Latent-based Neural Net for Non-intrusive Speech Quality Assessment

Fredrik Cumlin
KTH Royal Institute of Technology
Stockholm, Sweden
fcumlin@gmail.com

Christian Schüldt
Google LLC,
Stockholm, Sweden
schuldt@google.com

Saikat Chatterjee
Digital Futures and KTH Royal Institute of Technology
Stockholm, Sweden
sach@kth.se

***Abstract*—For non-intrusive speech quality assessment, we treat the mean-opinion-score (MOS) of a speech signal as a latent, and propose a latent MOS network (LaMOSNet) to estimate the MOS. At the time of training, the proposed LaMOSNet has two parts in series, with the first part providing the latent estimate, i.e. the MOS of an input speech signal, and the second part providing an estimated score by a given judge. Only the first part is used for testing. We address two inherent aspects - limited-data and noisy-data aspects - in training using stochastic gradient noise and a student-teacher type of training, motivated by semi-supervised learning. It is shown that LaMOSNet provides good performance on the Voice Conversion Challenge 2018 dataset, and state-of-the-art correlation performance on the Voice Conversion Challenge 2016 dataset.**

I. Introduction

Neural networks have commonly been used in automatic non-intrusive speech quality assessment for MOS prediction of an input speech signal [1]–[8]. Due to lack of clean references the training is challenging. A training dataset has speech clips marked with labels as scores from human judges using listening tests. A training dataset typically has two inherent aspects: 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. Due to the limited-data aspect, it is hard to estimate the 'true' MOS, which we here loosely define as an average of 'sufficiently many' scores from the entire human population, as an average of scores from the limited number of judges. On the other hand, the *noisy-data* aspect arises due to the fact that human judges are noisy by nature. For the same speech clip, judges typically provide different scores at different times.

In presence of the two inherent aspects (limited-data and noisy-data), our main contribution in this article is to propose a neural network architecture, called LaMOSNet, that treats the true MOS as a latent (hidden) variable and endeavors to estimate it.

Two important issues to develop LaMOSNet are: model architecture and model training. During the training phase, the LaMOSNet architecture has two neural networks in series. The first neural network provides an estimate of the true MOS for an input speech clip; that means, an estimate of the latent. Then the estimated MOS is used as input to the second neural network that outputs the score of an individual judge. The second neural network also uses the speech clip and a judge's identity. Therefore we can train LaMOSNet efficiently using the training dataset.

For LaMOSNet training, we propose a new cost function that has three main parts. The first part helps to provide an estimation of the true MOS (the latent). The second part uses the scores of judges per clip directly and handles the 'limited-data' aspect. Finally, the third part imposes consistency in training, like the cases of student-teacher networks applied in image classification tasks [9], [10]. We also use stochastic gradient noise-based training for robustness, which is successful in semi-supervised image classification tasks [9], [11], [12]. The use of student-teacher networks and stochastic gradient noise together addresses the 'noisy-data' aspect.

Relevant Literature: In this paper, the focus area is the use of neural networks, mainly deep neural networks (DNNs), for non-intrusive speech quality assessment. Contemporary works including DNSMOS, NISQA, and MOSNet [4]–[6], all use the observable MOS as target output. DNSMOS regularizes possible biases in MOS scores using student-teacher networks [5]. NISQA uses attention [6], an idea obtained from much-cited transformer network [13]. MOSNet investigated architectural designs and training parameters, and used convolutional neural network (CNN) together with bidirectional-LSTM [4]. There exist variants of MOSNet [14], [15], for examples, one based on global-style-tokens (GST) [16], and another based on a multi-task learning approach [15].

