AN ADAPTER BASED MULTI-LABEL PRE-TRAINING FOR SPEECH SEPARATION AND ENHANCEMENT

Tianrui Wang\textsuperscript{1}, Xie Chen\textsuperscript{2,3}, Zhuo Chen\textsuperscript{4}, Shu Yu\textsuperscript{2}, Weibin Zhu\textsuperscript{1}

\textsuperscript{1}Institute of Information Science, Beijing Jiaotong University, Beijing, China
\textsuperscript{2}MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China
\textsuperscript{3}Peng Cheng Laboratory, Shenzhen, China
\textsuperscript{4}Microsoft, One Microsoft Way, Redmond, WA, USA

ABSTRACT
In recent years, self-supervised learning (SSL) has achieved tremendous success in various speech tasks due to its power to extract representations from massive unlabeled data. However, compared with tasks such as speech recognition (ASR), the improvements from SSL representation in speech separation (SS) and enhancement (SE) are considerably smaller. Based on HuBERT, this work investigates improving the SSL model for SS and SE. We first update HuBERT’s masked speech prediction (MSP) objective by integrating the separation and denoising terms, resulting in a multiple pseudo label pre-training scheme, which significantly improves HuBERT’s performance on SS and SE but degrades the performance on ASR. To maintain its performance gain on ASR, we further propose an adapter-based architecture for HuBERT’s Transformer encoder, where only a few parameters of each layer are adjusted to the multiple pseudo label MSP while other parameters remain frozen as default HuBERT. Experimental results show that our proposed adapter-based multiple pseudo label HuBERT yield consistent and significant performance improvements on SE, SS, and ASR tasks, with a faster pre-training speed, at only marginal parameters increase.

Index Terms—Self-Supervised Learning, Speech Separation, Speech Enhancement, Adapter

1. INTRODUCTION
Self-supervised learning (SSL) has recently become a fast-growing topic in speech processing [1, 2]. Different from supervised learning, the supervision of SSL is derived from the data itself, which allows the models to leverage massive unlabeled data and thus achieve significant improvements in various tasks [3–8]. A typical SSL approach for speech processing includes two stages. Firstly, an upstream model is pre-trained based on the unlabeled data. Secondly, a small set of supervised data is employed to fine-tune the network for each downstream task by updating the entire upstream model or adding task-specific layers [9]. In this way, each downstream task can be benefited from the extended data scale, resulting in improved performance. The SSL systems demonstrated their clear performance advantages over the supervised baselines in the SUPERB [10] and SUPERB-SG [11] benchmarks that cover evaluations for 15 speech tasks.

However, compared with other speech tasks, the SSL models perform mediocrely on speech separation (SS) and enhancement (SE) [11]. One potential reason for this performance discrepancy lies in the mismatches between the pre-training scheme and their fine-tuning targets. Although features of shallow encoder layers and the introduction of additional acoustic features have been verified to be helpful for SE [12], the mismatches caused by SSL objective and data configuration have not been essentially addressed. SS aims to separate overlapped speech into each speech component [13], and SE aims to extract speech from a noisy environment [14, 15]. In contrast, most SSL methods are only pre-trained on clean speech with a single active speaker. Therefore, the data domain mismatch is the first barrier to SS and SE. Although [16] verified that data augmentation could reduce the data domain mismatch to improve the performance on out-of-domain ASR, no such exploration was investigated for SS and SE tasks. On the other hand, most SSL training targets recovering only one speaker from input audio [17, 18], which could be sub-optimal for tasks such as SS, where all involved speakers are equally important. We refer to this as a task mismatch. Moreover, considering the different modeling requirements for separation tasks (SS, SE, etc.) and semantic tasks (ASR, etc.), it is challenging for one universal SSL upstream model to achieve the optimum performance in both [19].

