ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10448240

AN ADAPTER-BASED UNIFIED MODEL FOR MULTIPLE SPOKEN LANGUAGE PROCESSING TASKS

Varsha Suresh[1], Salah Aït-Mokhtar[2], Caroline Brun[2], Ioan Calapodescu[2]*

[1]National University of Singapore, [2]Naver Labs Europe

ABSTRACT

Self-supervised learning models have revolutionized the field of speech processing. However, the process of fine-tuning these models on downstream tasks requires substantial computational resources, particularly when dealing with multiple speech-processing tasks. In this paper, we explore the potential of adapter-based fine-tuning in developing a unified model capable of effectively handling multiple spoken language processing tasks. The tasks we investigate are Automatic Speech Recognition, Phoneme Recognition, Intent Classification, Slot Filling, and Spoken Emotion Recognition. We validate our approach through a series of experiments on the SUPERB benchmark, and our results indicate that adapter-based fine-tuning enables a single encoder-decoder model to perform multiple speech processing tasks with an average improvement of 18.4 % across the five target tasks while staying efficient in terms of parameter updates.

Index Terms— Spoken Language Processing, Multitask Learning, Adapters, Self-supervised Models

1. INTRODUCTION

The fine-tuning of self-supervised learning (SSL) models, such as wav2vec 2.0 [1], has improved the performance of Spoken Language Processing (SLP) tasks. However, as the quality of representations generated by these models improves, there is a corresponding increase in their size, necessitating additional storage and computational resources. This issue becomes particularly pronounced when dealing with multiple speech-processing tasks, with each target task requiring separate model fine-tuning, further increasing the need for resources.

Modular architectures, such as adapters, have been widely used in NLP to tackle both parameter efficiency and multi-tasking [2]. While adapter-based fine-tuning has been utilized in speech-related tasks, such as speech translation [3, 4, 5] and domain adaptation [6], its efficiency in developing a unified model capable of handling multiple Spoken Language Processing (SLP) tasks remains relatively unexplored. Existing attempts to model multiple SLP tasks with a single model utilises task-specific decoders [7]. However, this approach becomes less scalable as the number of tasks increases.

In this work, we aim to develop a scalable and parameter-efficient unified encoder-decoder model to effectively handle multiple spoken language processing (SLP) tasks. For this, we use adapters [2], which allows new tasks to be added without the need to re-train the entire model and which also mitigates the need for dedicated decoders [7]. Moreover, since adapters facilitate Multi-Task Learning (MTL), we investigate two approaches: Stacking [8] and Fusion [9], in addition to single-task adapters. To evaluate our approach, we choose five speech-processing tasks from the SUPERB benchmark [7]: Automatic Speech Recognition (ASR), Phoneme Recognition (PR), Intent Classification (IC), Slot Filing (SF), and Spoken Emotion Recognition (ER). The detailed model description is provided in Figure 1. From our experiments, we observed that adapter-based fine-tuning outperformed the SUPERB benchmark with an average improvement of 18.4 % achieved across 5 target tasks. We summarise our contributions below:

- We investigate the feasibility and efficiency of using adapters to build a unified encoder-decoder model that can tackle multiple spoken language processing tasks in a simple and scalable manner.
- We explore multi-task learning within our unified framework with two methods: stacking and fusion, which combine adapters to enhance the performance of positively correlated tasks.

2. RELATED WORK

In the field of NLP, researchers have used a single model to handle multiple tasks and adapt them to different domains [10]. In the speech domain, most approaches that deal with multiple tasks fall under multi-task models. They either focus on improving a primary task by using auxiliary tasks, like performing ASR to enhance Emotion Recognition [11, 12, 13], or simultaneously perform multiple tasks —slot-filling and intent classification [14], ASR and speech translation [15, 16] etc. However, these approaches are not easily scalable for new tasks and are mostly applied for tasks that are known to be positively correlated.