The proposed LaMOSNet has a conceptual resemblance with the mean-bias network (MBNet) [7]. At the time of training, MBNet uses two networks in parallel: the first network predicts MOS, and the second network provides a bias term, which when added to the MOS predicts a judge's score as a final outcome. Like LaMOSNet, the predicted MOS in MBNet can be interpreted as a latent variable. While MBNet uses a simple additive model of estimated MOS and bias to predict a judge's score, LaMOSNet uses a DNN to capture the complex non-linear relation between MOS, signal and judge. This is a major difference between LaMOSNet and MBNet. Finally, for a speech clip, the listener dependent network (LDNet) predicts the scores of all judges as an extrapolation task, and finally predicts the MOS score as the average of all judges' scores [8]. LaMOSNet does not perform the extrapolation task.

II. Problem Formulation

Let $\mathbf{x}$ denote the features of a speech clip and y the corresponding MOS. Let us consider a '*MOS-providing regression function*' $f_{\boldsymbol{\theta}}(\mathbf{x})$ that provides an estimate of the MOS as

$$\hat{y} = f_{\boldsymbol{\theta}}(\mathbf{x}), \tag{1}$$

where $\boldsymbol{\theta}$ denotes the set of parameters.

Dataset and inherent aspects: Let N be the number of speech clips and let J be the number of judges in the dataset. For the n'th clip we denote its features as $\mathbf{x}_n$. Further, denote a set for the identity of the judges as $\mathcal{J} = \{1, 2, \ldots, J\}$. For a speech clip n, only a subset $\mathcal{J}_n \subset \mathcal{J}$ provides scores. Let $s^j_{\mathbf{x}}$ denote the score of j'th judge for the speech clip's features $\mathbf{x}_n$, where $j \in \mathcal{J}$, and let $\mathcal{S}_n = \{s^{(j)}_{\mathbf{x}_n}; (j) \in \mathcal{J}_n\}$ be the corresponding set of scores. The dataset available to us is $\mathcal{D} = \{(\mathbf{x}_n, \mathcal{S}_n)\}_{n=1}^N$.

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 a few scores. This is the reason for the *limited-data* aspect of the dataset $\mathcal{D}$. For example, the dataset we will use later for our experiments has only 4 scores per clip, which means $|\mathcal{S}_n| = 4$, for all n. Moreover the scores $s^{(j)}_n$ 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 way of estimating y_n can be an average over the available scores, as follows

$$\tilde{y}_n = \frac{1}{|\mathcal{S}_n|} \sum_{s^{(j)}_{\mathbf{x}_n} \in \mathcal{S}_n} s^{(j)}_{\mathbf{x}_n}. \tag{2}$$

Due to the two aspects, the estimate $\tilde{y}_n$ is expected to be highly noisy and hence its use is not reliable.

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. Such an approach was used for MOSNet [4], DNSMOS [5], NISQA [6]. As $\tilde{y}_n$ is a highly noisy estimate of y_n, the approach has limitations, and may not generalize well for testdata.

The problem: We recognize that $\tilde{y}_n$ is a crude (noisy) estimation of y_n. Naturally, a question is: Can we use $\tilde{y}_n$ and $\mathcal{S}_n$ **together** to train a regression function to estimate y_n? This question motivates us to formulate the problem: how to use a new dataset $\mathcal{D}_2 = \{(\mathbf{x}_n, \mathcal{S}_n, \tilde{y}_n)\}_{n=1}^N$ and develop a regression function that is better than a regression function learned using the dataset $\mathcal{D}_1 = \{(\mathbf{x}_n, \tilde{y}_n)\}_{n=1}^N$. To address the problem, we develop a new method where the output of a regression function is treated as a suitable latent variable in the training phase.

III. LaMOSNet Architecture and Training

Our objective is to develop a MOS-providing regression method that can use judges' scores $s^{(j)}_{\mathbf{x}_n}$ in $\mathcal{S}_n$ at the time of training. That means we wish to use $\mathcal{D}_2$.