To alleviate the data domain and task mismatches, we introduce two extensions to HuBERT [7] in this study. Firstly, a data augmentation that includes raw speech, noisy speech, overlapped speech, and noisy overlapped speech is employed for pre-training. And a multiple pseudo label MSP loss is proposed to integrate separation and denoising into pre-training.
In this way, HuBERT’s performances are significantly improved on SS and SE but degraded on ASR. Secondly, the adapter layers are further integrated into the proposed system to address this degradation. With the adapter-based architecture and the multiple-label MSP, we show that satisfactory performances in all three tasks can be obtained by only updating a few new parameters from the HuBERT baseline.

2. PRELIMINARIES

2.1. HuBERT

HuBERT [7] is an SSL method which benefits from an offline clustering to generate pseudo labels \( Z = [z_1, z_2, \cdots, z_T] \), where each \( z \) is a \( C \)-class categorical variable for a BERT-like pre-training [20], as shown in Fig. 1. A convolutional module (CNN) \( f(\cdot) \) transforms the waveform into the frame-level feature \( X = [x_1, x_2, \cdots, x_T] \) with a stride size of 20ms. Then, the frame-level feature is encoded by the Transformer into the representation \( O = [o_1, o_2, \cdots, o_T] \). During pre-training, the frame-level features are masked randomly and successively, then fed to Transformer, the model is finally trained to predict the pseudo labels of the masked frames. The distribution over codewords is parameterized as,

\[
p(c|o_t) = \frac{\exp(\text{sim}(A o_t, e_c)/\tau)}{\sum_c \exp(\text{sim}(A o_t, e_c)/\tau)}
\]

(1)

where \( A \) is a projection matrix, \( e_c \) is the embedding for codeword \( c \), \( \text{sim}(\cdot, \cdot) \) computes the cosine similarity between two vectors, and \( \tau \) scales the logit which is set to 0.1. The prediction loss is only applied over the masked frames.

2.2. Adapter Based Transfer Learning

Adapters are lightweight modules inserted in layers of a deep neural network [21, 22], and only the inserted adapters need to be trained for a new task. In this way, adapter-based training can retain the original capabilities of the model while being competent in a new job [19, 23]. The fully connected bottleneck adapter is widely employed in BERT for various tasks [22]. We use these adapters for HuBERT, which will be detailed in section 3.2.

3. PROPOSED METHOD

Our proposed method consists of two parts. First, the multiple pseudo label based on the data augmentation is proposed to alleviate the data domain and task mismatches. Second, the adapters are employed to learn the representations for SS and SE while retaining the original task performance.

3.1. Multiple Pseudo Label based HuBERT

The proposed multiple pseudo label (multi-label) based HuBERT is pre-trained to predict the pseudo labels of speech components from the given input that includes overlapped and noisy utterances. This scheme applies to the situation of any overlapped speakers, and the 2-speaker condition is our major focus, as shown in Fig. 2.

![Fig. 2 Diagram of the proposed multi-label HuBERT](image)

Four kinds of audio are applied for pre-training, raw speech, noisy speech, overlapped speech, and noisy overlapped speech. Following HuBERT’s clustering, each speech component of the mixtures is annotated with the pseudo labels \( Z \), resulting in \( Z \in \mathbb{R}^{N \times T} = [Z_1, Z_2, \cdots, Z_N] \), where \( N \) is the number of speech involved in the pre-training data.

Following Eq. (1), during pre-training, two series of labels \( Z^1, Z^2 \) are predicted by two pairs of codewords and projection matrix, respectively. The targets of raw and noisy speech are set respectively as the label of speech and \( 0 \) to mimic denoising (\( Z^1 = Z_n, Z^2 = 0 \)). The targets of overlapped and noisy overlapped speech are set as the labels of speech components to mimic separation (\( Z^1 = Z_{n_1}, Z^2 = Z_{n_2} \)), where the average energy of speech \( n_1 \) is greater than that of \( n_2 \), avoiding the permutation problem during training [24].

3.2. Adapter Based HuBERT

Adapter-based HuBERT learns the representations for SS and SE by the inserted adapter modules, and other parameters re-
main frozen as default HuBERT, as shown in Fig. 3. In the inference phase, route II with specific-task adapters can be adopted for SS and SE. Route I will lead to the standard HuBERT and can be applied to the original tasks, such as ASR.

