovementioned methods, several works have studied the usage of self-supervised learning (SSL) models such as Hubert [6] and Wav2vec2.0 [7] as feature extractors for the non-intrusive speech quality task. Experiments suggest that these models have a better generalization ability on speech quality datasets [8] but at the cost of not being trained end-to-end and typically being more than 100 times larger than end-to-end counterparts. Examples include UTMOS, ZevoMOS, and LE-SSL-MOS [9, 10, 11]. Most works are inspired by the LDNet model design and use individual raters' scores.

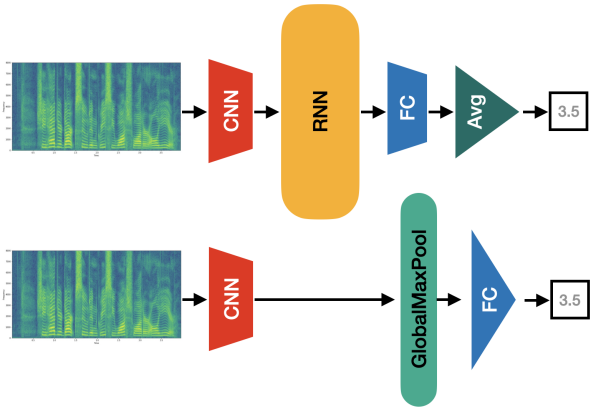
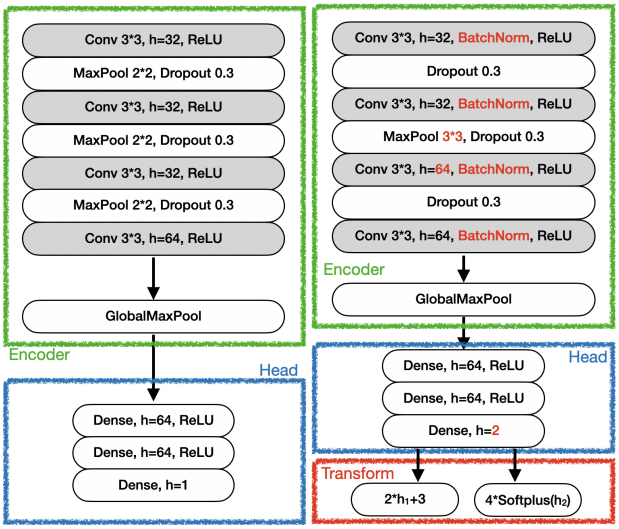


Figure 1: *DNSMOS (bottom) removes the heavily parameterized RNN module used in other models, and uses a parameter-free GlobalMaxPool layer instead. Frame-wise computations of the dense layers are also removed thereafter.*

For online processing purposes, DNSMOS [12] was introduced that dropped the computation-heavy RNN module and replaced it with a simple yet strong architecture. The architecture itself has not to this day been well-studied on public benchmark datasets, so the usefulness hereof compared to similar end-to-end models is poorly understood.

All these mentioned methods predict MOS as a point estimate, which doesn't contain information about the confidence for these estimations. One method based on the design of MeanNet from MBNet, called DeePMOS [13], has been studied using a Gaussian posterior to fit the MOS labels in a Bayesian training framework. However, this model is too big to fit in an online processing pipeline, thus it's of interest to design a reduced-size model for probabilistic estimation of MOS.

Our contribution: We present a novel architecture that predicts a posterior for MOS given a speech clip. It's a reduced-size model trained end-to-end towards MOS labels by maximizing the likelihood. With several architectural improvements based on DNSMOS, we achieve a better performance compared to DeePMOS on several datasets, but with less than 6% of the number of parameters. We call our work **DNSMOS Pro**. The architecture and training design are simple and general: training can be done on any speech quality dataset that has speech quality labels, without the need for individual raters' scores.



(a) DNSMOS architecture. (b) DNSMOS Pro architecture.