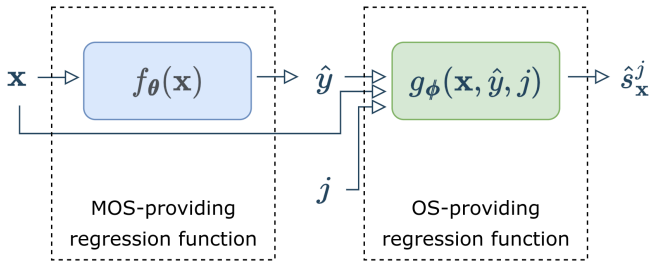

Fig. 1. Graphical illustration of the LaMOSNet architecture.

A. LaMOSNet Architecture in the training phase

In the training phase, the LaMOSNet architecture outputs two scores: a MOS estimate $\hat{y}$ and an estimate $\hat{s}^{(j)}_{\mathbf{x}_n}$ of $s^{(j)}_{\mathbf{x}_n}$. This facilitates the use of scores $s^{(j)}_{\mathbf{x}_n}$ in $\mathcal{S}_n$ efficiently at the time of training, in turn the use of $\mathcal{D}_2$.

The architecture is designed as follows. We have two neural networks that work in series. The first network is the '*MOS-providing regression function*' providing the estimate of the true latent MOS. The second network we call the '*OS-providing regression function*' and is denoted by $g_{\boldsymbol{\phi}}(\mathbf{x}_n, \hat{y}_n, j)$ given a judge j, where $\boldsymbol{\phi}$ are the parameters and $\hat{y}_n = f_{\boldsymbol{\theta}}(\mathbf{x})$. This means the LaMOSNet architecture is given by the two regression functions introduced above, in cascade (series) and jointly trained. The LaMOSNet architecture at the time of training is shown in Fig. 1.

B. Training of LaMOSNet

To use $\mathcal{D}_2$, we propose the following cost function, which consists of three parts:

$$L(\boldsymbol{\theta}, \boldsymbol{\phi}) = L_M + \lambda_S L_S + \lambda_C L_C. \tag{3}$$

Here L_M is the MOS-enforcing part, L_S is the score-enforcing part, and L_C is a consistency part; λ_S and λ_C are appropriate regularization parameters. Since judges' scores are noisy, and hence also the observed MOS, we propose to use the method Stochastic Gradient Noise (SGN), induced by stochastic label noise [9]. This means that instead of using $\tilde{y}_n$ and $s^{(j)}_{\mathbf{x}_n}$ as targets, we perturb these values with Gaussian noise. Let $z_1, z_2 \in \mathcal{N}(0, \sigma^2)$ be Gaussian noise drawn at each iteration (i.e., re-drawn each time the model sees the speech clips), and let

$$\begin{aligned} \tilde{y}^z_n &= \tilde{y}_n + z_1, \\ s^{(j),z}_{\mathbf{x}_n} &= s^{(j)}_{\mathbf{x}_n} + z_2. \end{aligned} \tag{4}$$

Then the parts L_M and L_S are defined as follows

$$\begin{aligned} L_M &= \sum_{n=1}^N \|\tilde{y}^z_n - \hat{y}_n\|^2 = \sum_{n=1}^N \|\tilde{y}^z_n - f_{\boldsymbol{\theta}}(\mathbf{x}_n)\|^2, \\ L_S &= \sum_{n=1}^N \sum_{(j) \in \mathcal{J}_n} \|s^{(j),z}_{\mathbf{x}_n} - \hat{s}^{(j)}_{\mathbf{x}_n}\|^2 \\ &= \sum_{n=1}^N \sum_{(j) \in \mathcal{J}_n} \|s^{(j),z}_{\mathbf{x}_n} - g_{\boldsymbol{\phi}}(\mathbf{x}_n, f_{\boldsymbol{\theta}}(\mathbf{x}_n), (j))\|^2. \end{aligned} \tag{5}$$

We use $\sigma^2 = 0.01$, obtained from a small hyperparameter experiment on the validation data.

