Interspeech 2024
1-5 September 2024, Kos, Greece

SVSNet+: Enhancing Speaker Voice Similarity Assessment Models with Representations from Speech Foundation Models

Chun Yin[1,2]*, Tai-Shih Chi*[1]*, Yu Tsao*[2]*, and Hsin-Min Wang*[2]

[1]National Yang Ming Chiao Tung University, Taiwan
[2]Academia Sinica, Taiwan

nash814@gmail.com, tschi@nycu.edu.tw, yu.tsao@citi.sinica.edu.tw, whm@iis.sinica.edu.tw

Abstract

Representations from pre-trained speech foundation models (SFMs) have shown impressive performance in many downstream tasks. However, the potential benefits of incorporating pre-trained SFM representations into speaker voice similarity assessment have not been thoroughly investigated. In this paper, we propose SVSNet+, a model that integrates pre-trained SFM representations to improve performance in assessing speaker voice similarity. Experimental results on the Voice Conversion Challenge 2018 and 2020 datasets show that SVSNet+ incorporating WavLM representations shows significant improvements compared to baseline models. In addition, while fine-tuning WavLM with a small dataset of the downstream task does not improve performance, using the same dataset to learn a weighted-sum representation of WavLM can substantially improve performance. Furthermore, when WavLM is replaced by other SFMs, SVSNet+ still outperforms the baseline models and exhibits strong generalization ability.

Index Terms: pre-trained speech foundation model, speaker voice similarity assessment

1. Introduction

A useful voice conversion (VC) system can be used in many areas, such as dubbing, personalized virtual assistants, preserving voices, and aiding in voice recovery after surgery. However, effectively evaluating the performance of such systems remains challenging. As a common evaluation for VC systems, the speaker voice similarity assessment aims to evaluate the resemblance between generated speech and natural speech. The evaluation can be objective [1] or subjective. While the results of subjective evaluation are closer to human perception, the time and cost required to conduct assessments such as listening tests are considerable. Therefore, automated and efficient methods for assessing speaker voice similarity are valuable.

In [2], Hu et al. proposed SVSNet, an end-to-end neural model for speaker voice similarity assessment, and showed satisfactory performance at both the utterance and system levels. SVSNet takes the raw speech waveforms as inputs and processes them with an encoder consisting of a SincNet module, four stacked residual-skipped-WaveNet convolution (rSWC) layers, and a BLSTM layer. However, given the constraint of limited training data, the encoder may not be sufficiently adept at extracting meaningful representations from the waveforms. Therefore, to further reinforce the speaker voice similarity assessment model, we propose SVSNet+, which integrates a large pre-trained speech foundation model (SFM) to extract speech representations. By leveraging a pre-trained SFM that learns effective speech representations from large-scale training data, SVSNet+ can acquire valuable information for the similarity prediction task. When evaluated on the Voice Conversion Challenge 2018 [3] and 2020 [4] (VCC2018 and VCC2020) datasets, SVSNet+ significantly outperforms previous work at the system level and demonstrates strong generalization ability. This study contributes to further future exploration of applying pre-trained SFM representations to speaker voice similarity assessment.

2. Related Work

2.1. Pre-trained speech foundation models

Pre-trained SFMs can undergo training in a supervised or unsupervised learning manner. Although supervised learning typically excels over unsupervised learning in various tasks, the process of collecting large-scale labeled data can be time-consuming and sometimes impractical. Therefore, many state-of-the-art pre-trained models are based on self-supervised learning (SSL), allowing them to acquire meaningful representations from large amounts of unlabeled data [5]. As one of the most commonly used models, wav2vec 2.0 [6] acquires contextual information by discerning representations that correspond to true quantized latent speech representations. On the other hand, HuBERT [7] utilizes clustering to generate labels and predicts hidden cluster assignments for masked speech representations. Modified from HuBERT, WavLM [8] uses a larger dataset during pre-training for joint learning of masked speech prediction and denoising. Moreover, the Massively Multilingual Speech (MMS) model [9] expands the number of languages in the training dataset and builds pre-trained models covering 1,406 languages based on wav2vec 2.0. Unlike the aforementioned SSL pre-trained models, Whisper [10] is an SFM trained in a weakly supervised manner, using a multitask training format, showcasing not only high robustness but also strong generalization ability. All of these SFMs have been employed to enhance performance on diverse speech processing tasks, demonstrating remarkable effectiveness. In this study, we employ all these SFMs to extract speech representations and evaluate their suitability for the speaker voice similarity assessment task.