Figure 2: *Architectures of DNSMOS and DNSMOS Pro.*

The proposed DNSMOS Pro is evaluated using VCC2018, BVCC, and NISQA datasets [14, 15, 16] for a comprehensive evaluation. We implement and compare with other MOS-only DNN-based non-intrusive SQA models developed for $16\,kHz$ sample rate input, namely DNSMOS, MOSNet, and DeePMOS. To the best of the authors' knowledge, this is the first paper that compares these models vis-à-vis using several datasets. Results show DNSMOS Pro has significantly better performance compared to these other MOS-only trained models. It has competitive performance compared to end-to-end listener-dependent speech quality methods.

2. Method

2.1. Problem formulation

Let $\mathbf{x}$ denote the features of a speech clip, and y denote the corresponding MOS of the speech clip. A (non-intrusive) speech quality dataset is given by $\mathcal{D} = \{(\mathbf{x}_n\, y_n)\}_{n=1}^{N}$, where N is the total number of speech clips.

Instead of modeling the point estimate y directly, we choose to estimate the posterior of the MOS for the speech clip $\mathbf{x}$. This means we are interested in

$$p_{\boldsymbol{\psi}}(y|\mathbf{x}), \tag{1}$$

where $\boldsymbol{\psi}$ are the parameters of the posterior distribution. Following DeePMOS [13], we model the posterior as a Gaussian motivated by the analytical tractability. This means the problem is to estimate $\boldsymbol{\psi}(\mathbf{x}) = (\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$, since $p_{\boldsymbol{\psi}}(y|\mathbf{x}) = \mathcal{N}(y; \mu(\mathbf{x}), \sigma^2(\mathbf{x}))$.

Consider a DNN as a regression function $f_{\boldsymbol{\theta}}(\mathbf{x})$, with parameters $\boldsymbol{\theta}$, that maps to the Gaussian parameters $\boldsymbol{\psi}(\mathbf{x})$; that is,

$$\boldsymbol{\psi}(\mathbf{x}) = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}). \tag{2}$$

Thus, we can train $\mathbf{f}_{\boldsymbol{\theta}}$ in a maximum-likelihood manner using the dataset $\mathcal{D}$. The optimization problem formulation is given by

$$\arg\max_{\boldsymbol{\theta}} \log \prod_{n=1}^{N} p_{\boldsymbol{\psi}}(y_n|\mathbf{x}) = \mathcal{N}(y_n; \boldsymbol{\psi}(\mathbf{x}_n) = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_n)). \tag{3}$$

2.2. DNSMOS Pro architecture

The DNSMOS Pro architecture is based on the DNSMOS architecture. It takes the log-magnitude spectrogram of a speech clip as input, and outputs two scalar values; the mean and the variance of the posterior distribution of MOS modeled as a Gaussian. DNSMOS Pro consists of three major components: an encoder, a global max pooling layer, and a head. The encoder mainly consists of convolutional layers. The global max pooling layer is an operation that takes the maximum value along the time and frequency axis. The head maps the output of the global max pooling layer to a 2-dimensional vector. Fig. 2 visualizes the architecture of DNSMOS and DNSMOS Pro.

The encoder consists of 4 convolutional layers with batch normalization and ReLU activation function. The head consists of 3 dense layers and a 2-dimensional vector is predicted. The two values can be seen as the mean and variance of a normal distribution. The motivation for this architectural design is given in an ablation study in the Experimental section.

Further, we do a linear transformation of the output distribution of the prediction head. Let $\boldsymbol{\psi}'(\mathbf{x}) = (h_1, h_2)$ be the output of the head. To map the parameters to a normal distribution $\mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$, we apply a transform module as

$$\mu(\mathbf{x}) = 2h_1 + 3, \tag{4}$$

$$\sigma^2(\mathbf{x}) = 4\text{Softplus}(h_2). \tag{5}$$

Equivalently, this is a transformation of the target label from the domain $[1, 5]$ onto $[-1, 1]$, given by $x \mapsto (x-3)/2$ (including the Softplus in both cases).

The reason for using this transformation is due to problems with training bias when having labels not symmetric around zero [17]; DNSMOS Pro is not an unbiased estimator of MOS if we do not map the labels onto $[-1, 1]$. We will explore this mapping in an ablation study, and justify its use.