Fig. 3. The architecture of the adapter-based Transformer blocks. Two adapters with LNs are inserted into the blocks to learn the new representations for SS and SE. The modules with dashed are frozen during pre-training. Two inferencing routes can lead to different performances during evaluation.

Our HuBERT is initialized with the open pre-trained model at first. Then the bottleneck adapter layers with near-identity initialization [22] are inserted after the multi-head self-attention (MHSA) and feedforward layer. The adapter consists of layer normalization (LN), fully-connect layer (FC), and ReLU, which first projects the d-dimensional feature into a smaller dimension with nonlinearity, then projects back to d dimensions. The different features between mixed and clean speech will substantially vary the parameters of LNs. So the adapter uses a new LN layer, which is not shared with the original task. In other words, the parameters of the original HuBERT are entirely frozen, and only the inserted adapters and LNs are pre-trained for multi-label MSP.

4. EXPERIMENTS

4.1. Dataset

We conducted the experiments on the data from the LibriSpeech [25], WHAM! [26], and DNS-Challenge [27] datasets. All the audio is sampled at 16KHz.

Pre-training. A total of 3840 hours of training and validation sets (Librispeech960 and three-fold simulated data) were prepared from LibriSpeech’s speech and DNS-Challenge’s noise. The ratio of raw speech, noisy speech, overlapped speech, and noisy overlapped speech was 1:1:1:1. The speech was randomly mixed with another one at a signal noise ratio (SNR) within 1dB to 6dB for an overlapped speech (the SNR represents the ratio of speech $n_1$ to speech $n_2$ mentioned in Section 3.1). The noise was randomly mixed with speech or overlapped speech at SNRs within -5dB to 15dB.

SS fine-tuning Following SUPERB-SG [11], a total of 50 hours of training, validation, and test sets were simulated in Section 3.1. The noise was randomly mixed with speech or overlapped speech at SNRs within -5dB to 15dB.

4.2. Model and Training Setup

The HuBERT Base [7] was selected as the baseline and the backbone. The data-augmentation pre-training (D-A) [16] and primary-speaker denoising pre-training (PS-D) [17] were employed as the comparison methods.

Backbone. The convolutional module consists of seven 512-channel layers with strides [5, 2, 2, 2, 2, 2, 2] and kernels [10, 3, 3, 3, 3, 3, 2, 2]. The Transformer contains 12 layers with 768 dimensions, 3072 inner dimensions, and 12 attention heads. Moreover, the encoder-post module utilizes the FC and code embedding layer to predict masked speech codes.

The HuBERT of D-A, PS-D, and our methods were initialized with the official HuBERT. Then the D-A, PS-D, and multi-label HuBERT (ML) without adapters were pre-trained for 450K steps on 8 GPUs with a batch size of 150 seconds samples per GPU. The multi-label adapter-based HuBERT was pre-trained for 337.5K steps with a batch size of 200 seconds samples per GPU. Adam optimizer was used with the warm-up learning rate, which ramped up linearly from 0 to 3e-4 for the first 8% steps and then decayed linearly to 0.

For SS, a 3-layer 896-cell BiLSTM [28] with the 2-speaker ideal non-negative phase sensitive mask [29] was adopted and optimized by the permutation-invariant [24] based mean square error. For SE, a same BiLSTM as SS with the complex mask [30] was employed and optimized by the scale-invariant SNR [13]. For ASR, a 2-layer BiLSTM with 1024 cells was used and optimized by the character-unit CTC loss. The SUPERB-SG-style fine-tuning [11] was firstly executed for three tasks. And then, we fine-tuned the pre-trained and downstream models together for SS and SE, respectively.

4.3. SUPERB-SG-style Fine-tuning

Following the SUPERB-SG, the fine-tuning with the frozen pre-trained model was executed to evaluate the model’s universality on SS, SE, and ASR. The outputs of all Transformer layers were weighted with learnable weights and summed as the input to the downstream model. The evaluation metrics for SS and SE are the scale-invariant signal-to-distortion ratio (SI-SDR) [13], perceptual evaluation of speech quality...
(PESQ) [31], and short-time objective intelligibility (STOI) [32]. The word error rate (WER) is counted for ASR.