2.2. Speech assessment tasks using SFM-extracted representations

In recent years, SFM representations have been applied in various tasks, such as speech enhancement (SE) [11], automatic speech recognition (ASR) [12], automatic speaker verification (ASV) [13], and voice conversion (VC) [14]. In speech assessment tasks, it is a prevalent practice to leverage pre-trained SSL representations for mean opinion score (MOS) prediction [15–18]. The latent representations extracted by wav2vec 2.0, HuBERT, and WavLM have been proven beneficial for these tasks. Furthermore, recent work by Zezario et al. [19] also

10.21437/Interspeech.2024-1540

showed that Whisper and MMS representations help predict human-perceived speech quality and intelligibility.

3. Proposed Method

The architecture of SVSNet+ is shown in Fig. 1. The waveforms of the test and reference utterances X_T and X_R are fed to a pre-trained SFM, which encodes them into layer-wise representations. Next, the corresponding weighted-sum representations are derived from layer-wise representations, and a linear layer is used to adjust the representation dimension. Then, the representations R_T and R_R are aligned by the co-attention module in both directions for maintaining symmetry. Afterwards, the distance module calculates the distance between R_T and $\hat{R}_R$ and that between R_R and $\hat{R}_T$. Finally, the prediction module uses these two distances to calculate a similarity score.

3.1. Pre-trained model and weighted sum

In this study, we study several SFMs, each containing a feature extractor and a transformer encoder, as shown in Fig. 1(b). For SSL-based SFMs, the feature extractors are CNN-based encoders that generate feature sequences at a frame rate of 20ms for audio sampled at 16kHz. For Whisper, the feature extractor first preprocesses the 16kHz audio input into 30-second chunks by zero-padding or trimming. Each chunk is then transformed into an 80-channel log-magnitude Mel spectrogram at a frame rate of 10ms and further processed by an encoder consisting of two convolutional layers and a GELU activation function. The stride of the second convolutional layer is 2 [10]. Therefore, the down-sampling factor of the feature extractor in each pre-trained SFM is 320x. For all SFMs, the extracted features are fed to the transformer encoder and processed by L hidden layers. Finally, to exploit the information from each hidden layer, the representations generated from all hidden layers are combined using the weighted sum module:

$$R_{WS} := \sum_{\ell=0}^{L-1} w^{\ell} R^{\ell}, \tag{1}$$

where $w^{\ell} \geq 0$ is the learnable weight for layer ℓ and $\sum_{\ell} w^{\ell} = 1$, and R^{ℓ} is the representation of layer ℓ. The weighted-sum representation is then passed through an additional linear layer for dimension adjustment.

3.2. Co-attention module

Following [2], the co-attention module is used to align the representation of one input with that of the other input by

$$\begin{aligned} \hat{R}_R &= Attention(R_T, R_R, R_R), \\ \hat{R}_T &= Attention(R_R, R_T, R_T), \end{aligned} \tag{2}$$

and output two aligned pairs $(R_T, \hat{R}_R)$ and $(R_R, \hat{R}_T)$, which will be input to the distance module. In this study, the scaled dot-product attention mechanism [20] is implemented.