*work done during internship at Naver Labs Europe

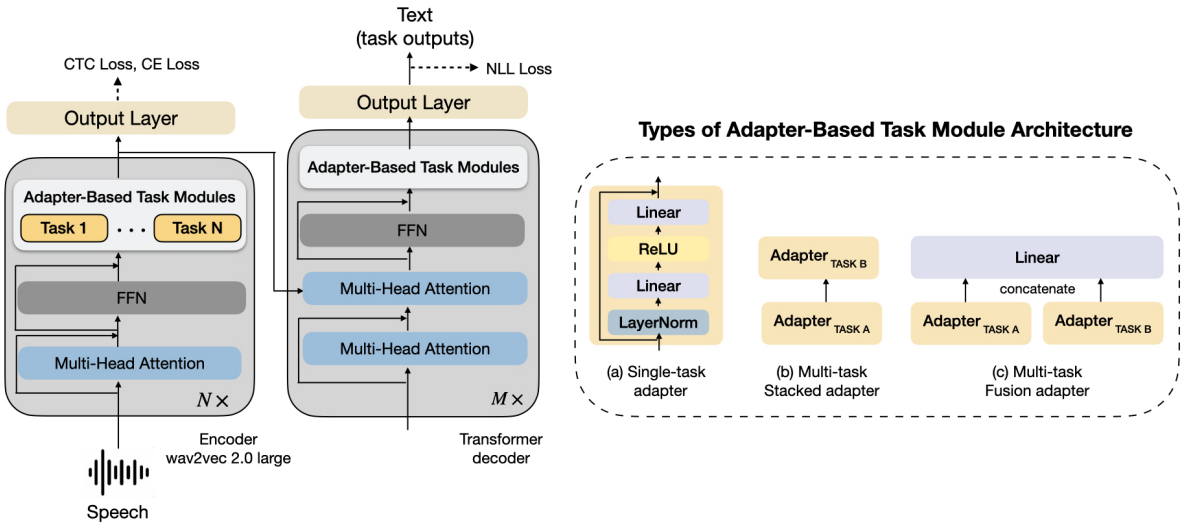

Fig. 1. **Left**: Overall model architecture with a unified encoder-decoder model with adapter-based task modules on each transformer layer. **Right**: Three types of adapter-based task modules used.

Some studies have aimed to create a unified model for multiple speech-processing tasks by training different modules and composing them to perform each task. These architectures comprise encoders and decoders trained to capture features from different modalities, such as text and speech [17], or speech characteristics like prosody [18] etc. In contrast, our approach focuses on constructing task-specific modular architectures. Furthermore, adding a task is straightforward as these task-specific modules are trained independently.

Our work is inspired by SUPERB [7], where multiple tasks are modeled using pre-trained frozen encoders (such as wav2vec 2.0) and task-specific decoders, the task performance relying on the type of decoder used for the task [19]. However, this approach does not scale well as the number of tasks increases. Instead, in our work, we aim to develop a single encoder-decoder model and show that adapters on the decoder side help us adapt to different types of tasks (both classification and generative) without the need for dedicated decoders.

3. MODEL ARCHITECTURE

3.1. Pre-trained Encoder-Decoder model

We use wav2vec 2.0-large[1] as encoder and a 6-layer transformer decoder which is randomly initialized. We fine-tuned this encoder-decoder model on the LibriSpeech 100-hr dataset [20] for the ASR task using hybrid CTC/attention objective [21] as the base model for our unified model. We use SentencePiece (BPE) vocabulary of size 5000. This base model achieved a word error rate (WER) of 3.54 on the test-clean split of LibriSpeech which is comparable to the 3.1 WER reported in the SUPERB benchmark.

3.2. Adapter-based Task Modules

To enable the above-mentioned pre-trained encoder-decoder model to perform multiple SLP tasks, we insert task-specific adapter modules into the transformer [25] layers of both the encoder and the decoder. These modules are depicted in Figure 1. The remainder of the model is frozen.

We focus on three types of architecture: i) single adapter ii) adapter stacking, and iii) adapter fusion as shown on the right side of Figure 1. In the standard setting, a single adapter is trained for each task [2]. However, some SLP tasks are known to benefit from MTL, such as performing ASR and emotion recognition [12, 13] and Intent classification and Slot-filling [26]. As adapters naturally support MTL [2] in addition to using a single adapter per task, we use the adapter stacking [8] and adapter fusion settings (same as the fast fusion setting in [9]) to perform positively correlated tasks together.