2.3. DNSMOS Pro Training

Consider a speech quality dataset $\mathcal{D} = \{(\mathbf{x}_n\, y_n)\}_{n=1}^{N}$, where $\mathbf{x}_n$ are the features of speech clip n and y_n is the MOS score of the speech clip. Training is done on the dataset to maximize the log-likelihood given in (3). Since $\tilde{y}_n$ is modeled by a Gaussian, we have

$$p_{\boldsymbol{\psi}}(y_n|\mathbf{x}_n) = \frac{1}{\hat{\sigma}(\mathbf{x}_n)\sqrt{2\pi}} exp\{-\frac{1}{2\hat{\sigma}^2(\mathbf{x}_n)}(\hat{\mu}(\mathbf{x}_n) - y_n)^2\}, \tag{6}$$

where $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) = (\hat{\mu}(\mathbf{x}_n), \hat{\sigma}^2(\mathbf{x}_n))$. Thus, maximizing the log-likelihood is equivalent to minimizing the Gaussian negative log-likelihood (GNLL);

$$\arg\min_{\boldsymbol{\theta}} \sum_{n=1}^{N} \frac{1}{2}\left[\log \hat{\sigma}(\mathbf{x}_n)^2 + \frac{(\hat{\mu}(\mathbf{x}_n) - y_n)^2}{\hat{\sigma}^2(\mathbf{x}_n)}\right]. \tag{7}$$

The training objective is to minimize the GNLL.

2.4. DNSMOS Pro Inference

Due to the Bayesian problem formulation and the architectural design, DNSMOS Pro naturally predicts a Gaussian distribution of the MOS given a speech clip. For each speech clip feature $\mathbf{x}_n$, DNSMOS Pro predicts $\hat{\mu}(\mathbf{x}_n)$, $\hat{\sigma}^2(\mathbf{x}_n)$.

For point estimate, we use the maximum likelihood estimator of y_n, given by

$$\hat{y}_n = \max_{y_n} p(y_n|\hat{\mu}(\mathbf{x}_n), \hat{\sigma}^2(\mathbf{x}_n)) = \hat{\mu}(\mathbf{x}_n). \tag{8}$$

Table 1: *Performance results of DNSMOS Pro to other methods. For each dataset and model, the mean and standard deviation of a measure are presented. The bold-faced values are the best scores in each performance measure when comparing the MOS-only methods.*

Model	Model size	VCC2018 MSE	VCC2018 LCC	VCC2018 SRCC	BVCC MSE	BVCC LCC	BVCC SRCC	NISQA simulated MSE	NISQA simulated LCC	NISQA simulated SRCC
Results quoted from literature.										
MBNet [3]	1.38M	0.955	0.658	0.630	0.669	0.757	0.765	-	-	-
LDNet [3]	1.48M	0.432	0.676	0.641	0.324	0.794	0.790	-	-	-
LaMOSNet [4]	1.39M	0.432	0.687	0.656	-	-	-	-	-	-
Simulated in our experiments. 10 runs each.										
MOSNet [1]	1.18M	0.490±0.014	0.632±0.010	0.603±0.008	0.551±0.066	0.611±0.058	0.602±0.061	0.592±0.113	0.719±0.058	0.703±0.063
DeePMOS [13]	1.31M	0.504±0.050	0.661±0.013	0.627±0.013	0.602±0.183	0.759±0.026	0.758±0.023	0.477±0.202	0.830±0.031	0.837±0.017
DNSMOS [12]	0.06M	1.141±0.141	0.655±0.006	0.622±0.005	1.186±0.172	0.750±0.014	0.748±0.015	0.901±0.163	0.860±0.008	0.854±0.008
DNSMOS Pro	0.07M	**0.441**±0.011	**0.677**±0.002	**0.641**±0.003	**0.338**±0.024	**0.787**±0.015	**0.783**±0.015	**0.379**±0.051	**0.866**±0.006	**0.865**±0.008

3. Experiments

In this section, we implement DNSMOS Pro[1] and train on several datasets respectively, and compare to existing methods. We do not include SSL models or pre-training methods for fairness of evaluation. We do not include the NISQA model since it is developed for $48\,kHz$ sample rate data. We also omit MOSLight since the training design is tailored to different voice conversion datasets, and thus is not a 'general' method.