3.3. Distance module and prediction module

Following [2], the utterance embedding is obtained by averaging its representations over time, and the 1-norm distance of each dimension of two embeddings is calculated:

$$\begin{aligned} D_{T,R} &= \| \, Mean(R_T) - Mean(\hat{R}_R) \, \|_1, \\ D_{R,T} &= \| \, Mean(R_R) - Mean(\hat{R}_T) \, \|_1 \, . \end{aligned} \tag{3}$$

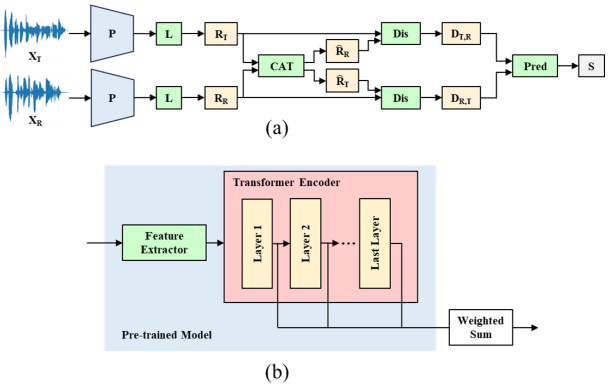

Figure 1: *(a) The architecture of SVSNet+. P, L, CAT, Dis, and Pred respectively denote the pre-trained model, linear layer, co-attention module, distance module, and prediction module. (b) The pre-trained model followed by the weighted sum module.*

Then, the prediction module takes in the two distances to derive the similarity scores:

$$\begin{aligned} \hat{S}_T &= \sigma(f_{lin2}(\rho_{ReLU}(f_{lin1}(D_{T,R})))), \\ \hat{S}_R &= \sigma(f_{lin2}(\rho_{ReLU}(f_{lin1}(D_{R,T})))), \end{aligned} \tag{4}$$

where $\sigma(.)$ is an activation function, $f_{lin1}(.)$ and $f_{lin2}(.)$ are linear layers, and $\rho_{ReLU}(.)$ is the rectified linear unit (ReLU) activation function. There are two types of prediction modules: regression and classification. For regression tasks, the identity function is used as the activation function, while for classification tasks, the softmax function is used. The output size of the second linear layer is 1 (for regression) and 4 (for classification). The final score is the average of the two predicted scores:

$$\hat{S} = (\hat{S}_T + \hat{S}_R)/2. \tag{5}$$

The training objective of the proposed model is to match these scores with the corresponding human-labeled similarity scores in the training set.

4. Experiments

4.1. Experimental setup

4.1.1. Datasets

Following [2], the proposed method was evaluated on the Voice Conversion Challenge 2018 and 2020 (VCC2018 and VCC2020) datasets. In each challenge, participants submitted audio files produced by their VC systems. Subsequently, subjective listening tests were conducted to evaluate these systems. Subjects were asked to rate the converted utterances based on their quality and similarity to the reference utterances. This study focuses on similarity assessment.

In the VCC2018 dataset [3], the utterances were derived from 36 VC systems and two reference systems. A total of 21,562 converted-natural utterance pairs were evaluated, yielding 30,864 speaker similarity scores ranging from 1 to 4. Higher scores indicate that the speakers in an utterance pair sound more similar to each other. we split the dataset into training and test sets, with 24,864 and 6,000 ratings, respectively.

In VCC2020 [4], there were two tasks, intra-lingual semi-parallel VC and cross-lingual VC. Of the 33 participants, 31 teams submitted results for the intra-lingual task, and 28 teams

submitted results for the cross-lingual task. There were 5,840 converted-target utterance pairs. Each pair was evaluated by multiple subjects, and the average score was used as the final score for the pair. Including source-target and target-target utterance pairs as lower and upper performance bounds, the full VCC2020 test set contains 6,090 scored pairs.

4.1.2. Evaluation metrics

Performance was evaluated in terms of linear correlation coefficient (LCC) [21], Spearman's rank correlation coefficient (SRCC) [22], and mean squared error (MSE) at the utterance and system levels. The utterance-level evaluation was calculated based on the predicted score and the human-labeled score for each test utterance pair, while the system-level evaluation was calculated based on the average predicted score and average human-labeled score for each system. System-level evaluation is more valuable because it directly ranks systems.