The consistency loss part L_C is realized using a teacher-student learning approach as per [10]. The reason for using the approach is to provide robust learning in presence of noisy labels (noisy-data aspect). We use the approach the following way. Two LaMOSNet models are initialized, one is called the teacher model and the other one is called the student model. The teacher model inherits the student model's parameters at initialization. During training, the teacher model updates its parameters $\boldsymbol{\theta}'$, $\boldsymbol{\phi}'$ with respect to an exponential moving average of the student's parameters $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, i.e. $\boldsymbol{\theta}' = \alpha\boldsymbol{\theta}' + (1-\alpha)\boldsymbol{\theta}$, $\boldsymbol{\phi}' = \alpha\boldsymbol{\phi}' + (1-\alpha)\boldsymbol{\phi}$, where α is a hyperparameter [10]. In particular, the teacher model is not directly trained to minimize any loss function. The consistency loss $\mathcal{L}_C = \mathcal{L}_{C,M} + \mathcal{L}_{C,B}$, where the parts are given by

$$\begin{aligned} \mathcal{L}_{C,M} &= \sum_{n=1}^{N} \|f_{\boldsymbol{\theta}}(\mathbf{x}_n) - f_{\boldsymbol{\theta}'}(\mathbf{x}_n)\|^2, \\ \mathcal{L}_{C,B} &= \sum_{n=1}^{N} \sum_{(j)\in\mathcal{J}_n} \|g_{\boldsymbol{\phi}}(\mathbf{x}_n, f_{\boldsymbol{\theta}}(\mathbf{x}_n), (j)) \\ &\qquad - g_{\boldsymbol{\phi}'}(\mathbf{x}_n, f_{\boldsymbol{\theta}'}(\mathbf{x}_n), (j))\|^2. \end{aligned} \tag{6}$$

For clarity, the student model updates its parameters using a numerical algorithm minimizing the loss $L(\boldsymbol{\theta}, \boldsymbol{\phi})$ in (3). The teacher model updates its parameter according to an exponential moving average of the student's parameters.

C. Use of LaMOSNet in testing phase

After training, the MOS-providing regression function $f_{\boldsymbol{\theta}}(\mathbf{x})$ can be directly used to estimate the MOS $\hat{y}$, given $\mathbf{x}$. Thus, we discard the OS-providing regression function.

D. Some details of the architecture

The DNN of LaMOSNet architecture has similar architectural components as MBNet, with the difference of having predicted MOS as input to the OS-providing regression function. The MOS-providing regression function consists of 12 convolutional layers, followed by 1 BLSTM, and then 2 fully connected layers, which is the same configuration as in [7]. The reason for the CNN-BLSTM architecture is because it gave the best performance in an architectural study in [4].

For the OS-providing regression function, the speech clip is first processed by a convolutional layer with 16 kernels, resulting in a $heigth \times width \times 16$ feature map. The ID of the judge is embedded into $\mathbb{R}^{height \times width}$ according to a one-to-one mapping. The mapping is done by assigning a value in $\mathbb{R}^{height \times width}$ to each ID randomly from a normal distribution with zero mean and unit variance. The predicted MOS is also embedded into $\mathbb{R}^{height \times width}$ such that the MOS is repeated $height \times width$ times into a vector in $\mathbb{R}^{height \times width}$. After this has been done, both the embedded judge ID and embedded predicted MOS are concatenated to the feature map of the speech clip along the channel dimension. This gives a feature map of size $height \times weight \times 18$, which is the input to the OS-regression function.

The OS-providing regression function consists of 4 convolutional layers, followed by 1 BLSTM, and then 2 fully connected layers. For both the MOS-providing and OS-providing regression functions, batch normalization and dropout were used. See [7] for more details on model architecture.

IV. Experiments

In this section, we evaluate LaMOSNet and compare it vis-à-vis several existing methods using appropriate datasets and performance measures. The methods we compare are MOSNet [4], MOSNet+EL (here EL means the use of Encoding Layer for MOSNet) [14], MOSNet+GQT (here GQT means use of Global Quality Tokens for MOSNet) [14], MOSNet+MTL (here MTL means use of Multi Task Learning for MOSNet) [15], MOSNet+MTL+FL (here MTL means Multi-Task learning, and FL means the use of Focal Loss for MOSNet) [15], MBNet [7], and LDNet [8].