3.1. Datasets

We will consider three datasets, the Voice Conversion Challenge 2018 (VCC2018) [14], the VoiceMOS Challenge 2022 (BVCC) [15], and the NISQA simulated dataset [16]. Both VCC2018 and BVCC are popular benchmark datasets partly because they include individual raters' scores. The NISQA simulated dataset only includes the MOS scores, which makes listener-dependent methods unusable here. We train and evaluate on the three datasets respectively. We measure the performance on predicting the MOS per speech clip.

The VCC2018 dataset consists of 20 580 speech clips collected from 38 voice conversion systems. Every speech clip was rated by at most 4 raters. We use the same split as done in [4, 13], that of 13 580, 3 000, and 4 000 for train, validate and test respectively.

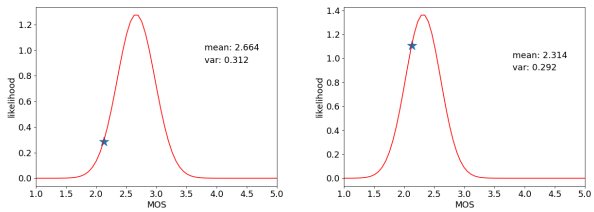
The BVCC dataset combines multiple datasets from past BCs, VCCs, and TTS systems in ESPNet [18, 19]. The dataset consists of 7106 speech clips, and each speech clip is rated by 8 raters. The dataset has a predefined split of 4 974, 1 066, and 1 066 for train, validate and test.

The NISQA simulated dataset consists of 12 500 speech clips generated from simulated distortions on clean speech data. Distortions include additive noise, low-pass filtering, and codec artifacts, etc. Approximately 5 raters have rated each speech clip, and only the MOS values are given. The dataset has a predefined split of 10 000 and 2 500 for train and validate respectively. We split the validate dataset into two equally sized sets; one for validation and one for testing. This means we have 10 000, 1 250, and 1 250 for train, validate and test.

3.2. Feature Extraction

The feature extraction is the same on all datasets. All speech clips are downsampled to $16\,kHz$ as done in most previous works [1, 2, 3, 4, 13]. Further, repetitive padding to $10\,s$ is used.

[1] https://github.com/fcumlin/DNSMOSPro



(a) DeePMOS (b) DNSMOS Pro

Figure 3: *Example of DNSMOS Pro posteriors vs DeePMOS on BVCC test data.*

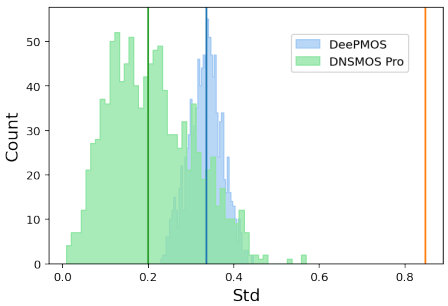
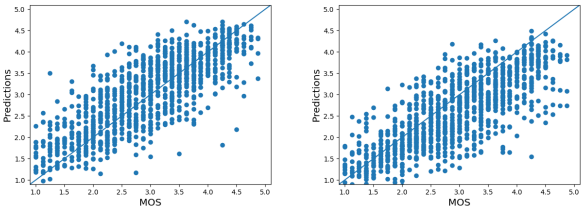


Figure 4: *Histogram of standard deviation predictions of the posteriors for DNSMOS Pro and DeePMOS respectively, on BVCC test data. The green and blue line are the median of the respective distribution. The orange line is the prior standard deviation, $\sigma = 0.847$.*

Previous experiments suggest that repetitive padding improves stability in training when using batch normalization in the architecture [2]. For simplicity, the padding step is considered as a preprocessing step, and the same for all datasets.

DNSMOS Pro takes log-magnitude spectrograms as input. We use a window duration of $20\,ms$ and a hop duration of $10\,ms$, using a Hann window. The magnitude is taken on the signals, and then the natural logarithm. For stability during training, we clip the values to the interval $[-7, 7]$.



