INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

DeePMOS: Deep Posterior Mean-Opinion-Score of Speech

Xinyu Liang[1], *Fredrik Cumlin*[2], *Christian Schüldt*[3], *Saikat Chatterjee*[1]

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

xinyulia@kth.se, fcumlin@gmail.com, cschuldt@google.com, sach@kth.se

Abstract

We propose a deep neural network (DNN) based method that provides a posterior distribution of mean-opinion-score (MOS) for an input speech signal. The DNN outputs parameters of the posterior, mainly the posterior's mean and variance. The proposed method is referred to as deep posterior MOS (DeePMOS). The relevant training data is inherently limited in size (limited number of labeled samples) and noisy due to the subjective nature of human listeners. For robust training of DeePMOS, we use a combination of maximum-likelihood learning, stochastic gradient noise, and a student-teacher learning setup. Using the mean of the posterior as a point estimate, we evaluate standard performance measures of the proposed DeePMOS. The results show comparable performance with existing DNN-based methods that only provide point estimates of the MOS. Then we provide an ablation study showing the importance of various components in DeePMOS.

Index Terms: Speech quality assessment, deep neural network, maximum-likelihood, voice conversion challenge.

1. Introduction

Deep neural network (DNN)-based (non-intrusive) speech quality assessment is a recent trend. Early works in this area include AutoMOS and QualityNet [1, 2]. AutoMOS uses a long-short-time-memory (LSTM) [3] network in its architecture and an end-to-end training approach of a speech clip and its mean-opinion-score (MOS). QualityNet uses a bi-directional LSTM and uses an end-to-end training approach for a speech clip and its perceptual evaluation of speech quality (PESQ) score [4].

Contemporary works include DNSMOS, NISQA, and MOSNet [5, 6, 7], all using the observed MOS as target output. DNSMOS regularizes possible biases in the MOS score using student-teacher networks [5]. NISQA uses attention [6], an idea obtained from much cited [8]. MOSNet investigated the effect on prediction with respect to architectural designs and training parameters and found that a convolutional neural network (CNN) together with a bi-directional-LSTM (BLSTM) had the best performance [7].

Choi et. al. (2020, 2021) proposed two variants of MOSNet [9, 10], one based on the idea of Global Style Tokens (GST) [11], and the other a multi-task learning approach, with the objective to predict the speech quality and spoofing detection (i.e., identifying real or synthesized speech). They show that GST improves performance, and that multi-task learning has a significant effect on speech quality assessment [10].

There are also DNN-based methods that are conditioned on the identity of a human listener at the time of training. Examples are mean-bias network (MBNet) [12] and listener dependent network (LDNet) [13].

Motivation: All the aforementioned DNN-based methods provide a point estimate of the MOS for an input speech signal. In this paper, our main contribution is to provide a DNN-based posterior distribution estimate of the MOS. To the best of our knowledge, DNN-based posterior of MOS was not considered in the literature. We call the proposed method DeePMOS (Deep Posterior MOS), which provides confidence in terms of standard deviation (or spread) across the posterior mean. The posterior mean can be treated as a point estimate of MOS.

Contributions: We formulate a maximum-likelihood based optimization problem to train DeePMOS. Among many types of DNNs, we use a recursive neural network (RNN), called bi-directional LSTM, to handle variable lengths of speech signals. In our DNN architecture of DeepMOS, we use a suitable combination of bi-directional LSTM and convolutional layers, motivated by MBNet. In this context, we mention that the work of [14] provides a posterior MOS using logistic regression and traditional model-driven speech features. On the other hand, DNN-based methods, including the proposed DeePMOS, exploit a complex non-linear relationship between MOS and speech spectrogram by using data-driven features.