Among the above methods, we simulated MBNet and LDNet using their publicly available codebases. Other methods are quoted from the relevant literature.

A. Datasets for experiments

We use two datasets for experiments: Voice Conversion Challenge 2018 (VCC2018) [17], and Voice Conversion Challenge 2016 (VCC2016) [18]. VCC2018 is used for training, validation, and testing, while VCC2016 is only used for testing. Hence, we address an important issue of *statistical variability* by training on VCC2018 and testing on VCC2016.

VCC2018 Dataset: VCC2018 consists of 20 580 speech clips with various degrees of synthetic speech artifacts. The speech clips are obtained from 38 different audio-to-audio voice conversion systems, where each system has transformed the vocal identity of the speaker. Each speech clip was rated by 4 judges from a crowdsourcing program, where each judge assessed the naturality of the spoken words in each speech clip according to the MOS scale (i.e., a discrete value from 1 to 5) [19]. There are in total 270 judges, and each judge has rated on average 226 speech clips. Among the 20 580 speech clips, we use 13 580, 3 000, and 4 000 clips for training, validation, and testing, respectively.

For the VCC2018 dataset, we can evaluate performance in two ways – system-level and utterance-level. As there are 38 conversion systems in VCC2018, we can assess each system's performance as the mean of the speech quality of each speech clip. The system-level performance is the performance to predict the mean of the speech quality of a system's speech clips.

On the other hand, utterance-level performance is the performance of a model to predict the MOS of a given speech clip. Hence, the system which has operated on a speech clip is irrelevant in the utterance-level performance.

VCC2016 Dataset: To test generalizability with unseen judges, VCC2016 is used for testing while VCC2018 is used for training. VCC2016 dataset consists of 26028 speech clips from 20 different systems [18]. Only system-level performance is public, hence only system-level performance is evaluated in our experiments.

TABLE I
PERFORMANCE OF LAMOSNET AND COMPARISON WITH PRIOR METHODS. BOLDFACE NUMBERS HIGHLIGHT THE BEST VALUE IN EACH RESPECTIVE COLUMN.

		VCC2018						VCC2016		
	Training	Utterance-level			System-level			System-level		
Model	Data Size	MSE	LCC	SRCC	MSE	LCC	SRCC	MSE	LCC	SRCC
Not simulated. Results quoted from literature.										
MOSNet [14], [15]	15.6k	0.448	0.651	0.619	0.039	0.966	0.924	0.316	0.896	0.858
MOSNet+EL [14]	15.6k	0.444	0.656	0.617	0.031	0.974	0.938	0.242	0.908	0.855
MOSNet+GQT [14]	15.6k	0.447	0.654	0.621	0.041	0.968	0.931	0.242	0.921	0.853
MOSNet+MTL [15]	15.6k	0.435	0.664	0.618	0.019	0.983	0.944	0.227	0.925	0.883
MOSNet+MTL+FL [15]	15.6k	0.431	0.668	0.622	**0.016**	**0.985**	0.944	0.208	0.904	0.864
Simulated in our experiments.										
MBNet	13.6k	0.713	0.662	0.632	0.309	0.943	0.943	**0.162**	0.935	0.881
LDNet	13.6k	**0.428**	0.680	0.644	0.023	0.984	**0.963**	0.295	0.885	0.864
LaMOSNet	13.6k	0.432	**0.687**	**0.656**	0.034	0.982	0.960	0.325	**0.936**	**0.888**

B. Performance measures

Performance measures for evaluations are mean-square-error (MSE), linear-correlation-coefficient (LCC), and Spearman's-rank-correlation-coefficient (SRCC). In our experiments, while we are using the three stated performance measures, relatively higher importance could be given to LCC and SRCC measures as per [7].

For system-level evaluation, we use performance measures appropriately. For example, VCC2018 has 38 systems. There we compute MOS for each system across all speech clips. In this way, we get 38 MOS numbers and correspondingly 38 predicted MOS. Then we compute MSE, LCC, and SRCC.