To facilitate a unified model, we encompass both classification (e.g., emotion recognition) and generative (e.g., slot filling) SLP tasks into this single encoder-decoder model. To achieve this, we model the classification task as a generative task i.e., the classification labels are generated. To accommodate multiple tasks using a single decoder output, some task-specific tokens are allocated in the vocabulary—slot value for SF task, emotion labels for ER task, etc. These tokens are selected from the least frequently used tokens in the vocabulary. For MTL, we combine the ground truth of the tasks involved using a task separator token. For example, to perform ASR

[1] https://huggingface.co/facebook/wav2vec2-large-lv60

	LibriSpeech		IEMOCAP	SNIPS		FSC	Avg
	ASR (WER ↓)	PR (PER ↓)	ER (Acc % ↑)	IC (Acc % ↑)	SF (F1 ↑,CER ↓)	IC (Acc % ↑)	
WavLM large SUPERB [7]	3.4	3.1	**70.6**	-	92.2, 18.4	99.0	89.5
wav2vec2.0 large SUPERB [7]	**3.1**	4.7	65.6	-	87.1, 27.3	95.2	85.5
wav2vec2.0 large (Ours)	3.5	**2.4**	68.2	**99.1**	**95.4, 11.8**	**99.5**	**90.9**

Table 1. Performance comparison in various speech processing tasks from the SUPERB benchmark. WavLM is currently ranked first in SUPERB's leaderboard and we choose wav2vec2-large to compare multi-decoder (SUPERB) and single decoder (Ours) solutions. Models from SUPERB have different decoder implementations for each task (e.g. Bi-LSTM, CNNs, linear projections) on top of the chosen SSL model. Our approach is a single transformer encoder-decoder model capable of performing all six tasks using various adapters for each task and initialized on the encoder side with the chosen SSL model. The metrics are computed with the s3prl framework and Avg denotes the average performance across all the tasks.

	IEMOCAP		SNIPS			FSC	
	ASR (WER ↓)	ER (Acc % ↑)	ASR (WER ↓)	IC (Acc % ↑)	SF (F1 ↑,CER ↓)	ASR (WER ↓)	IC (Acc % ↑)
MTL: ESP-net [22]	-	67.6	-	91.7	-	-	**99.6**
MTL: ASR+SER [12]	32.7	63.4*	-	-	-	-	-
MTL: ASR+IC [23]	-	-	-	-	-	-	98.2
MTL: ASR+IC [24]	-	-	11.8	98.6	-	-	-
wav2vec2.0 large (Ours)							
- Single task Adapter	22.3	65.6	8.5	98.4	94.7, 12.9	0.6	99.4
- MTL: Stacked	24.2	**68.2**	7.7	98.7	94.4, 13.5	0.6	99.5
- MTL: Fusion	**22.1**	65.4	**7.3**	**99.1**	**95.4, 11.8**	0.6	99.3

Table 2. Performance comparison between various MTL implementations and our three different adapter-based architectures. *uses weighted accuracy

	# of tasks	
	6 tasks	9 tasks
SUPERB [7]	126.6M	252.8M
Ours	**113.1M**	**135.6M**
Ratio	89.3%	53.6%

Table 3. Comparison between the total # of additional trainable parameters required to accommodate 6 tasks depicted in Table 1 and 9 tasks which includes the additional ASR tasks in Table 2.

along with ER, we format the ground truth as <transcript> <task separator> <emotion label>.

For training the encoder-decoder model, we use a combination of losses depending on the adapter architecture and the task. The overall objective $\mathbf{L}$ can be written as,

$$L_{nll} = \sum_{task=1}^{N} \lambda_{task} \cdot L_{task} \quad (1)$$

$$\mathbf{L} = (1 - \lambda_{ctc}) \cdot L_{nll} + \lambda_{ctc} \cdot L_{ctc} + \mathbb{1}_{ce} \cdot L_{ce} \quad (2)$$

where L_{nll} denotes the Negative Log-Likelihood loss at the decoder end. The output tokens during multi-task training comprise tokens from N tasks which are weighted using the hyperparameter λ_{task}. L_{ctc} denotes CTC loss applied at the encoder end, similar to hybrid CTC/attention objective [21]. Hyperparameter λ_{ctc} is used to weigh between the NLL and CTC loss. Finally, L_{ce} denotes Cross Entropy Loss applied at the encoder end for classification tasks, and hyperparameter $\mathbb{1}_{ce}$ is 1 when it's a classification task.