To train DeePMOS, we address two important aspects that are inherently present in relevant training datasets collected through crowd-sourcing. The aspects are limited-data and noisy-data. The *limited-data* aspect arises because each speech clip is labeled with scores from a limited number of judges among many judges. The reason for limited judges is simple - human judge-based labeling is costly. Due to the limited-data aspect, it is hard to estimate the true MOS as a mean (average) of scores from the limited number of judges and use in training. On the other hand, the *noisy-data* aspect arises due to human judges being noisy by nature; for the same speech clip judges may provide different scores. Even the same judge may provide different scores to a speech clip at different times; scores vary due to mood, work pressure, the nature of a person, etc. A judge can be optimistic or pessimistic by nature. Further, the set of judges may vary across clips in a training dataset.

We address the limited-data and noisy-data aspects in the training of DeePMOS using stochastic gradient noise and a student-teacher learning setup, mainly motivated by their success in semi-supervised image classification [15]. We also provide an ablation study to show the importance of various components in the training of the proposed method.

2. DeePMOS method

2.1. Problem formulation

Let $\mathbf{x}$ denote the features of a speech clip, and y denote the MOS of the speech clip. The task is to estimate the posterior of

10.21437/Interspeech.2023-1436

the MOS for the speech clip $\mathbf{x}$, as

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

where $\boldsymbol{\psi}$ are the parameters of the posterior distribution.

In this article we assume that the posterior is a Gaussian distribution, motivated by analytical tractability in training, mainly solving the relevant optimization problem discussed later in this section. A Gaussian distribution is fully described by its mean and variance, i.e. $p_{\boldsymbol{\psi}}(y|\mathbf{x}) = \mathcal{N}(y; \mu(\mathbf{x}), \sigma^2(\mathbf{x})) = \mathcal{N}(y; \boldsymbol{\psi}(\mathbf{x}))$, where $\boldsymbol{\psi}(\mathbf{x}) = \{\mu(\mathbf{x}), \sigma^2(\mathbf{x})\}$.

We use a DNN as a regression function $f_{\boldsymbol{\theta}}(\mathbf{x})$ that outputs $\boldsymbol{\psi}(\mathbf{x})$, as

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

The DNN is a '*MOS-posterior-parameter providing regression function*' with parameters $\boldsymbol{\theta}$. Using a training dataset comprised of many pair-wise $(\mathbf{x}, y)$ examples, we could train the parameters $\boldsymbol{\theta}$ of the regression function (DNN) by optimizing an appropriate function in a maximum-likelihood manner.

Let N be the number of speech clips in the dataset and J be the number of judges. We denote the n'th clip's features by $\mathbf{x}_n$. Further, let $\mathcal{J} = \{1, 2, \ldots, J\}$ be the set of identity of the judges. For each clip, a subset of all the judges provides scores. We denote the subset of judges $\mathcal{J}_n \subset \mathcal{J}$ for the n'th clip; that means the indices of the judges who scored the n'th clip are kept in the set $\mathcal{J}_n$. The subset $\mathcal{J}_n$ varies across clips, and $\cup_{n=1}^{N} \mathcal{J}_n = \mathcal{J}$. Let $s_{\mathbf{x}}^j$ denote the score of the j'th judge for the speech clip's features $\mathbf{x}$, where $j \in \mathcal{J}$. Now, let the set of scores for the n'th clip be denoted by $\mathcal{S}_n = \{s_{\mathbf{x}_n}^{(j)}; (j) \in \mathcal{J}_n\}$; here $s_{\mathbf{x}_n}^{(j)}$ is the score of the n'th clip by (j)'th judge of the subset $\mathcal{J}_n$. The dataset available to us is $\mathcal{D} = \{(\mathbf{x}_n, \mathcal{S}_n, \mathcal{J}_n)\}_{n=1}^{N}$.