C. Features and training

All speech clips were downsampled to $16\,kHz$. We use a spectrogram as the feature input to our system. For spectrogram computation, we used $32\,ms$ window length and $8\,ms$ window shift. We use repetitive padding of a speech clip instead of zero-padding, to stabilize mean and variance estimation in batch normalization [20].

LaMOSNet was implemented and trained on VCC2018. It was trained for 60 epochs with Adam optimizer using a learning rate of 10^{-4}, weight decay of 10^{-5}, and dropout of 30%. For the loss function in eq. (3), we selected the hyperparameters $\lambda_S = 4$ and $\lambda_C = 1$. 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 LCC on validation data, where testing on validation data was performed after each epoch.

For comparison, we simulated MBNet and LDNet. We used an unofficial, open-sourced implementation of MBNet[1], as per [8]. We used the official open-sourced implementation of LDNet[2], and used the MobileV3/RNN/- model architecture as this had the best LCC performance on the utterance-level [8]. Best models are chosen as per standard validation.

[1]https://github.com/sky1456723/Pytorch-MBNet/

[2]https://github.com/unilight/LDNet

D. Experimental results for comparison

In Table I we show both the utterance-level and system-level performances for VCC2018, and the system-level performance for VCC2016. LaMOSNet, MBNet, and LDNet are trained using the train-and-validation datasets of VCC2018. Each of the models LaMOSNet, MBNet, and LDNet was trained 10 independent times respectively, and the average test performance is reported in the table. The performances of other methods are quoted from the relevant literature.

From Table I, we note that LaMOSNet provides state-of-the-art performances for the utterance-level study on VCC2018 in the sense of LCC and SRCC. It also shows state-of-the-art performances in the generalizability test on VCC2016 for LCC and SRCC. The results indicate that LaMOSNet is a competitive performer.

E. Experimental results for ablation study

To investigate the effect of different components of LaMOSNet, we conducted an ablation study. First, LaMOSNet was trained without SGN. Second, LaMOSNet was trained without the student-teacher training methodology. Third, LaMOSNet was trained without consistency in the cost function, which means $\lambda_C = 0$ in (3). Finally, LaMOSNet was constructed without the OS-providing regression function $g_{\boldsymbol{\phi}}$. Removal of the OS-providing regression function makes the architecture being comprised of solely the MOS-providing regression function $f_{\boldsymbol{\theta}}$, which means it is close to the MOSNet architecture.

The ablation study results are reported in Table II where '-SGN' means LaMOSNet without SGN. In the case of the VCC2018 dataset, the removal of any component leads to a loss in all three performance measures. For system-level performances of both VCC2018 and VCC2016, the removal of components leads to a loss in the LCC performance measure. Therefore, we argue that all the components and training methodologies that we use for LaMOSNet are important.

V. CONCLUSION

In this work, we have proposed the model LaMOSNet that predicts both the MOS and individual judges' scores. It uses semi-supervised learning techniques from image classification,

TABLE II
ABLATION STUDY OF LAMOSNET. BOLDFACE NUMBERS HIGHLIGHT THE BEST VALUE IN EACH RESPECTIVE COLUMN.

	VCC2018						VCC2016		
	Utterance-level			System-level			System-level		
LaMOSNet	MSE	LCC	SRCC	MSE	LCC	SRCC	MSE	LCC	SRCC
Normal	**0.432**	**0.687**	**0.656**	**0.034**	**0.982**	0.960	0.325	**0.936**	0.888
- SGN	0.480	0.686	**0.656**	0.088	0.975	0.952	0.350	0.930	0.886
- student-teacher	0.477	0.676	0.647	0.075	0.975	0.949	0.537	0.930	0.898
- $\mathcal{L}_C$	0.455	0.685	0.655	0.055	0.978	0.955	**0.249**	0.935	0.881
- $g_{\boldsymbol{\phi}}$	0.435	0.683	0.654	0.036	0.978	**0.961**	0.264	0.934	**0.906**

namely stochastic gradient noise (SGN) and teacher-student consistency, as human judgment can be noisy and biased. By the correlation measures LCC and SRCC, LaMOSNet achieves state-of-the-art performance on the utterance-level on VCC2018 and on the system-level on VCC2016. The results illustrate that LaMOSNet is a competitive performer.

