



# RAMP: Retrieval-Augmented MOS Prediction via Confidence-based Dynamic Weighting

Hui Wang<sup>1</sup>, Shiwan Zhao<sup>†</sup>, Xiguang Zheng<sup>2</sup>, Yong Qin<sup>1,\*</sup>

<sup>1</sup>Nankai University, China  
<sup>2</sup>Kuaishou Technology, China

wanghui\_hlt@mail.nankai.edu.cn, zhaosw@gmail.com, zhengxiguang@kuaishou.com, qinyong@nankai.edu.cn

## Abstract

Automatic Mean Opinion Score (MOS) prediction is crucial to evaluate the perceptual quality of the synthetic speech. While recent approaches using pre-trained self-supervised learning (SSL) models have shown promising results, they only partly address the data scarcity issue for the feature extractor. This leaves the data scarcity issue for the decoder unresolved and leading to suboptimal performance. To address this challenge, we propose a retrieval-augmented MOS prediction method, dubbed **RAMP**, to enhance the decoder’s ability against the data scarcity issue. A fusing network is also proposed to dynamically adjust the retrieval scope for each instance and the fusion weights based on the predictive confidence. Experimental results show that our proposed method outperforms the existing methods in multiple scenarios.

**Index Terms:** MOS prediction, speech assessment, retrieval-augmented method, confidence-based dynamic weighting

## 1. Introduction

Evaluation of synthetic speech typically involves the use of objective and subjective methods. Objective methods, such as Mel Cepstral Distortion (MCD) and F0 Frame Error (FFE), require reference audio, making them impractical or even impossible to use in scenarios where reference audio is unavailable [1]. On the other hand, subjective methods, such as Mean Opinion Score (MOS), rely on listening tests conducted by crowd-sourced listeners, which can be time and resource consuming.

Automatic MOS prediction using machine learning [2, 3, 4] has recently gained popularity as it allows for quality evaluation that matches human perception with low time and resource cost. However, training of such methods still relies on manually scored data and often suffers from data scarcity due to limited budgets. To alleviate this problem, self-supervised learning (SSL) models [5, 6] trained on large-scale unlabeled data are employed as the feature extractor, followed by a downstream MOS prediction decoder trained on small-scale data with labeled MOS scores [7, 8]. Recent works based on this SSL-based structure [9, 10, 11] have outperformed the earlier works trained from scratch [2, 12, 13] in the MOS prediction tasks.

Existing SSL-based frameworks, as depicted in Figure 1, mainly focus on improving the feature extractor based on the help of large-scale pre-training corpora to obtain better representations. To name a few, Yang et al. [11] fuse seven pre-trained SSL models, as both the data and model configurations used for pre-training can impact SSL model performance. Tseng et al. [9] demonstrate that domain adaptive pre-training (DAPT) can reduce domain mismatch between speech

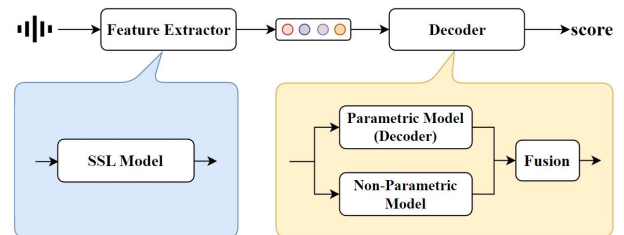


Figure 1: The SSL-based framework (bottom left) improves the feature extractor with an SSL model and the proposed framework (bottom right) augments the decoder by introducing a non-parametric model and a fusing network.

in the pre-training corpus and fine-tuning MOS corpus. Saeki et al. [10] concatenate as much information as possible from phonemes, raters, and other sources with the SSL representation vector. Vioni et al. [14] include prosodic and linguistic features as inputs to improve system performance.

However, the decoder, which decodes the features into scores, is only trained with the MOS dataset. The data scarcity issue for the decoder remains unsolved, leading to suboptimal performance. A few works attempt to improve the neural decoder. Tseng et al. [9] replace the linear layer of the decoder with a DNN, but their results indicate that increasing the decoder’s parameters does not necessarily improve performance. Chen et al. [15] design multi-task heads to predict both quality and intelligibility scores simultaneously. In summary, these neural decoders can be classified as parametric methods by learning the mapping from representations to scores, which generally require large-scale labeled data to achieve good generalization ability and tend to suffer difficulties in adapting to new domains with distribution shifts.