In our problem setup, $|\mathcal{J}_n| \triangleq |\mathcal{S}_n|$ is small, where $|.|$ denotes the cardinality of a set. That means each clip has few scores. This is the *limited-data* aspect of the dataset $\mathcal{D}$. For example, the dataset we for our experiments has at most 4 scores per clip, meaning $|\mathcal{S}_n| \leq 4$, for all n. Moreover, the scores $s_n^{(j)}$ are noisy due to human nature, leading to the *noisy-data* aspect. Let the true MOS of the n'th clip be denoted by y_n, which is unknown. Then a standard estimation of y_n is an average over the available scores, i.e.

$$\tilde{y}_n = \frac{1}{|\mathcal{S}_n|} \sum_{s_{\mathbf{x}_n}^{(j)} \in \mathcal{S}_n} s_{\mathbf{x}_n}^{(j)}. \tag{3}$$

Due to the two mentioned aspects, the estimate $\tilde{y}_n$ is expected to be noisy.

Using the dataset $\mathcal{D}$, we can create a new dataset $\mathcal{D}_1 = \{(\mathbf{x}_n, \tilde{y}_n)\}_{n=1}^{N}$, and then use the new dataset for an end-to-end training of a suitable regression function $f_{\boldsymbol{\theta}}(\mathbf{x})$. The regression function can be realized using a suitable DNN. The parameters $\boldsymbol{\theta}$ of DNN can be learned by maximizing the likelihood function

$$\arg\max_{\boldsymbol{\theta}} \log \prod_{n=1}^{N} p_{\boldsymbol{\psi}}(y = \tilde{y}_n|\mathbf{x}_n) = \mathcal{N}(\tilde{y}_n; \boldsymbol{\psi}(\mathbf{x}_n) = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_n)). \tag{4}$$

2.2. DeePMOS Architecture

DeePMOS uses a spectrogram as input. The DNN architecture of DeePMOS is similar to the MeanNet used in MBNet [12], except that instead of having only a MOS prediction $\hat{y}$ we use two prediction heads in our design: one predicts a mean estimate $\hat{\mu}_y$ and the other predicts a non-negative variance estimate $\hat{\sigma}_y^2$, to describe the posterior MOS with a Gaussian distribution. This architectural design is motivated by the field of spatiotemporal attention [16], where an LSTM has been used for temporal attention. The architecture of DeePMOS can be conceptually viewed as a special case of the Mixture Density Network [17, 18], here only using one Gaussian for the output distribution, but exploiting the power of modern DNNs.

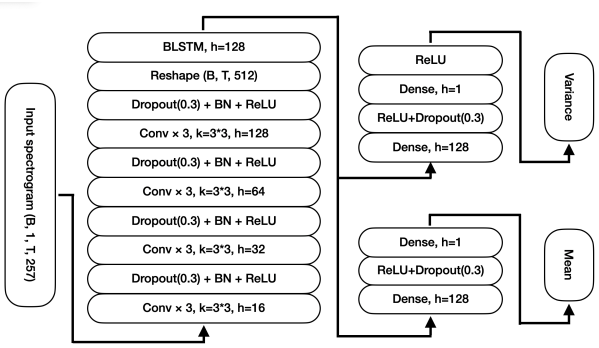

Figure 1: *DeePMOS Architecture*

Fig. 1 shows the layer structure of DeePMOS, whose main components are 12 convolutional layers followed by a bi-directional LSTM. The prediction head consists of two separate networks, each having 2 fully-connected layers, and the one for variance prediction has an extra ReLU [19] activation function to make the prediction non-negative.

The model is designed to take 257-dimensional spectrogram features $\mathbf{x}_n$ of any length T as input, and two scalar sequences $\hat{\mu}_T(\mathbf{x}_n)$, $\hat{\sigma}_T^2(\mathbf{x}_n)$ of the same length will be generated. The estimated mean $\hat{\mu}(\mathbf{x}_n)$ and variance $\hat{\sigma}^2(\mathbf{x}_n)$ for the whole clip is then obtained by averaging over the sequence length.