As shown in Table 1, compared to the Fbank, significant improvements in ASR but limited ones on SS and SE are brought by the original HuBERT. Based on HuBERT, the target domain data augmentation and PS-D improve the performance on SS and SE. But the absence of multi-speaker information limits their improvements on SS. Unsurprisingly, our proposed multi-label pre-training achieves the best results on SS and SE (SI-SDR:10.59/PESQ:2.08/STOI:89.20), which can retain multiple-speaker information while extracting targets from a mixture. However, compared to the original HuBERT, the WER on ASR of multi-label pre-training relatively increases by 40%. We speculate that multiple-information retaining is adverse to single-speaker ASR.

The degradation of the original tasks is undesired. As can be seen from the results, based on multi-label pre-training, the adapter-based methods with a few pre-trained parameters achieve comparable performance to the full pre-training on SS and SE, and the degradation on ASR is substantially suppressed. We speculate that training only a few additional parameters for a new task can reduce the bias introduced by the new task. In this way, the model can leverage its prior knowledge to learn a new task. As expected, inferencing without adapters performs the same as the original HuBERT on ASR.

### 4.4. Fine-tuning for SS or SE

The SSL models were frozen and operated in a constrained condition in previous experiments. Fine-tuning experiments with the unfrozen SSL models for SS and SE were conducted to evaluate our proposed method under a more realistic condition. As shown in Table 2, the overall results are consistent with Table 1, which shows that multi-label pre-training performs excellently on SS and SE. As the adapter size increases, the performance of adapter-based multi-label pre-training gradually approaches the performance of full pre-training.

### 4.5. Consumed Time for Pre-training

The consumed time per epoch of different methods was counted, as shown in Table 3. The adapter-based pre-training is 19% relative faster than the full pre-training. We believe that the adapter-based pre-training may be a practical choice to evaluate a pre-training method, which converges faster with less GPU memory (less trainable parameters) and performs consistently with full pre-training.

### 5. CONCLUSION

We believe that the data domain mismatch and task mismatch are the culprits for the mediocre performance of SSL features on SS and SE. In this study, we propose a multi-label pre-training scheme to introduce separation and denoising into the masked prediction loss of HuBERT, and its significant improvements on SS and SE are experimentally verified. However, the differences in task requirements degrade its performance on other tasks, such as ASR. Therefore, the lightweight adapters are further inserted into HuBERT’s Transformer to be pre-trained for SS and SE. In this way, comparable improvements on SS and SE can be achieved while retaining outstanding performance on the original tasks.
6. REFERENCES

[1] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, “An unsupervised autoregressive model for speech representation learning,” in INTERSPEECH, 2019, pp. 146–150.

[2] A. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” arXiv preprint arXiv:1807.03748, 2018.

[3] A. T Liu, S.-W. Li, and H.-Y. Lee, “Tera: Self-supervised learning of transformer encoder representation for speech,” TASLP, vol. 29, pp. 2351–2366, 2021.

[4] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised pre-training for speech recognition,” in INTERSPEECH, 2019, pp. 3465–3469.

[5] A. Baevski, S. Schneider, and M. Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” in ICLR, 2019.

[6] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in NeurIPS, 2020, vol. 33, pp. 12449–12460.

[7] W.-N. Hsu, B. Bolte, Y. H. Tsai, K. Lakkotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” TASLP, vol. 29, pp. 3451–3460, 2021.

[8] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, “Data2vec: A general framework for self-supervised learning in speech, vision and language,” arXiv preprint arXiv:2202.03555, 2022.

[9] X. Chang, T. Maekaku, Y. Fujita, and S. Watanabe, “End-to-end integration of speech recognition, speech enhancement, and self-supervised learning representation,” arXiv preprint arXiv:2204.00540, 2022.

[10] S.-W. Yang, P.-H. Chi, Y.-S. Chuang, C.-I Lai, K. Lakkotia, Y. Y Lin, A. T Liu, J. Shi, X. Chang, G.-T. Lin, et al., “Superb: Speech processing universal performance benchmark,” arXiv preprint arXiv:2105.10151, 2021.