Given the powerful SSL-based feature extractor and the weak decoder, we share a similar hypothesis with  $k$ NN-LM [16] that *the representation learning is easier than the prediction*. Thus, we propose a retrieval-augmented MOS prediction method, dubbed **RAMP**, to enhance the decoder for addressing the data scarcity issue. Specifically, as shown in Figure 1, we augment the decoder of the SSL-based framework by linearly interpolating its prediction score with the  $k$ -nearest neighbors ( $k$ NN) model. The  $k$ NN model, which is a non-parametric method, not only is good at memorizing rare patterns [17] but also handles cross-domain issues in a flexible way [16].

The fusion of the parametric and non-parametric methods has been successfully utilized in tasks such as language modeling [16, 18] and machine translation [19]. However, they combine the two methods by static weights. We argue that the predictive ability of parametric methods varies for different in-

<sup>†</sup>Independent researcher.

\* Corresponding author.

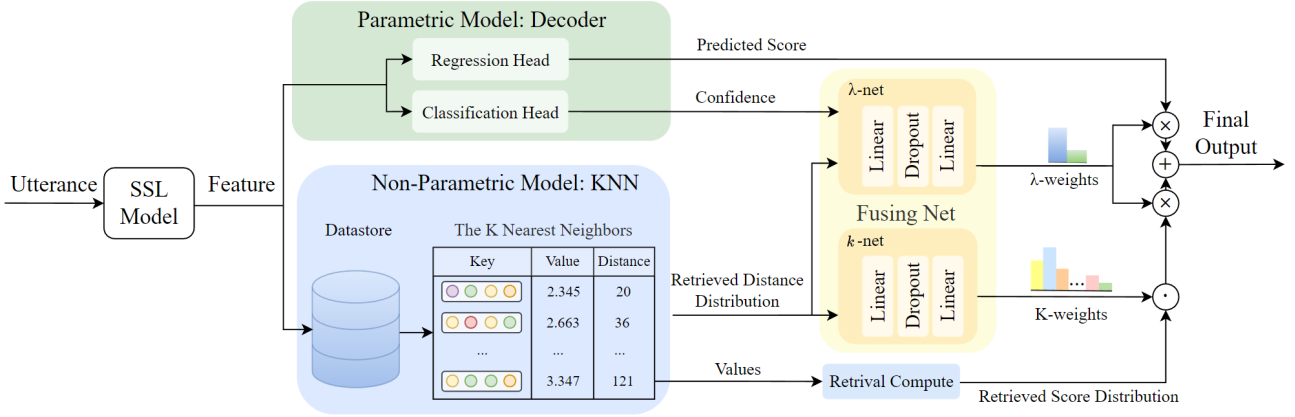


Figure 2: The illustration of the system flow at the inference time.

stances due to the uneven distribution of data in the dataset. For instance, the model may be more confident in predicting scores for some instances than for others. In such cases, less help from non-parametric methods may be needed. To this end, we propose a *confidence-based dynamic weighting* scheme to balance the outcomes of the two methods. Furthermore, the number of retrievals of the  $k$ NN model is preset and sensitive to noise [20, 21]. To improve the robustness of the model, we automatically predict the importance of the number of retrievals and average the scores for different numbers.

The main contributions of this work are threefold:

- We propose RAMP, a novel retrieval-augmented MOS prediction method, to enhance the neural decoder in SSL-based frameworks.
- We design a fusing network to dynamically adjust the retrieval scope for each instance and the fusion weights based on the predictive confidence.
- We have demonstrated the effectiveness of the method through extensive comparative and ablation experiments.

## 2. The System Overview

The training process of our system consists of two stages. In the first training stage, we fine-tune the SSL-based model (i.e., the feature extractor) with multi-task heads (i.e., the decoder). In the second stage, we freeze the feature extractor and decoder to train the fusing network to combine results from the decoder and the  $k$ NN model. Note that we use the frozen feature extractor on training data to construct the datastore.