2.3. DeePMOS Training

At training time, the main objective is to maximize the log-likelihood function in (4). Since we fit $\tilde{y}_n$ with a Gaussian distribution, this gives

$$p_{\boldsymbol{\psi}}(y = \tilde{y}_n|\mathbf{x}_n) = \frac{1}{\hat{\sigma}(\mathbf{x}_n)\sqrt{2\pi}} e^{-\frac{1}{2\hat{\sigma}^2(\mathbf{x}_n)}(\hat{\mu}(\mathbf{x}_n) - \tilde{y}_n)^2}, \tag{5}$$

which turns the maximization of the likelihood into minimizing the Gaussian negative log-likelihood loss (GNLL loss)

$$\arg\min_{\boldsymbol{\theta}} \sum_{n=1}^{N} \frac{1}{2} \left[\log \hat{\sigma}(\mathbf{x}_n)^2 + \frac{(\hat{\mu}(\mathbf{x}_n) - \tilde{y}_n)^2}{\hat{\sigma}^2(\mathbf{x}_n)} \right]. \tag{6}$$

Due to the noisy nature of the labels $\tilde{y}_n$, we introduce stochastic gradient noise (SGN) [20] in training. Instead of directly using $\tilde{y}_n$ as the targets, we perturb these values with Gaussian noise. At each iteration, we draw an independent random Gaussian noise sample $z \in \mathcal{N}(0, \sigma_z^2)$ for each clip and train on the new labels

$$\tilde{y}_n^z = \tilde{y}_n + z. \tag{7}$$

We use $\sigma_z^2 = 0.01$ by default, which is chosen from a small hyperparameter-tuning experiment on the validation set.

Another concept we used in our training is the mean-teacher approach [15], a student-teacher framework. The reason for using it is to provide robust learning in presence of noisy labels. We apply it in the following manner. First, two DeePMOS

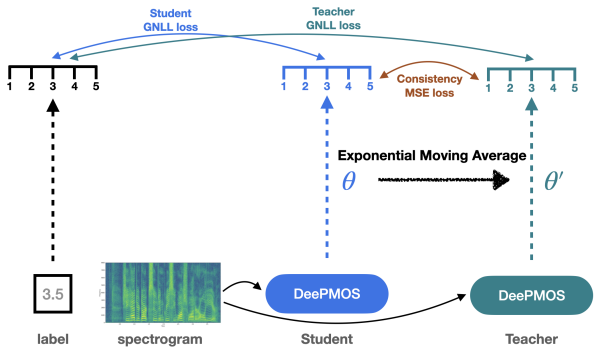

Figure 2: *Illustration of the teacher-student setup.*

models are initialized, with one called the teacher model $\mathbf{f}_{\boldsymbol{\theta}'}(\cdot)$ and another being the student model $\mathbf{f}_{\boldsymbol{\theta}}(\cdot)$. At initialization, the teacher model would copy the student model's parameters. Then, during training time the student model does normal back-propagation to update the parameters $\boldsymbol{\theta}$, while the teacher model also does normal back-propagation but also updates its parameters $\boldsymbol{\theta}'$ with respect to an exponential moving average (EMA) of the student's parameters. That is, after each training batch, we let

$$\boldsymbol{\theta}' = \alpha\boldsymbol{\theta}' + (1-\alpha)\boldsymbol{\theta}, \tag{8}$$

where $0 \leq \alpha \leq 1$ is a hyperparameter. In this way, the teacher model acts as an ensemble of the student model at different time stamps [15], with the purpose of increasing robustness to noisy labels. Figure 2 shows an illustration of the setup.

We design our overall training loss function as a weighted sum of several components

$$L(\boldsymbol{\theta}, \boldsymbol{\theta}') = L_s + \lambda_t L_t + \lambda_c L_c, \tag{9}$$

where L_s is the GNLL loss for the student model, L_t is the GNLL loss for the teacher model, and L_c is a consistency loss. The consistency loss L_c is defined as

$$L_c = \sum_{n=1}^{N} \|f_{\boldsymbol{\theta}}(\mathbf{x}_n) - f_{\boldsymbol{\theta}'}(\mathbf{x}_n)\|^2, \tag{10}$$

which is the mean square error (MSE) loss between the outputs of the student model and the teacher model. We followed the design in MBNet to use per-frame losses by replicating the target $\tilde{y}_n$ to the same length of the model output, which is the number of frames in the input feature $\mathbf{x}_n$. Here λ_t and λ_c are both hyperparameters, chosen by cross-validation.