[11] H.-S. Tsai, H.-J. Chang, W.-C. Huang, Z. Huang, K. Lakkotia, S.-W. Yang, S. Dong, A. Liu, C.-I Lai, J. Shi, et al., “Superb-sg: Enhanced speech processing universal performance benchmark for semantic and generative capabilities,” in ACL, 2022, pp. 8479–8492.

[12] 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,” arXiv preprint arXiv:2204.03339, 2022.

[13] J. R Hershey, Z. Chen, J. Le Roux, and S. Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” in ICASSP. IEEE, 2016, pp. 31–35.

[14] T. Wang, W. Zhu, Y. Gao, J. Feng, and S. Zhang, “Hgen: Harmonic gated compensation network for speech enhancement,” in ICASSP, 2022, pp. 371–375.

[15] T. Wang, W. Zhu, Y. Gao, Y. Chen, J. Feng, and S. Zhang, “Harmonic gated compensation network plus for icassp 2022 dnn challenge,” in ICASSP, 2022, pp. 9286–9290.

[16] W.-N. Hsu, A. Sriram, A. Baevski, T. Likhomanenko, Q. Xu, V. Pratap, J. Kahn, A. Lee, R. Collobert, G. Synnaeve, et al., “Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training,” arXiv preprint arXiv:2104.01027, 2021.

[17] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, et al., “Wavlms: Large-scale self-supervised pre-training for full stack speech processing,” IJSTSP, 2022.

[18] S. Chen, Y. Wu, C. Wang, Z. Chen, Z. Chen, S. Liu, J. Wu, Y. Qian, F. Wei, J. Li, et al., “Unispeech-sat: Universal speech representation learning with speaker aware pre-training,” in ICASSP. IEEE, 2022, pp. 6152–6156.

[19] S. Kessler, B. Thomas, and S. Karout, “An adapter based pre-training for efficient and scalable self-supervised speech representation learning,” in ICASSP. IEEE, 2022, pp. 3179–3183.

[20] K. Lee J. Devlin, M.-W. Chang and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in NAACL, 2019, pp. 4171–4186.

[21] A. Bapna and O. Firat, “Simple, scalable adaptation for neural machine translation,” in EMNLP-IJCNLP, 2019, pp. 1538–1548.

[22] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-efficient transfer learning for nlp,” in ICML. PMLR, 2019, pp. 2790–2799.

[23] B. Thomas, S. Kessler, and S. Karout, “Efficient adapter transfer of self-supervised speech models for automatic speech recognition,” in ICASSP. IEEE, 2022, pp. 7102–7106.

[24] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, “Permutation invariant training of deep models for speaker-independent multilalker speech separation,” in ICASSP. IEEE, 2017, pp. 241–245.

[25] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in ICASSP. IEEE, 2015, pp. 5206–5210.

[26] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. Le R., “Wham!: Extending speech separation to noisy environments,” in INTERSPEECH, 2019, pp. 1368–1372.

[27] C. K. Reddy, V. Gopal, R. Cutler, E. Beyrami, R. Cheng, H. Dubey, S. Matusouvych, R. Aichner, A. Aazami, S. Braun, et al., “The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results,” in INTERSPEECH, 2020, pp. 2492–2496.

[28] A. Graves, N. Jaitly, and A. Mohamed, “Hybrid speech recognition with deep bidirectional lstms,” in ASRU. IEEE, 2013, pp. 273–278.

[29] H. Erdogan, J. R Hershey, S. Watanabe, and J. Le Roux, “Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks,” in ICASSP. IEEE, 2015, pp. 708–712.

[30] Y. Hu, Y. Liu, S. Lv, M. Xing, S. Zhang, Y. Fu, J. Wu, B. Zhang, and L. Xie, “Dccrn: Deep complex convolution using deep recurrent neural networks,” in INTERSPEECH, 2015, pp. 2125–2130.

[31] N. Kanda, T. Yoshioka, X. Xiao, et al., “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” IEEE TASLP, 2021.