4.1.3. Pre-trained models

We evaluated several SFMs. WavLM-Large [8], wav2vec 2.0-Large (LV-60) [6], MMS-300M, and MMS-1B [9] were obtained from their GitHub websites. HuBERT-Large and HuBERT-XLarge [7] were accessed via the pipelines subpackage of *TorchAudio* [23, 24]. Whisper-Medium and Whisper-Large [10] were taken from HuggingFace's *Transformers* library [25]. Note that all SFMs are versions without any fine-tuning. Furthermore, Whisper's original settings involve preprocessing the input waveform into 30-second chunks via zero-padding or trimming. Since all other SFMs use full-length audio inputs and each utterance in the VCC2018 and VCC2020 datasets is shorter than 10 seconds, we modified Whisper's configuration for a fair comparison. Specifically, the chunk length was set to 10 seconds, and the resulting representation was trimmed back to the original length in the time domain.

4.1.4. Training details

All models were implemented using PyTorch (v2.0.1) in Python 3.10. Each model was trained using an NVIDIA GeForce RTX 3090 with 24GB RAM. All utterances were downsampled to a sampling rate of 16 kHz. The linear layer after SFM has a hidden size of 256, and the first linear layer in the prediction module has a hidden size of 128. The output size is 1 for regression tasks and 4 for classification tasks. The Adam optimizer [26] was used to train the models with a learning rate of 1e-4. For regression tasks, we used the MSE loss for training, while for classification tasks, we used the cross entropy (CE) loss. Model parameters were initialized by the default method of PyTorch. We trained each model for 30 epochs on the VCC2018 training set with a batch size of 5, and evaluated the performance of the model for each epoch using the VCC2018 test set. The model with the best system-level performance on the VCC2018 test set was selected and tested on the VCC2020 test set.

4.2. Results

4.2.1. SVSNet+ with WavLM-Large

In SVSNet [2], the input waveform is encoded by an encoder composed of SincNet, rSWC and BLSTM modules. To enhance SVSNet, we integrate WavLM-Large [8] into it and verify the necessity of modules in the original encoder. The results corresponding to the best system-level performance for each combination on the VCC2018 test set are shown in Table 1. We trained two types of SVSNet models, regression and classification, as baselines, termed SVSNet(R) and SVSNet(C), respectively. There are discrepancies between the reproduced models and those in [2], which may be caused by software version and hyperparameter differences. Our proposed models are termed SVSNet+. For in-depth analysis, we implemented the following operations: (1) whether an additional linear layer for dimension adjustment is used; (2) whether WavLM-Large is fine-tuned for the speaker voice similarity assessment task; (3) whether the weighted sum (WS) of representations from all transformer encoder layers or only the representation from the last layer (LL) is used.

From Table 1, several observations can be drawn. First, when integrated with WavLM-Large, SVSNet+ outperforms SVSNet in system-level evaluation in most configurations. Second, the rSWC and BLSTM modules used in SVSNet do not bring notable benefits to SVSNet+ (SVSNet+_rBPL(R) vs. SVSNet+_BPL(R) and SVSNet+_BPL(R) vs. SVSNet+_PL(R)). The reason may be that the transformer encoder in WavLM-Large is already good enough at capturing contextual information in the waveform. Third, fine-tuning WavLM-Large during SVSNet+ training does not help (SVSNet+_PLFW(R) vs. SVSNet+_PLW(R) and SVSNet+_PFW(R) vs. SVSNet+_PW(R)). This may be due to the inappropriateness of fine-tuning large pre-trained models with small datasets. Such phenomenon was also mentioned in [27]. Fourth, utilizing the weighted-sum representation of WavLM-Large can improve performance compared to using the last-layer representation (SVSNet+_PW(R) vs. SVSNet+_P(R)). Lastly, although the additional linear layer may not provide significant advantages to SVSNet+ (SVSNet+_PLW(R) vs. SVSNet+_PW(R)), it is still valuable for representation resizing, allowing a fairer comparison between SVSNet+ and SVSNet. Among all models, the two best-performing models are SVSNet+_PLW(R) and SVSNet+_PW(R), which are regression types with weighted sum and no fine-tuning. Therefore, these two SVSNet+ configurations will be used in subsequent experiments.