The overall flow of our system at the inference time is presented in Figure 2. During inference, the SSL model extracts the feature representation for the input utterance. The features are fed into the parametric and non-parametric paths separately. The corresponding outputs are then fused to get the final output. In the case of evaluating cross-domain audios, the steps are identical, except replacing the datastore with the target domain data without an additional fine-tuning stage.

## 3. The Proposed Method

The proposed method consists of the parametric path, the non-parametric path, and the fusing network.

### 3.1. Parametric path

The *parametric path* refers to the use of a neural network-based decoder to handle representations. The decoder is designed as

a multi-task architecture consisting of a regression head and a classification head, which are implemented using several linear layers following SSL-MOS [7]. The purpose of introducing a classification head is to guide the model during fine-tuning, as well as to output the confidence of each score bin. Obtaining the confidence of the parametric model is crucial for subsequent result fusion. For the  $i$ -th instance  $(u_i, s_i)$ , we first map the score  $s_i$  to the bin id  $b_i$ . Then the model outputs a prediction score  $S_p$  along with a confidence probability distribution  $[c_1, c_2, \dots, c_n]$  on  $n$  bins. The loss function is defined by:

$$\mathcal{L} = \mathcal{L}_{reg}(u_i, s_i) + \alpha \mathcal{L}_{cls}(u_i, b_i), \quad (1)$$

where  $\mathcal{L}_{reg}$  and  $\mathcal{L}_{cls}$  are the MSE and cross entropy losses, for the regression and classification heads, respectively. The  $\alpha$  is a hyper-parameter that balances the two types of losses.

### 3.2. Non-parametric path

Compared to the parametric path, the *non-parametric path* directly exploits the representations with a  $k$ NN model. The representation of the utterance being evaluated is used as the query to retrieve the most similar data instances from the datastore, along with their corresponding distances and labels. This information is then used as part of the input to the fusing network.

**Datastore:** Let  $f(\cdot)$  be the SSL model that maps an utterance  $u$  to its representation. For a training sample  $(u_i, s_i) \in D$ , we create a key-value pair  $(k_i, v_i)$ , where the key  $k_i = f(u_i)$  is the representation vector of the utterance  $u_i$  and the value  $v_i = s_i$  is its target score. The set of all key-value pairs constructed from all training examples in  $D$  forms the datastore  $(\mathcal{K}, \mathcal{V})$ :

$$(\mathcal{K}, \mathcal{V}) = \{(f(u_i), s_i) | (u_i, s_i) \in D\}. \quad (2)$$

**Inference:** During testing, the SSL model generates a representation  $q$  for the input utterance, which is then utilized by the  $k$ NN model to retrieve the  $k$  nearest neighbors, denoted by  $\mathcal{N}_k$ , from the datastore based on a given distance function  $d(q, \cdot)$ . The retrieved score  $S_{r,k}$  is then computed as follows:

$$S_{r,k} = \sum_{(k_i, v_i) \in \mathcal{N}_k} w_{ik} v_i, \quad (3)$$

where the weight  $w_{ik}$  is the inverse of its distance. Thus, neighbors that are closer will exert more influence than those that are farther away.

Table 1: The performances of our framework along with two systems on BVCC and BC2019 test set.

(a) The results of the in-domain experiments.

dataset	model	U_MSE↓	U_LCC↑	U_SRCC↑	U_KTAU↑	S_MSE↓	S_LCC↑	S_SRCC↑	S_KTAU↑
BVCC	SSL-MOS	0.246	0.875	0.872	0.697	0.113	0.928	0.923	0.770
	DDOS	0.212	0.880	0.880	0.707	0.110	<b>0.933</b>	<b>0.932</b>	0.782
	RAMP	<b>0.195</b>	<b>0.881</b>	<b>0.881</b>	<b>0.708</b>	<b>0.097</b>	0.931	<b>0.932</b>	<b>0.784</b>
BC2019	SSL-MOS	0.253	0.901	0.871	0.690	0.098	0.980	0.970	0.871
	DDOS	<b>0.169</b>	0.914	0.887	0.710	<b>0.052</b>	0.976	0.955	0.848
	RAMP	0.188	<b>0.916</b>	<b>0.891</b>	<b>0.717</b>	0.053	<b>0.987</b>	<b>0.987</b>	<b>0.926</b>

