Efficient infusion of self-supervised representations in Automatic Speech Recognition

Darshan Prabhu[*]
Sony Research India
darshan.prabhu@sony.com

Sai Ganesh Mirishkar
Sony Research India
saiganesh1.mirishkar@sony.com

Pankaj Wasnik
Sony Research India
pankaj.wasnik@sony.com

Abstract

Self-supervised learned (SSL) models such as Wav2vec and HuBERT yield state-of-the-art results on speech-related tasks. Given the effectiveness of such models, it is advantageous to use them in conventional ASR systems. While some approaches suggest incorporating these models as a trainable encoder or a learnable frontend, training such systems is extremely slow and requires a lot of computation cycles. In this work, we propose two simple approaches that use (1) framewise addition and (2) cross-attention mechanisms to efficiently incorporate the representations from the SSL model(s) into the ASR architecture, resulting in models that are comparable in size with standard encoder-decoder conformer systems while also avoiding the usage of SSL models during training. Our approach results in faster training and yields significant performance gains on the Librispeech and Tedlium datasets compared to baselines. We further provide detailed analysis and ablation studies that demonstrate the effectiveness of our approach.

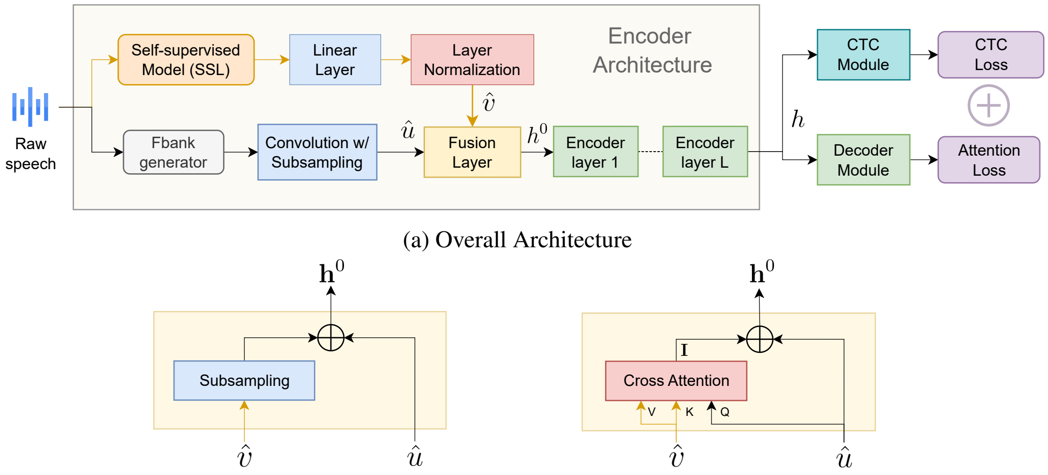
1 Introduction

Since speech is a complex signal, audio-related tasks like Automatic Speech Recognition(ASR) rely heavily on having robust speech representations as input. While standard ASR systems use conventional signal processing techniques [13] for generating these representations, another set of models, called representation models, are tasked with learning to generate such representations using large unlabeled data. These models are trained using the masked language modeling [3] objective, where the model attempts to reconstruct parts of the text/audio that are masked. As a result, they can understand raw text/speech well and generate high-quality text/speech representations. BERT [3], Wav2Vec [1], and HuBERT [6] are some popular models that fall under this category. They are also called Self-Supervised Learned(SSL) models, as they do not need labeled data during training. These models are popular for downstream tasks as they perform exceptionally well even with small amounts of data. However, since these models are huge, training an ASR system retrofitted with such models as a learnable frontend [16] or an Encoder with limited computational resources becomes challenging, if not impossible. To address this issue, several works propose freezing the representation model and using its output as auxiliary information to a custom encoder. This enables us to entirely forgo the usage of SSL models during training, as the extraction of these representations can be offloaded to preprocessing [2]. Early work in this regard is found in Natural Language Processing(NLP), where BERT representations are integrated into the Neural Machine Translation system [18]. Recent works perform this integration for multimodal inputs as well [17, 7]. In speech, fine-tuned Wav2vec2 embeddings have been used as auxiliary data alongside the domain adversarial setup [9, 10] for the accented ASR setting.