4.2.2. SVSNet+ with different SFMs

Next, we compare the performance of SVSNet+ with different SFMs, including WavLM-Large, wav2vec 2.0-Large, HuBERT-Large, HuBERT-XLarge, MMS-300M, MMS-1B, Whisper-Medium, and Whisper-Large. The results for the models with and without an additional linear layer are shown in Tables 2 and 3, respectively. The number of transformer encoder layers, embedding dimension, and number of attention heads are noted after each model. From Table 2, it can be found that SVSNet+_HuBERT-Large achieves the best performance in system-level LCC and SRCC, followed by SVSNet+_Whisper-Large. In terms of system-level MSE, all models show excellent and almost equivalent performance, although SVSNet+_wav2vec 2.0-Large performs slightly better than the others. No matter which SFM is employed, the proposed SVSNet+ consistently outperforms the original SVSNet in all system-level metrics. The results reconfirm the advantage of integrating pre-trained SFMs to extract speech representations. Furthermore, it's worth noting that larger SFMs do not always result in better performance. This may be attributed to the difference in characteristics between the training speech of the upstream model and the test speech of the downstream task.

Comparing Table 3 with Table 2, it can be seen that removing the additional linear layer results in poorer performance in system-level metrics for most models. Only SVSNet+_WavLM-

Table 1: *Performance of different configurations of SVSNet+ with WavLM-Large evaluated on VCC2018. Modules included in SVSNet+ are marked after underline, where r, B, P, L, F, and W denote rSWC, BLSTM, pre-trained model, linear layer, fine-tuning, and weighted sum, respectively.*

Method	SincNet	rSWC	BLSTM	Pre-trained Model	Linear Layer	Fine-tuning	Pre-trained Representation Type	Model Type	Utterance-level			System-level		
									LCC ↑	SRCC ↑	MSE ↓	LCC ↑	SRCC ↑	MSE ↓
SVSNet(R)	O	O	O	X	X	X	-	Regression	0.530	0.531	0.952	0.924	0.902	0.009
SVSNet(C)	O	O	O	X	X	X	-	Classification	0.540	0.542	1.160	0.914	0.874	0.032
SVSNet+_BP(R)	X	X	O	O	X	X	LL	Regression	0.546	0.547	0.899	0.952	0.938	0.005
SVSNet+_BPL(R)	X	X	O	O	O	X	LL	Regression	0.521	0.524	0.994	0.952	0.950	0.015
SVSNet+_BPL(C)	X	X	O	O	O	X	LL	Classification	0.484	0.488	1.358	0.932	0.919	0.028
SVSNet+_rBPL(R)	X	O	O	O	O	X	LL	Regression	0.523	0.524	0.966	0.926	0.894	0.008
SVSNet+_PL(R)	X	X	X	O	O	X	LL	Regression	0.546	0.548	0.902	0.943	0.953	0.006
SVSNet+_PLF(R)	X	X	X	O	O	O	LL	Regression	0.498	0.502	1.022	0.941	0.923	0.007
SVSNet+_PLW(R)	X	X	X	O	O	X	WS	Regression	0.567	0.567	0.879	0.953	0.957	**0.005**
SVSNet+_PLFW(R)	X	X	X	O	O	O	WS	Regression	0.537	0.540	0.921	0.937	0.927	0.009
SVSNet+_P(R)	X	X	X	O	X	X	LL	Regression	0.506	0.501	0.950	0.867	0.887	0.015
SVSNet+_PF(R)	X	X	X	O	X	O	LL	Regression	0.497	0.496	1.010	0.951	0.929	0.005
SVSNet+_PW(R)	X	X	X	O	X	X	WS	Regression	**0.578**	**0.577**	**0.846**	**0.960**	**0.969**	0.009
SVSNet+_PFW(R)	X	X	X	O	X	O	WS	Regression	0.554	0.554	1.062	0.946	0.949	0.129

Large slightly improves over its counterpart with the additional linear layer. Since WavLM jointly learned masked speech prediction and denoising on mixed audio during pre-training, the additional linear layer may blur the extracted speech features, potentially harming the performance of downstream tasks.