(b) The results of the cross-domain experiments.

dataset	model	U_MSE↓	U_LCC↑	U_SRCC↑	U_KTAU↑	S_MSE↓	S_LCC↑	S_SRCC↑	S_KTAU↑
BC2019	SSL-MOS	3.187	0.527	0.549	0.403	2.976	0.590	0.655	0.569
	DDOS	1.331	0.678	0.694	0.502	1.119	0.766	0.797	0.637
	RAMP	0.658	0.826	0.780	0.587	0.493	0.929	0.907	0.772
	RAMP(np)	<b>0.294</b>	<b>0.842</b>	<b>0.789</b>	<b>0.596</b>	<b>0.093</b>	<b>0.955</b>	<b>0.926</b>	<b>0.797</b>

Since  $k$ NN is sensitive to  $k$ , we improve its robustness by using various values of  $k$  in the range  $[1, 2, \dots, K]$ , where  $K$  is the hyperparameter. As a result, we obtain the retrieved score distribution  $[S_{r,1}, S_{r,2}, \dots, S_{r,K}]$ , as well as the retrieved distance distribution  $[d_1, d_2, \dots, d_K]$ , where  $d_k$  is the distance between  $q$  and the  $k$ -th nearest neighbor.

### 3.3. Fusing network

We introduce two lightweight networks,  $k$ -net and  $\lambda$ -net, to dynamically predict the probability of the retrieval scope  $k$  and the fusion weight distribution  $\lambda$  based on the predictive confidence for each instance.

**$k$ -net:** The  $k$ -net is a lightweight net consisting of only two linear layers which dynamically predicts the probability of each  $k$  for each instance. It takes as input the distance distribution  $d = [d_1, d_2, \dots, d_K]$  of the retrieved neighbors and outputs the probability distribution  $p_{knet}$  over different values of  $k$ :

$$p_{knet} = \text{softmax}(knet(d)). \quad (4)$$

Therefore, the final retrieved score can be computed as a weighted average of different  $S_{r,k}$ :

$$S_r = \sum_{k=1}^K p_{knet}(k) S_{r,k}. \quad (5)$$

**$\lambda$ -net:** The  $\lambda$ -net has the same structure as  $k$ -net. To balance the outcomes of two paths, it takes the confidences as the input, in addition to the distance distribution  $d$ . The confidences include the top-8 values  $c^{top}$  from the confidence probability distribution  $[c_1, c_2, \dots, c_n]$ , and two confidences  $c^{S_r}$  and  $c^{S_p}$  of the bins to which  $S_r$  and  $S_p$  belong. Thus, the weight and then the final score can be computed as:

$$w = (w_p, w_r) = \text{softmax}(\lambda net([d; c^{top}; c^{S_r}; c^{S_p}])), \quad (6)$$

$$S = w_p S_p + w_r S_r, \quad (7)$$

where,  $S_p$  and  $S_r$  are the scores of the parametric and non-parametric paths, respectively.

## 4. Experiments

### 4.1. Datasets and Metrics

The experiments in this paper use three corpora: BVCC [22], BC2019 [23], and SOMOS [24]. BVCC contains 7,106 English

samples from the Blizzard Challenges, the Voice Conversion Challenges, and published samples of ESPNet [25]. The ratio of training/development/test is 70%/15%/15%, respectively. BC2019 contains Mandarin TTS samples submitted to the 2019 Blizzard Challenge [23]. There are 136 samples for training, 136 samples for validation, and 540 samples for testing. We also use SOMOS in the ablation study. It consists of 20K TTS audio files generated from several Tacotron-like acoustic models [26] and an LPCNet vocoder [27].

Model evaluation is performed at the utterance level and system level, denoted as ‘U\_’ and ‘S\_’, respectively. Mean square error (MSE) and various correlation coefficient metrics are used. In particular, the Linear Correlation Coefficient (LCC), the Spearman Rank Correlation Coefficient (SRCC) and the Kendall Tau Rank Correlation (KTAU) scores are evaluated. In general, smaller errors and higher correlations indicate better model performance.