4. EXPERIMENTAL SETUP

We train adapters to perform five different SLP tasks (corresponding datasets are denoted in the brackets), 1) ASR (LibriSpeech [20]), 2) PR (LibriSpeech), 3) ER (IEMOCAP [27]), 4) IC (Fluent Speech Commands [28]), and 5) SF (SNIPS [29]). We chose datasets used by the SUPERB benchmark[2] [7] for the corresponding tasks. In addition, we also train ASR adapters specifically for each domain (IEMOCAP, SNIPS and FSC) which helps in MTL (e.g, ASR + ER) and IC adapter for SNIPS which helps when performed with SF.

For evaluation, we follow the same setting as SUPERB. The adapter dimension was set to 128. For λ settings, in single-label classification tasks such as ER, λ_{ctc} is set to 0. For the rest, λ_{ctc} is set to 0.3 and in experiments where CTC loss was used, we combined the attention-based and CTC

[2] https://github.com/s3prl/s3prl

scores for joint decoding, assigning a weight of 0.4 to the CTC scores (following the SpeechBrain recipe). For MTL, in the stacked adapter setting [8] only the additional adapter is trained, while the rest of the model, including the bottom adapter(s) remains frozen. Here, λ_{task} assigns a higher weight to the tokens corresponding to the new task. In our experiments, this value was set to 0.9 for the new task and 0.1 for the tasks of the already-trained bottom adapters. In fusion, the adapters are already trained with respective tasks, so we experimented with two settings: first, λ_{task} is set to 1, and second, we set it to equal weights for all tasks and chose the best. We modified SpeechBrain [3] recipes for our implementation.

5. RESULTS AND DISCUSSION

Table 1 presents our results, showing that our approach achieves better performance compared to the wav2vec2 SUPERB [7] benchmark (+5.4) and actually also with the WavLM model (+1.4) also from SUPERB. This performance improvement can be attributed to our design choice of utilizing adapters that allows combining different tasks for improved multi-task learning. For example, for SF performance on SNIPS, the adapters where ASR, SF and IC are learned simultaneously allows an improvement of 8.3 F1 and 15.5 CER (see Table 2). Detailed results regarding the performance of different adapter combinations are discussed in the Ablation Study. In contrast to having a frozen encoder and task-specific decoders, we incorporate task-specific adapters on a single encoder-decoder model to perform multiple SLP tasks which leads to both efficient utilization of encoder representations, and memory efficiency (see Table 3).

5.1. Ablation Study: Comparison between different types of adapter-based task modules

Research has shown that certain tasks, like ASR and ER [12, 13], can benefit from simultaneous learning, enhancing each other's performance. As adapters naturally enable MTL [2], in addition to single-adapter task modules, we investigate two adapter-based MTL approaches: Stacking and Fusion. We hypothesize that performing MTL with adapters produces less increase in computational overhead compared to the performance improvement.[4]

Table 2 compares the performance amongst three different adapter settings and also with existing works that perform MTL. In the IEMOCAP dataset, the Adapter Stacking setting achieves the highest performance in Emotion Recognition. On the SNIPS dataset, the Adapter Fusion setting performs the best in SF and IC. Our performance is comparable to studies that use gold-text directly, such as [30] with an SF-F1 score of 95.9 and IC-Acc of 98.8% on SNIPS. For FSC, there is minimal performance variation in the literature, since models already achieve above 99% accuracy. The WER is also comparable with existing works — Ours: 0.6, and [31]: 0.5. This performance improvement of adapter-based MTL architectures aligns with previous research indicating that MTL enhances task performance [30]. Furthermore, fine-tuning our ASR adapters for each dataset performs better than approaches that use generic ASR models, as previously demonstrated by [12].

In addition to the improvements in performance, our unified model shows multi-task capabilities in a parameter-efficient and scalable manner. Table 3 illustrates the comparison between our approach and the SUPERB benchmark in terms of the number of trainable parameters needed for accommodating six tasks (as in Table 1) and nine tasks (including the additional ASR tasks from Table 2). Notably, our approach requires fewer parameters, and more importantly, even as the number of tasks increases, the increase in the parameter count remains significantly lower with the ratio dropping to 53.6%.

6. CONCLUSION

Our work shows that adapter-based task modules effectively enable a unified encoder-decoder model for handling multiple speech-processing tasks. Our experiments show that we are able to achieve performance improvements compared to the SUPERB benchmark, while being more efficient in terms of parameters by eliminating the need for dedicated task-specific decoders. This work highlights the potential to develop simple and scalable model architectures that are capable of performing multiple SLP tasks within a unified model. In the future, our goals include evaluating our approach for different choices of SSL models such as HuBERT and WavLM and exploring different adapter architectures. Additionally, we also aim to broaden the scope of our approach to add the remaining tasks in the SUPERB benchmark such as Speaker Identification, Speaker Diarization, and other speech-processing tasks/datasets.