Table 2: *Performance of SVSNet+ with different SFMs (with the additional linear layer) evaluated on VCC2018.*

Method	Utterance-level			System-level		
	LCC ↑	SRCC ↑	MSE ↓	LCC ↑	SRCC ↑	MSE ↓
SVSNet(R)	0.530	0.531	0.952	0.924	0.902	0.009
SVSNet+_WavLM-Large [24/1024/12]	0.567	0.567	0.879	0.953	0.957	0.005
SVSNet+_wav2vec 2.0-Large [24/1024/16]	0.573	0.573	0.856	0.958	0.943	**0.004**
SVSNet+_HuBERT-Large [24/1024/16]	0.569	0.570	0.866	**0.967**	**0.960**	0.006
SVSNet+_HuBERT-XLarge [48/1280/16]	**0.581**	**0.578**	**0.842**	0.953	0.955	0.005
SVSNet+_MMS-300M [24/1024/16]	0.524	0.525	0.959	0.954	0.943	0.006
SVSNet+_MMS-1B [48/1280/16]	0.575	0.573	0.843	0.950	0.958	0.006
SVSNet+_Whisper-Medium [24/1024/16]	0.506	0.512	1.022	0.945	0.941	0.007
SVSNet+_Whisper-Large [32/1280/20]	0.561	0.564	0.876	0.961	0.956	0.006

Table 3: *Performance of SVSNet+ with different SFMs (without the additional linear layer) evaluated on VCC2018.*

Method	Utterance-level			System-level		
	LCC ↑	SRCC ↑	MSE ↓	LCC ↑	SRCC ↑	MSE ↓
SVSNet(R)	0.530	0.531	0.952	0.924	0.902	0.009
SVSNet+_WavLM-Large [24/1024/12]	**0.578**	**0.577**	**0.846**	0.960	**0.969**	0.009
SVSNet+_wav2vec 2.0-Large [24/1024/16]	0.572	0.571	0.848	0.926	0.902	0.008
SVSNet+_HuBERT-Large [24/1024/16]	0.566	0.564	0.866	**0.964**	0.959	**0.004**
SVSNet+_HuBERT-XLarge [48/1280/16]	0.563	0.562	0.864	0.939	0.931	0.007
SVSNet+_MMS-300M [24/1024/16]	0.560	0.555	0.869	0.943	0.941	0.006
SVSNet+_MMS-1B [48/1280/16]	0.561	0.558	0.879	0.939	0.929	0.007
SVSNet+_Whisper-Medium [24/1024/16]	0.554	0.553	0.892	0.951	0.949	0.013
SVSNet+_Whisper-Large [32/1280/20]	0.553	0.553	0.906	0.956	0.938	0.006

4.2.3. Evaluated on VCC2020

To evaluate the generalization ability of SVSNet+, the above models trained on VCC2018 were tested on the VCC2020 test set. The system-level evaluation results are shown in Table 4. Unlike VCC2018, VCC2020 consists of two tasks: intra-lingual semi-parallel VC and cross-lingual VC. Moreover, in VCC2018, most VC systems employed conventional vocoders, while in VCC2020, neural vocoders were more common. All of the above differences can lead to serious corpus mismatch. Comparing Table 4 with Tables 2 and 3, due to corpus mismatch, the scores of all models on VCC2020 are worse than those reported on VCC2018. However, certain SVSNet+ models, such as SVSNet+_HuBERT-Large and SVSNet+_Whisper-Large, achieve fairly good performance on VCC2020, although there is still room for further improvement.