2.4. DeePMOS Inference

At inference time, our model naturally predicts the distribution of MOS y_n for each input feature $\mathbf{x}_n$ as a Gaussian distribution with parameters $\hat{\mu}(\mathbf{x}_n), \hat{\sigma}^2(\mathbf{x}_n)$ obtained from the network output. If a point estimate is required, a maximum likelihood estimator of y_n is obtained by

$$\hat{y}_n = \max_{y_n} p(y_n|\hat{\mu}(\mathbf{x}_n), \hat{\sigma}^2(\mathbf{x}_n)) = \hat{\mu}(\mathbf{x}_n), \tag{11}$$

which is the predicted mean value for the Gaussian distribution. We use this point estimate later in comparison with other DNN-based methods.

Table 1: *25%−, 50%−, and 75%−quantiles of the likelihood on the test data, using the prior and posterior distributions.*

	25%-quantile	Median	75%-quantile
Prior	0.210	0.343	0.421
Posterior	0.158	0.426	0.654

Table 2: *Comparison of DeePMOS with other methods. Boldface numbers highlight the best value in the respective column.*

	Utterance-level			System-level		
Model	MSE	LCC	SRCC	MSE	LCC	SRCC
Not simulated. Results quoted from literature.						
MOSNet [10, 9]	0.448	0.651	0.619	0.039	0.966	0.924
Simulated in our experiments.						
MBNet [12]	0.713	0.662	0.632	0.309	0.943	0.943
LDNet [13]	**0.428**	**0.680**	**0.644**	**0.023**	**0.984**	**0.963**
DeePMOS	0.497	0.662	0.628	0.055	0.981	**0.963**

3. Experiments

In this section we evaluate DeePMOS [1] and compare it with several existing methods, using appropriate datasets and performance measures.

3.1. Datasets

We use Voice Conversion Challenge 2018 (VCC2018) [21] dataset as a benchmark. This standard dataset consists of 20 580 speech clips collected from 38 different voice conversion (VC) systems. Every speech clip was rated by at most 4 judges on a scale of 1 to 5 at discrete values based on the assessment of the spoken words, and in our case, we use the average of the scores as the training target $\tilde{y}_n$.

We split the total dataset into training, validation, and test sets containing 13 580, 3 000, and 4 000 speech clips, respectively. The model performance to predict the point estimate is evaluated on two levels - the utterance level and system level, as per [7, 12, 13]. On the utterance level, the point estimate is given by $\hat{\mu}(\mathbf{x}_n)$, and we measure the performance per speech clip using suitable performance measures. On the system level, the performance is about predicting the average of the speech quality for *all* speech clips belonging to a VC system. That is, the 38 systems partition the data, and the model point estimate prediction is given by the average prediction over a partitioned set. The performance is then measured on the 38 scores (one per system): the average predicted score to the average of the observed MOS scores per VC system, using appropriate performance measures.

3.2. Feature Extraction

The speech clips were downsampled to $16\,kHz$, and we used a spectrogram of it as feature input. In the spectrogram computations, we used $32\,ms$ window length and $8\,ms$ window shift. Each signal was preprocessed with repetitive padding to the longest signal duration, in order to stabilize computation in batch normalization [22] layers, as per [12].