(a) DNSMOS Pro (b) DNSMOS Pro w/o transformation

Figure 5: *DNSMOS Pro predictions vs MOS on BVCC test data.*

3.3. Results

DNSMOS Pro is trained for 500 epochs using the Adam optimizer with a learning rate of 10^{-4}, with moving average parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ [20]. MOSNet, DeePMOS, and DNSMOS were implemented and trained according to the literature [1, 13, 12]. We did not employ offline teacher learning for DNSMOS. For all model and training configurations, we did model selection based on the highest Linear Correlation Coefficient [21] on the validation data.

We compare point estimates using standard performance measures, namely mean square error (MSE), linear correlation coefficient (LCC) [21], and Spearman rank correlation coefficient (SRCC) [22]. Given a model and dataset, we report the average performance measure for 10 runs. We also report the standard deviation of the 10 results to show the variability of the measure. The result is shown in Table 1.

As can be seen, DNSMOS and DNSMOS Pro have good performance compared to MOSNet and DeePMOS, despite being more than 15 times smaller. However, DNSMOS exhibits some problems with properly mapping the underlying data. It has a negative bias w.r.t. to the training data, a problem we initially experienced with DNSMOS Pro. The transformation module in the architecture solves the problem, which will be shown in the ablation study.

Also, it seems that MOSNet is most suited for the VCC2018 dataset; it performs significantly worse on BVCC and NISQA simulated than DNSMOS Pro. We believe this is a strength of DNSMOS Pro, an efficient architecture and training design that generalizes. A similar statement can be said about DeePMOS, which also performs well across datasets. However, DeePMOS is more complicated than DNSMOS Pro: it is around 19 times larger, uses stochastic gradient noise, and has an online teacher learning set-up. DNSMOS Pro has a simplified and reduced-size design with improved performance.

3.3.1. Posterior distribution results

As with DeePMOS, DNSMOS Pro naturally predicts a Gaussian posterior of the MOS given a speech clip. This is another strength of the model compared to MOSNet and DNSMOS. However, as outlined in [13], it remains unclear how to evaluate the performance in predicting a distribution due to the limited number of raters per speech clip in the datasets.

To visualize the difference in the posterior distributions, we provide a histogram plot over the standard deviations predicted by DeePMOS and DNSMOS Pro respectively, shown in Fig 4. It can be seen that DNSMOS Pro has on average a significantly lower standard deviation compared to DeePMOS. A lower standard deviation can be interpreted as higher confidence and thus more useful posteriors are predicted.

Table 2: *Ablation study of DNSMOS Pro. - means the removal of a component. + means the addition/change of a component. Bold values are the best values in the respective column.*

	BVCC		
DNSMOS Pro	MSE	LCC	SRCC
Regular	**0.338**$\pm$0.024	**0.787**$\pm$0.015	**0.783**$\pm$0.015
-Transform	0.466$\pm$0.067	0.781$\pm$0.010	0.777$\pm$0.012
-BN	0.382$\pm$0.029	0.752$\pm$0.016	0.747$\pm$0.016
+Depth	0.382$\pm$0.033	0.777$\pm$0.008	0.776$\pm$0.007
+Pool	0.471$\pm$0.150	0.764$\pm$0.024	0.760$\pm$0.025
+SiLU	0.365$\pm$0.027	0.779$\pm$0.012	0.778$\pm$0.013

3.3.2. Ablation study

To study the effect of different components of DNSMOS Pro, we conducted an ablation study on the BVCC dataset, which has the most ratings per speech clip and thus is expected to be of higher quality. Each configuration is trained 10 times and an averaged result with standard deviations is reported in Table 2. We list the modified component in the left column, including removing the transform component (but keeping Softplus, -Transform), removing batch normalization (-BN), adding an extra convolutional layer (+Depth), having the same pooling configuration as DNSMOS (+Pool), and replacing ReLU activation functions with SiLU (Swish) [23] activation (+SiLU).

We can see batch normalization and the current pooling setup contribute the most to the model performance. Adding an extra convolutional layer doesn't improve performance and the same for using SiLU activation.

Another important finding is when removing the transformation of the labels. Doing so results in an increase in the MSE and the reason is visualized in Fig. 5. Without the transformation, the target labels are distributed from 1 to 5 which causes a negative bias of the model. However, applying the transformation module, the labels can be seen as distributed from -1 to 1, which removes the negative bias.