### 4.2. Implementation Details

We use the published wav2vec2.0 [5] base model pre-trained on Librispeech [28] as the feature extractor. In both training stages, the models are trained for 1,000 epochs with a batch size of 4 and a learning rate of 0.0001. Training will be stopped early when the loss does not decrease for 20 epochs. Gradient accumulation is used to simulate large batch. In the experiments, the length of the score bin is 0.25. The  $\alpha$  is set to 1 in equation 1. We set  $K = 60$  for BVCC,  $K = 8$  for BC2019 in Table 1. The L2 distance is used in Section 3.2. We crop the long audio in the SOMOS dataset due to memory limitations for all compared methods.

### 4.3. In-domain Analysis

We first compare the performance of our method RAMP with SSL-MOS [7] and DDOS [9]. SSL-MOS is one of the first systems to employ the SSL model, which delivers high prediction performance with an easy-to-use framework. DDOS is one of the winning solutions of VoiceMOS Challenge 2022 [29], providing competitive outcomes across many metrics.

Table 1a shows the performance of three systems in BVCC and BC2019. On the BVCC test set, our system RAMP performs better than SSL-MOS in all metrics. Compared to DDOS, RAMP performs better in error metrics and similarly in correlation metrics. To evaluate the performance in BC2019, all three

Table 2: The performance of two paths in three datasets: SOMOS (*big*), BVCC (*medium*), and BC2019 (*small*).

	U_MSE↓		U_LCC↑		U_SRCC↑		U_KTAU↑		S_MSE↓		S_LCC↑		S_SRCC↑		S_KTAU↑	
	P	NP	P	NP	P	NP	P	NP	P	NP	P	NP	P	NP	P	NP
b	0.185	<b>0.179</b>	0.658	<b>0.668</b>	0.653	<b>0.658</b>	0.468	<b>0.473</b>	0.182	<b>0.174</b>	0.664	<b>0.676</b>	0.659	<b>0.665</b>	0.474	<b>0.480</b>
m	<b>0.200</b>	0.201	0.874	<b>0.880</b>	0.876	<b>0.880</b>	0.702	<b>0.708</b>	0.106	<b>0.101</b>	0.917	<b>0.934</b>	0.915	<b>0.934</b>	0.756	<b>0.787</b>
s	0.226	<b>0.196</b>	0.880	<b>0.904</b>	0.831	<b>0.868</b>	0.634	<b>0.682</b>	0.064	<b>0.045</b>	0.968	<b>0.984</b>	0.926	<b>0.972</b>	0.797	<b>0.889</b>

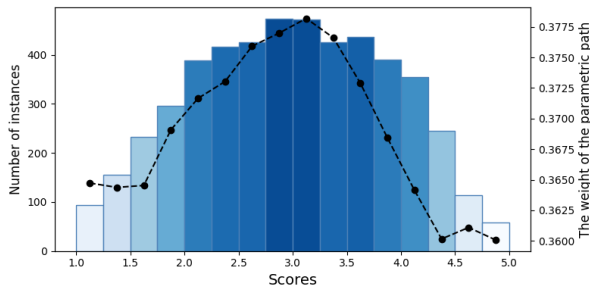


Figure 3: The bars are the score distribution of the BVCC dataset, and the curve represents the weights of the parametric path learned by the fusion network for each score bin. This shows dynamic weights are assigned aligning with the data distribution, indicating the effectiveness of the fusing net.

models are first trained in BVCC and fine-tuned in BC2019 following the same procedure of the VoiceMOS Challenge 2022 to make a fair comparison. On the BC2019 test set, RAMP also outperforms SSL-MOS in all metrics. While DDOS outperforms our system slightly in terms of error metrics, our system consistently performs well across all six correlation metrics.

Moreover, in order to investigate how the confidence-based dynamic weighting scheme combines the two paths for efficient prediction, we visualize the fusion weights for various instances. Figure 3 displays the score distribution of the BVCC data and the weights of the parametric path predicted by the fusing network. The data exhibit a long-tailed distribution with relatively small numbers of scores that are very low or very high. As mentioned in Section 1, this unbalanced distribution of data can cause the neural network to have different prediction capabilities for data located in different score intervals. For head data, the neural network-based decoder (i.e., the parametric path) can predict their scores with high confidence, requiring less help from the non-parametric path. Thus, smaller weights are assigned to the non-parametric path and higher weights to the parametric path. Conversely, for data at the tail, higher weights are assigned to the non-parametric path and smaller weights to the parametric path.