[1] https://github.com/Hope-Liang/DeePMOS

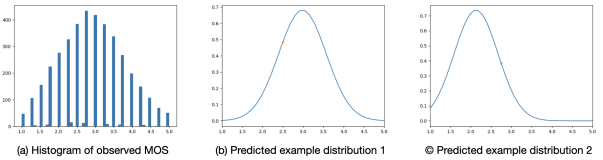

Figure 3: *Prior and posterior distributions of MOS. (a) Histogram of the MOS with mean* 2.902 *and standard deviation* 0.888*. This is a prior distribution. (b) Posterior of MOS for a speech clip, with mean* 2.984 *and standard deviation* 0.585*. (c) Posterior of MOS for another speech clip, with mean* 2.131 *and standard deviation* 0.540*.*

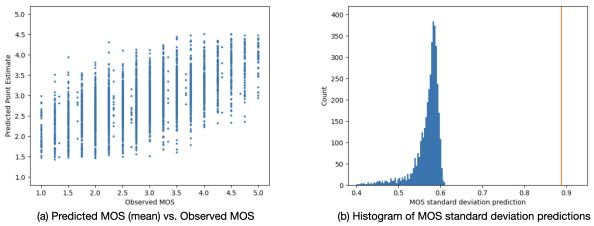

Figure 4: *Visualization of prediction results. (a) Scatter plot between* $\tilde{y}$ *and* $\hat{\mu}(\mathbf{x})$*. (b) Histogram of predicted standard deviation* $\hat{\sigma}(\mathbf{x})$ *against the prior standard deviation 0.888.*

3.3. Results

We trained DeePMOS on VCC2018 for 60 epochs with Adam [23] optimizer, a learning rate of 10^{-4}, weight decay of 10^{-5}, and dropout of 30%. For the loss function in Equation (9), we selected the hyperparameters $\lambda_t = 1$ and $\lambda_c = 0.5$ through cross-validation. The teacher model used $\alpha = 0.99$ for the first five epochs in the training phase since the student model has a faster learning curve in the beginning, and after the five epochs, we used $\alpha = 0.999$. The model selected was the teacher model with the highest linear correlation coefficient (LCC) on validation data, where testing on validation data was performed after each epoch. We trained the model using a single Nvidia A100 40GB GPU card and it takes roughly 4 hours per train.

We first examine how the posterior distribution of MOS looks against the prior. In Fig. 3(a), a histogram of $\tilde{y}$ for the test set is plotted, which provides an idea of the prior of MOS. Fig. 3 (b) and (c) provide the predicted Gaussian posterior for two randomly picked speech clips. We found that the posterior standard deviation is smaller than the prior standard deviation.

We now visualize prediction results for the test set. In Fig. 4(a), we show a scatter plot to demonstrate correlations between the observed MOS $\tilde{y}$ and its point estimate $\hat{y} = \hat{\mu}(\mathbf{x})$. As each speech clip has scores from at most 4 judges, and the observed MOS $\tilde{y}$ shows discrete nature due to the finite set of values in the interval $[1, 5]$. Then, in Fig. 4(b), we show a histogram of the posterior's standard deviation $\hat{\sigma}(\mathbf{x})$. In the same plot, we show the standard deviation of the prior, which is 0.888. Note that the posterior standard deviation is less than the prior standard deviation.

In order to quantify the performance of DeePMOS in the probabilistic sense, we compute the likelihood on the test data, using the prior distribution and the posterior distributions. We use a Gaussian prior, where the parameters are inferred using maximum likelihood given the MOS data. The result is shown in Table 1. The median and $75\%-$quantile, the posterior distribution given speech clip increases the likelihood compared to using the prior, which means DeePMOS provides a useful posterior.