Examining the performance of SVSNet+ models with the additional linear layer in Table 4, we can see that almost all SVSNet+ models outperform SVSNet in all metrics except SVSNet+_Whisper-Medium in MSE (1.175). The SVSNet+_HuBERT-Large and SVSNet+_Whisper-Large models with the best performance on VC2018 in Table 2 also achieve relatively high performance on VCC2020. Surprisingly, SVSNet+_wav2vec 2.0-Large performs very well and achieves the highest SRCC (0.910), although it does not perform particularly well compared to other models on VCC2018.

For SVSNet+ models without the additional linear layer, while most models outperform SVSNet, SVSNet+_wav2vec 2.0-Large performs poorly in both LCC and SRCC. Without further fine-tuning the wav2vec 2.0 representation using the additional linear layer, the resulting SVSNet+ model generalizes poorly. In contrast, removing the additional linear layer benefits SVSNet+_MMS-1B, which achieves the highest scores in LCC and SRCC among all models with the same settings. Since MMS-1B was pre-trained using a larger and more diverse set of data, it excels at extracting more intricate patterns. Additional processing by the linear layer may be detrimental to the extracted features.

Table 4: *Performance of SVSNet+ with different SFMs evaluated on VCC2020. Each model is the best-performing checkpoint selected from the VCC2018 system-level results.*

Method	System-level					
	w/ linear layer			w/o linear layer		
	LCC ↑	SRCC ↑	MSE ↓	LCC ↑	SRCC ↑	MSE ↓
SVSNet(R)	0.745	0.713	0.764	0.745	0.713	0.764
SVSNet+_WavLM-Large [24/1024/12]	0.768	0.766	0.535	0.822	0.851	0.402
SVSNet+_wav2vec 2.0-Large [24/1024/16]	0.885	**0.910**	0.502	0.694	0.630	0.526
SVSNet+_HuBERT-Large [24/1024/16]	0.854	0.844	0.487	0.831	0.806	0.227
SVSNet+_HuBERT-XLarge [48/1280/16]	0.848	0.840	0.499	0.760	0.746	0.395
SVSNet+_MMS-300M [24/1024/16]	0.803	0.775	0.571	0.770	0.732	0.486
SVSNet+_MMS-1B [48/1280/16]	0.794	0.764	**0.310**	**0.865**	**0.852**	0.339
SVSNet+_Whisper-Medium [24/1024/16]	0.771	0.775	1.175	0.838	0.808	**0.224**
SVSNet+_Whisper-Large [32/1280/20]	**0.892**	0.886	0.319	0.859	0.835	0.368

5. Conclusions

This study demonstrates that representations extracted by SFMs can effectively enhance the performance of speaker voice similarity assessment models. Experiments conducted on VCC2018 and VCC2020 show that SVSNet+ leveraging SFM surpasses its predecessor SVSNet. The results also show that for different SFMs, an additional linear layer can have significantly different effects on the performance of assessment models. Along this research path, in addition to integrating a single SFM into SVSNet+, we also conducted preliminary experiments on fused SFMs. By concatenating the representations extracted by HuBERT-Large and Whisper-Large as input, SVSNet+ can achieve better system-level performance on VCC2018, with LCC, SRCC, and MSE of 0.97, 0.969, and 0.004, respectively. We will conduct further research in this direction in the future.

6. Acknowledgements

This work was supported in part by the Co-creation Platform of the Speech-AI Research Center, Industry-Academia Innovation School, NYCU, under the framework of the National Key Fields Industry-University Cooperation and Skilled Personnel Training Act, from the Ministry of Education (MOE), the National Development Fund (NDF), and industry partners in Taiwan.

7. References

[1] R. K. Das, T. Kinnunen, W.-C. Huang, Z.-H. Ling, J. Yamagishi, Y. Zhao, X. Tian, and T. Toda, "Predictions of subjective ratings and spoofing assessments of Voice Conversion Challenge 2020 submissions," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge*, 2020, pp. 99–120.