4. Conclusion

In this work, we presented DNSMOS Pro, a probabilistic lightweight DNN-based non-intrusive speech quality model suitable for real-time communication systems. It is smaller than existing probabilistic models but has higher performance measures on several datasets. Compared to other end-to-end non-intrusive SQA methods, it has competitive performance despite its simplicity in training and small architectural design.

This study can be seen as a continuation study of the DNSMOS architecture, thus shedding light on the lineage of non-intrusive speech quality models that only use the MOS at the time of training. Such models are suitable for any subjectively rated dataset and are not limited by the need for individual ratings or the underlying processing system (such does not always exist). Future work is to better understand the posterior distribution. This is today not possible due to the limited data available.

Acknowledgements: The computations handling was enabled by the supercomputing resource Berzelius provided by National Supercomputer Centre at Linköping University and the Knut and Alice Wallenberg foundation.

5. References

[1] 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 *Interspeech 2019*, 09 2019, pp. 1541–1545.

[2] Y. Leng, X. Tan, S. Zhao, F. K. Soong, X. Li, and T. Qin, "MBNet: MOS prediction for synthesized speech with mean-bias network," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*. IEEE, 2021.

[3] W.-C. Huang, E. Cooper, J. Yamagishi, and T. Toda, "LDNet: Unified listener dependent modeling in MOS prediction for synthetic speech," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[4] F. Cumlin, C. Schüldt, and S. Chatterjee, "Latent-based neural net for non-intrusive speech quality assessment," in *2023 33th European Signal Processing Conference (EUSIPCO)*, ser. European Signal Processing Conference, sep 2023, pp. 36–40.

[5] Z. Li and W. Li, "MOSLight: A Lightweight Data-Efficient System for Non-Intrusive Speech Quality Assessment," in *Proc. INTERSPEECH 2023*, 2023, pp. 5386–5390.

[6] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[7] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[8] E. Cooper, W.-C. Huang, T. Toda, and J. Yamagishi, "Generalization ability of mos prediction networks," in *ICASSP 2022*. IEEE, 2022.

[9] 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*, 09 2022, pp. 4521–4525.

[10] A. Stan, "The ZevoMOS entry to VoiceMOS Challenge 2022," in *Proc. Interspeech 2022*, 2022, pp. 4516–4520. [Online]. Available: https://www.isca-speech.org/archive/pdfs/interspeech_2022/stan22_interspeech.pdf

[11] Z. Qi, X. Hu, W. Zhou, S. Li, H. Wu, J. Lu, and X. Xu, "Le-ssl-mos: Self-supervised learning mos prediction with listener enhancement," in *2023 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2023.

[12] C. Reddy, V. Gopal, and R. Cutler, "DNSMOS: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, 06 2021.

[13] X. Liang, F. Cumlin, C. Schüldt, and S. Chatterjee, "Deepmos: Deep posterior mean-opinion-score of speech," in *Interspeech 2023*. ISCA, Aug 2023.

[14] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," 2018.

[15] W.-C. Huang, E. Cooper, Y. Tsao, H.-M. Wang, T. Toda, and J. Yamagishi, "The voicemos challenge 2022," *arXiv preprint arXiv:2203.11389*, 2022.

[16] G. Mittag, B. Naderi, A. Chehadi, and S. Möller, "NISQA: A deep CNN-self-attention model for multidimensional speech quality prediction with crowdsourced datasets," in *Interspeech 2021*. ISCA, Aug 2021.

[17] C. Igel and S. Oehmcke, "Remember to correct the bias when using deep learning for regression!" *KI - Künstliche Intelligenz*, vol. 37, 04 2023.

[18] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[19] 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 *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2020, pp. 7654–7658.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] K. Pearson, "Notes on the history of correlation," *Biometrika*, vol. 13, no. 1, pp. 25–45, 1920.

[22] C. Spearman, "The proof and measurement of association between two things," *The American journal of psychology*, vol. 100, no. 3/4, pp. 441–471, 1987.

[23] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.