Table 3: *Ablation study of DeePMOS, with the* removed *component listed in the leftmost column. Boldface numbers highlight the best value in the respective column.*

	Utterance-level			System-level		
DeePMOS	MSE	LCC	SRCC	MSE	LCC	SRCC
Normal	**0.497**	0.662	0.628	**0.055**	**0.981**	**0.963**
- SGN	0.517	**0.667**	**0.631**	0.084	**0.981**	0.950
- teacher	0.670	0.646	0.614	0.220	0.976	0.946
- L_c	0.715	0.641	0.605	0.284	0.968	0.929
- L_t	0.528	0.658	0.624	0.076	0.980	0.934
- L_c, L_t	0.594	0.650	0.616	0.166	0.968	0.941

3.3.1. *Comparison with other methods*

To compare with other methods like MOSNet [7] and LDNet [13], we have to use the point estimate-based standard performance measures. Three widely referred performance measures in this area are the mean-square-error (MSE), linear-correlation-coefficient (LCC) [24], and Spearman's-rank-correlation-coefficient (SRCC) [25].

In our experiments, while we have used all three performance measures stated above, relatively higher importance was given to LCC and SRCC measures following [12]. The comparison results are given in Table 2, with the best performance scores marked in bold font. We see that our DeePMOS outperformed basic MOSNet and MBNet in terms of almost all performance measures, meanwhile giving a comparable performance with the state-of-the-art LDNet model on both utterance and system levels.

3.3.2. *Ablation study*

To investigate the effect of different components used in DeePMOS, we conducted an ablation study. We trained the model under each configuration six times on the VCC2018 dataset and took the average of the performance measures for each run. The results are reported in Table 3, with the removed component listed to the left.

We found that the teacher-student method and the consistency loss part are important for boosting performance, and the teacher model loss also helped find a better optimum. The existence of SGN on the labels stabilized the training by reducing the variance in the performance measures. Although it gives a slight decrease in the utterance-level LCC and SRCC, it also improved the system-level performance to a notable extent.

4. Conclusion

The approach of using appropriate distributions as posterior, such as Gaussian, which have few parameters is good where DNNs can provide the parameters. Learning of DNN parameters can be formulated using the maximum-likelihood based optimization principle. It is also important to use appropriate training and perform an ablation study to show the importance of participating components. Overall, DeePMOS is highly competitive vis-a-vis other DNN-based MOS prediction methods (which only provide point estimates), in addition to giving an interpretable posterior.

5. References

[1] B. Patton, Y. Agiomyrgiannakis, M. Terry, K. Wilson, R. A. Saurous, and D. Sculley, "AutoMOS: Learning a non-intrusive assessor of naturalness-of-speech," in *NIPS 2016 End-to-end Learning for Speech and Audio Processing Workshop*, 2016.

[2] S. Fu, Y. Tsao, H. Hwang, and H. Wang, "Quality-Net: An end-to-end non-intrusive speech quality assessment model based on BLSTM," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, B. Yegnanarayana, Ed. ISCA, 2018.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 2, 2001.

[5] 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.

[6] 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.

[7] 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.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[9] Y. Choi, Y. Jung, and H. Kim, "Deep MOS predictor for synthetic speech using cluster-based modeling," in *Interspeech 2020*. ISCA, oct 2020.

[10] ——, "Neural MOS prediction for synthesized speech using multi-task learning with spoofing detection and spoofing type classification," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021.

[11] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, Jul 2018.

[12] 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.

[13] 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.

[14] P. N. Petkov and W. B. Kleijn, "Probabilistic non-intrusive quality assessment of speech for bounded-scale preference scores," in *2010 Second International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 2010, pp. 188–193.

[15] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NIPS*, 2017.

[16] L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," *CoRR*, vol. abs/1603.08199, 2016. [Online]. Available: http://arxiv.org/abs/1603.08199

[17] C. Bishop, "Mixture density networks," Aston University, WorkingPaper, 1994.

[18] A. Graves, "Generating sequences with recurrent neural networks," *CoRR*, vol. abs/1308.0850, 2013. [Online]. Available: http://arxiv.org/abs/1308.0850

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[20] Y. Wu, R. Luo, C. Zhang, J. Wang, and Y. Yang, "Revisiting the characteristics of stochastic gradient noise and dynamics," 2021.

[21] 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.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

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

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

[25] 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.