#### 4.4. Cross-domain Analysis

The results of cross-domain experiments are presented in Table 1b. All systems are trained in BVCC and tested in the BC2019 test set without additional fine-tuning. As the results show, both SSL-MOS and DDOS exhibit a sharp drop in performance across domains. While our system can solve the cross-domain adaptation problem flexibly and excel in performance by updating the datastore without any additional training.

To further investigate the help from the non-parametric path for cross-domain settings. We create a new variant called RAMP(np), which only includes the non-parametric path. This results in further improvements in performance, demonstrating that the distribution shift undermines the mapping from repre-

sentations to scores, while the  $k$ NN model still performs well by directly leveraging the representations.

#### 4.5. Ablation study

We first compare the performance of the parametric and non-parametric paths across different data scales. In Table 2, ‘P’ and ‘NP’ denote the parametric and non-parametric paths, respectively. We conduct the experiments on three datasets: SOMOS, BVCC, and BC2019, which correspond to tens of thousands, thousands, and hundreds of scales, respectively. We can observe that as the size of the dataset decreases, the non-parametric path boosts the performance more significantly. This demonstrates that our proposed method is more applicable to low-resource tasks like most MOS prediction tasks.

We then conduct an ablation study on the fusing net. Vanilla  $k$ NN uses a fixed  $k$  for all instances and its performance is sensitive to the value of  $k$ . While our proposed fusing net specifies a predefined upper bound  $K$  of the retrieval range. Then for each instance, it dynamically computes the weighted average score from the  $K$   $k$ NN models, each trained with different numbers of nearest neighbors from 1 to  $K$ . The weight of each  $k$ NN is obtained through the network. We only show MSE and KTAU performance on the utterance level since the trends of system-level are similar. The results show that the fusing net performs better on average and performs more consistently when changing the hyper-parameter  $K$ , demonstrating that the fusing net can improve the accuracy and robustness of the model.

Table 3: The performance of the vanilla  $k$ NN and the fusing net.

$k/K$ -size	U_MSE↓		U_KTAU↑	
	Vanilla	Fusing	Vanilla	Fusing
5	0.216	0.197	0.695	0.704
10	0.203	0.199	0.703	0.704
15	0.198	0.197	0.707	0.705
30	0.197	0.194	0.708	0.708
60	0.197	0.195	0.708	0.708
mean	0.202	<b>0.196</b>	0.704	<b>0.706</b>
var	6.57E-05	<b>3.80E-06</b>	3.07E-05	<b>4.20E-06</b>

## 5. Conclusions

In this paper, we propose RAMP, a novel retrieval-augmented MOS prediction method, to enhance the neural decoder for alleviating the data scarcity issue in the SSL-based frameworks. We also design a fusing network to dynamically adjust the retrieval scope and the fusion weights based on the predictive confidence. The experimental results show that the proposed models perform well in both in-domain and cross-domain settings.

## 6. Acknowledgements

This work has been supported by the National Key R&D Program of China through grant 2022ZD0116307 and NSF China (Grant No.62271270).