7. REFERENCES

[1] Alexei Baevski et al., "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, 2020.

[2] Neil Houlsby et al., "Parameter-efficient transfer learning for nlp," in *Int. Conf. on Machine Learning*. PMLR, 2019.

[3] Hang Le et al., "Lightweight adapter tuning for multilingual speech translation," in *Proc. of the 59th Annual Meeting of the ACL and the 11th Int. Joint Conf. on Natural Language Processing*, 2021.

[3] https://github.com/speechbrain/speechbrain

[4] Adapter stacking: no change in the number of parameters and adapter Fusion introduces an additional 57M parameters.

[4] Edward Gow-Smith et al., "Naver labs europe's multilingual speech translation systems for the iwslt 2023 low-resource track," *arXiv preprint arXiv:2306.07763*, 2023.

[5] Anastasopoulos Antonios et al., "Findings of the iwslt 2022 evaluation campaign.," in *Proc. of the 19th Int. Conf. on Spoken Language Translation*. ACL, 2022.

[6] Bethan Thomas et al., "Efficient adapter transfer of self-supervised speech models for automatic speech recognition," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

[7] Shu-wen Yang et al., "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[8] Jonas Pfeiffer et al., "Adapterhub: A framework for adapting transformers," *arXiv preprint arXiv:2007.07779*, 2020.

[9] Yuting Zhao and Ioan Calapodescu, "Multimodal robustness for neural machine translation," in *Proc. of the 2022 Conf. on Empirical Methods in Natural Language Processing*, 2022.

[10] Bryan McCann et al., "The natural language decathlon: Multitask learning as question answering," *arXiv preprint arXiv:1806.08730*, 2018.

[11] Xingyu Cai et al., "Speech emotion recognition with multi-task learning.," in *Interspeech*, 2021.

[12] Yuanchao Li et al., "Fusing asr outputs in joint training for speech emotion recognition," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

[13] Han Feng et al., "End-to-end speech emotion recognition combined with acoustic-to-word asr model.," in *INTERSPEECH*, 2020.

[14] Changliang Li et al., "A joint multi-task learning framework for spoken language understanding," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[15] Yun Tang et al., "A general multi-task learning framework to leverage text data for speech to text tasks," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

[16] Alec Radford et al., "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[17] Junyi Ao et al., "Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing," in *Proc. of the 60th Annual Meeting of the ACL*, 2022.

[18] Yi-Chen Chen et al., "Speechnet: A universal modularized model for speech processing tasks," *arXiv preprint arXiv:2105.03070*, 2021.

[19] Salah Zaiem et al., "Speech self-supervised representation benchmarking: Are we doing it right?," *arXiv preprint arXiv:2306.00452*, 2023.

[20] Vassil Panayotov et al., "Librispeech: an asr corpus based on public domain audio books," in *Int. Conf. on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015.

[21] Shinji Watanabe et al., "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, 2017.

[22] Siddhant Arora et al., "Espnet-slu: Advancing spoken language understanding through espnet," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

[23] Quentin Meeus et al., "Multitask learning for low resource spoken language understanding," *INTERSPEECH*, 2022.

[24] Cheng-I Lai et al., "Towards semi-supervised semantics understanding from speech," *arXiv preprint arXiv:2011.06195*, 2020.

[25] Ashish Vaswani et al., "Attention is all you need," *Advances in neural information processing systems*, 2017.

[26] Henry Weld et al., "A survey of joint intent detection and slot filling models in natural language understanding," *ACM Computing Surveys*, 2022.

[27] Carlos Busso et al., "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, 2008.

[28] Loren Lugosch et al., "Speech model pre-training for end-to-end spoken language understanding," 2019.

[29] Alice Coucke et al., "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *CoRR*, vol. abs/1805.10190, 2018.

[30] Libo Qin et al., "A co-interactive transformer for joint slot filling and intent detection," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

[31] Xuandi Fu et al., "Multi-task rnn-t with semantic decoder for streamable spoken language understanding," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.