REFERENCES

[1] Brian Patton, Yannis Agiomyrgiannakis, Michael Terry, Kevin Wilson, Rif A. Saurous, and D. Sculley, "Automos: Learning a non-intrusive assessor of naturalness-of-speech," 2016.

[2] Szu-Wei Fu, Yu Tsao, Hsin-Te Hwang, and Hsin-Min Wang, "Quality-net: An end-to-end non-intrusive speech quality assessment model based on blstm," 2018.

[3] Pranay Manocha, Buye Xu, and Anurag Kumar, "Noresqa: A framework for speech quality assessment using non-matching references," 2021.

[4] Chen-Chou Lo, Szu-Wei Fu, Wen-Chin Huang, Xin Wang, Junichi Yamagishi, Yu Tsao, and Hsin-Min Wang, "Mosnet: Deep learning based objective assessment for voice conversion," *CoRR*, vol. abs/1904.08352, 2019.

[5] Chandan K. A. Reddy, Vishak Gopal, and Ross Cutler, "DNSMOS: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," *CoRR*, vol. abs/2010.15258, 2020.

[6] Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller, "NISQA: A deep CNN-self-attention model for multidimensional speech quality prediction with crowdsourced datasets," in *Interspeech 2021*. aug 2021, ISCA.

[7] Yichong Leng, Xu Tan, Sheng Zhao, Frank K. Soong, Xiangyang Li, and Tao Qin, "Mbnet: MOS prediction for synthesized speech with mean-bias network," *CoRR*, vol. abs/2103.00110, 2021.

[8] Wen-Chin Huang, Erica Cooper, Junichi Yamagishi, and Tomoki 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, pp. 896–900.

[9] Yixin Wu, Rui Luo, Chen Zhang, Jun Wang, and Yaodong Yang, "Revisiting the characteristics of stochastic gradient noise and dynamics," 2021.

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

[11] Pengfei Chen, Guangyong Chen, Junjie Ye, Jingwei Zhao, and Pheng-Ann Heng, "Noise against noise: stochastic label noise helps combat inherent label noise," in *International Conference on Learning Representations*, 2021.

[12] Jingfeng Wu, Wenqing Hu, Haoyi Xiong, Jun Huan, Vladimir Braverman, and Zhanxing Zhu, "On the noisy gradient descent that generalizes as sgd," 2019.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," 2017.

[14] Yeunju Choi, Youngmoon Jung, and Hoirin Kim, "Deep MOS predictor for synthetic speech using cluster-based modeling," in *Interspeech 2020*. oct 2020, ISCA.

[15] Yeunju Choi, Youngmoon Jung, and Hoirin Kim, "Neural mos prediction for synthesized speech using multi-task learning with spoofing detection and spoofing type classification," 2020.

[16] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif 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*, Jennifer Dy and Andreas Krause, Eds. 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 5180–5189, PMLR.

[17] Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, Tomi Kinnunen, and Zhenhua Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," 2018.

[18] Tomoki Toda, Ling-Hui Chen, Daisuke Saito, Fernando Villavicencio, Mirjam Wester, Zhizheng Wu, and Junichi Yamagishi, "The Voice Conversion Challenge 2016," in *Proc. Interspeech 2016*, 2016, pp. 1632–1636.

[19] Weisi Lin, Dacheng Tao, Janusz Kacprzyk, Zhu Li, Ebroul Izquierdo, and Haohong Wang, *Multimedia Analysis, Processing and Communications*, Springer Publishing, New York, 2011.

[20] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. 2015, ICML'15, p. 448–456, JMLR.org.