## 7. References

- [1] M. Chinen, J. Skoglund, C. K. A. Reddy, A. Ragano, and A. Hines, "Using Rater and System Metadata to Explain Variance in the VoiceMOS Challenge 2022 Dataset," in *Proc. Interspeech 2022*, 2022, pp. 4531–4535.
- [2] C. Lo, S. Fu, W. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H. Wang, "Mosnet: Deep learning-based objective assessment for voice conversion," *Proc. Interspeech 2019*, pp. 1541–1545, 2019.
- [3] B. Patton, Y. Agiomyriannakis, M. Terry, K. W. Wilson, R. A. Saurous, and D. Sculley, "Automos: Learning a non-intrusive assessor of naturalness-of-speech," *ArXiv*, vol. abs/1611.09207, 2016.
- [4] S. wei Fu, Y. Tsao, H.-T. Hwang, and H.-M. Wang, "Quality-Net: An End-to-End Non-intrusive Speech Quality Assessment Model Based on BLSTM," in *Proc. Interspeech 2018*, 2018, pp. 1873–1877.
- [5] 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.
- [6] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhota, 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] E. Cooper, W.-C. Huang, T. Toda, and J. Yamagishi, "Generalization ability of mos prediction networks," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8442–8446, 2021.
- [8] A. Ragano, E. Benetos, M. Chinen, H. B. Martinez, C. K. Reddy, J. Skoglund, and A. Hines, "A comparison of deep learning mos predictors for speech synthesis quality," *arXiv preprint arXiv:2204.02249*, 2022.
- [9] W.-C. Tseng, W.-T. Kao, and H. yi Lee, "DDOS: A MOS Prediction Framework utilizing Domain Adaptive Pre-training and Distribution of Opinion Scores," in *Proc. Interspeech 2022*, 2022, pp. 4541–4545.
- [10] 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*, 2022, pp. 4521–4525.
- [11] Z. Yang, W. Zhou, C. Chu, S. Li, R. Dabre, R. Rubino, and Y. Zhao, "Fusion of Self-supervised Learned Models for MOS Prediction," in *Proc. Interspeech 2022*, 2022, pp. 5443–5447.
- [12] Y. Leng, X. Tan, S. Zhao, F. Soong, X.-Y. Li, and T. Qin, "Mbnnet: Mos prediction for synthesized speech with mean-bias network," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 391–395.
- [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)*. IEEE, 2022, pp. 896–900.
- [14] A. Vioni, G. Maniati, N. Ellinas, J. S. Sung, I. Hwang, A. Chalamandaris, and P. Tsiakoulis, "Investigating content-aware neural text-to-speech mos prediction using prosodic and linguistic features," *ArXiv*, vol. abs/2211.00342, 2022.
- [15] Y.-W. Chen and Y. Tsao, "InQSS: a speech intelligibility and quality assessment model using a multi-task learning network," in *Proc. Interspeech 2022*, 2022, pp. 3088–3092.
- [16] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Generalization through memorization: Nearest neighbor language models," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.
- [17] Z. Wan, Q. Liu, Z. Mao, F. Cheng, S. Kurohashi, and J. Li, "Rescue implicit and long-tail cases: Nearest neighbor relation extraction," in *Conference on Empirical Methods in Natural Language Processing*, 2022.
- [18] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *International conference on machine learning*, 2020, pp. 3929–3938.
- [19] U. Khandelwal, A. Fan, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Nearest neighbor machine translation," in *International Conference on Learning Representations*, 2021.
- [20] X. Zheng, Z. Zhang, J. Guo, S. Huang, B. Chen, W. Luo, and J. Chen, "Adaptive nearest neighbor machine translation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Aug. 2021, pp. 368–374.
- [21] H. Jiang, Z. Lu, F. Meng, C. Zhou, J. Zhou, D. Huang, and J. Su, "Towards robust k-nearest-neighbor machine translation," in *Conference on Empirical Methods in Natural Language Processing*, 2022.
- [22] 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.
- [23] Z. Wu, Z. Xie, and S. King, "The blizzard challenge 2019," in *Proc. Blizzard Challenge Workshop*, 2019.
- [24] G. Maniati, A. Vioni, N. Ellinas, K. Nikitaras, K. Klapsas, J. S. Sung, G. Jho, A. Chalamandaris, and P. Tsiakoulis, "SOMOS: The Samsung Open MOS Dataset for the Evaluation of Neural Text-to-Speech Synthesis," in *Proc. Interspeech 2022*, 2022, pp. 2388–2392.
- [25] 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," *2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7654–7658, 2020.
- [26] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyriannakis, R. A. J. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Interspeech*, 2017.
- [27] J.-M. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5891–5895, 2018.
- [28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015.
- [29] W. C. Huang, E. Cooper, Y. Tsao, H.-M. Wang, T. Toda, and J. Yamagishi, "The VoiceMOS Challenge 2022," in *Proc. Interspeech 2022*, 2022, pp. 4536–4540.