[2] C.-H. Hu, Y.-H. Peng, J. Yamagishi, Y. Tsao, and H.-M. Wang, "SVSNet: An end-to-end speaker voice similarity assessment model," *IEEE Signal Processing Letters*, vol. 29, pp. 767–771, 2022.

[3] 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," in *Proc. The Speaker and Language Recognition Workshop (Odyssey)*, 2018, pp. 195–202.

[4] Y. Zhao, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Kinnunen, Z. Ling, and T. Toda, "Voice Conversion Challenge 2020: Intra-lingual semi-parallel and cross-lingual voice conversion," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge*, 2020, pp. 80–98.

[5] S. Liu, A. Mallol-Ragolta, E. Parada-Cabaleiro, K. Qian, X. Jing, A. Kathan, B. Hu, and B. W. Schuller, "Audio self-supervised learning: A survey," *Patterns*, vol. 3, no. 12, 2022.

[6] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 12 449–12 460.

[7] 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.

[8] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[9] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi, A. Baevski, Y. Adi, X. Zhang, W.-N. Hsu, A. Conneau, and M. Auli, "Scaling speech technology to 1,000+ languages," *arXiv preprint arXiv:2305.13516*, 2023.

[10] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[11] K.-H. Hung, S.-w. Fu, H.-H. Tseng, H.-T. Chiang, Y. Tsao, and C.-W. Lin, "Boosting self-supervised embeddings for speech enhancement," in *Proc. Interspeech*, 2022, pp. 186–190.

[12] T. Nguyen, S. Mdhaffar, N. Tomashenko, J.-F. Bonastre, and Y. Estève, "Federated learning for ASR based on wav2vec 2.0," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[13] Z. Chen, S. Chen, Y. Wu, Y. Qian, C. Wang, S. Liu, Y. Qian, and M. Zeng, "Large-scale self-supervised speech representation learning for automatic speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6147–6151.

[14] H.-H. Chen, Y.-L. Chien, M.-C. Yen, S.-W. Tsai, T.-s. Chi, H.-M. Wang, and Y. Tsao, "Mandarin electrolaryngeal speech voice conversion using cross-domain features," in *Proc. Interspeech*, 2023, pp. 5018–5022.

[15] E. Cooper, W.-C. Huang, T. Toda, and J. Yamagishi, "Generalization ability of MOS prediction networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8442–8446.

[16] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "UTMOS: UTokyo-SaruLab system for VoiceMOS Challenge 2022," in *Proc. Interspeech*, 2022, pp. 4521–4525.

[17] W.-C. Tseng, C.-y. Huang, W.-T. Kao, Y. Y. Lin, and H.-y. Lee, "Utilizing self-supervised representations for MOS prediction," in *Proc. Interspeech*, 2021, pp. 2781–2785.

[18] 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, pp. 5443–5447.

[19] R. E. Zezario, Y.-W. Chen, S.-W. Fu, Y. Tsao, H.-M. Wang, and C.-S. Fuh, "A study on incorporating Whisper for robust speech assessment," *arXiv preprint arXiv:2309.12766*, 2023.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[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] Y.-Y. Yang, M. Hira, Z. Ni, A. Astafurov, C. Chen, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Hwang, J. Chen, P. Goldsborough, S. Narenthiran, S. Watanabe, S. Chintala, and V. Quenneville-Bélair, "TorchAudio: Building blocks for audio and speech processing," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6982–6986.

[24] J. Hwang, M. Hira, C. Chen, X. Zhang, Z. Ni, G. Sun, P. Ma, R. Huang, V. Pratap, Y. Zhang, A. Kumar, C.-Y. Yu, C. Zhu, C. Liu, J. Kahn, M. Ravanelli, P. Sun, S. Watanabe, Y. Shi, and Y. Tao, "TorchAudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for PyTorch," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–9.

[25] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," in *Proc. Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[27] B. Tamm, H. Balabin, R. Vandenberghe, and H. Van hamme, "Pretrained speech representations as feature extractors for speech quality assessment in online conferencing applications," in *Proc. Interspeech*, 2022, pp. 4083–4087.