[*] Work done during an internship at Sony Research India. Corresponding to: Sai Ganesh Mirishkar

[2] This is a one-off step performed once at the beginning of training.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).



(a) Overall Architecture

(b) Subsampled Framewise Addition (SFA)

(c) Cross Attention (CA)

Figure 1: Overview of our proposed Architecture that integrates the representations from the self-supervised model into the ASR encoder using two approaches: Subsampled Framewise Addition(**SFA**) and Cross Attention(**CA**). $\hat{v}$ and $\hat{u}$ are the self-supervised and fbank representations respectively.

In this work, we propose two approaches that incorporate representations from pre-trained SSL models into an end-to-end ASR architecture. We also explore the possibility of introducing multiple such representations into the architecture and perform a comprehensive analysis of our approach in terms of convergence, efficiency, and understanding.

2 Proposed Methodology

Figure 1 shows our proposed modifications to the existing joint CTC-Attention [8] framework that allows auxiliary audio representations(extracted from pre-trained SSL models) to be easily introduced within the end-to-end ASR architecture without substantially increasing the model size [3]. Figure 1a shows the overall architecture that comprises three main modules: Encoder(Enc), Decoder(Dec-Att) and Connectionist Temporal Classification(Dec-Ctc) [4]. Although all our changes are made to the encoder, we first briefly explain the overall architecture, followed by our proposed modifications to the encoder.

Let $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$ be the raw representation of the input audio. This representation x is passed through (1) Fbank generator that generates T length fbank representation $\mathbf{u} = \{u_1, u_2, \ldots, u_T\}$ where $u_i \in \mathbb{R}^d$ and (2) Self supervised pre-trained model that generates T' length SSL representation $\mathbf{v} = \{v_1, v_2, \ldots, v_{T'}\}$ where $v_i \in \mathbb{R}^{d'}$. The encoder(Enc) jointly reasons on both $\mathbf{u}$ and $\mathbf{v}$ to generate contextualized audio representation $\mathbf{h} = \text{Enc}(\mathbf{u}, \mathbf{v}) = \{h_1, h_2, \ldots h_T\}$. This contextualized representation $\mathbf{h}$ is then consumed by Dec-Att and Dec-Ctc module which aim at predicting the output token sequence $\mathbf{y} = \{y_1, y_2, \ldots, y_i, \ldots y_M\}$ using CTC and Attention loss.

The encoder module begins with a Convolution_Subsampling block that applies convolution and two-factor subsampling [4] on the fbank representation $\mathbf{u}$ resulting in a $T/2$ length sequence $\hat{\mathbf{u}}$. Parallely, we employ a linear layer and layer normalization on the SSL representation $\mathbf{v}$ to obtain a T' length sequence $\hat{\mathbf{v}}$. While layer normalization helps with better generalization, the linear layer reduces the dimension from d' to d. These sequences are then fed to the Fusion_Layer that acts on both these inputs, performing a deterministic fusion of $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ to generate $\mathbf{h}^0$(In section 2.1, we discuss in detail the different choices available for the fusion layer along with their merits and demerits.). The output of the fusion layer is then passed through a stack of L identical encoder layers. Each encoder layer

[3] As the usage of SSL model during inference is unavoidable, no computational benefit would be observed using our approach for inference.

[4] Subsampling is a crucial step that reduces the computation complexity of the encoder while performing at par with the original encoder [2].

feeds on the output from the previous layer $\mathbf{h}^{i-1}$ and generates contextualized representation $\mathbf{h}^{i}$ by performing the standard operation of a conformer [5] encoder layer.

2.1 Fusion Layer

In this section, we discuss two simple approaches to generate the representation $\mathbf{h}^0$ using sequences $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$. To reiterate, $\mathbf{u}$ is a T length sequence that is passed through two-factor subsampling to generate the representation $\hat{\mathbf{u}}$ which is a sequence of length $T/2$. $\hat{\mathbf{v}}$ which is obtained by passing raw audio through the SSL model followed by layer normalization is a T' length sequence which in the case of Wav2Vec and HuBERT happens to be equal to T. Finally, the output of this fusion layer is $\mathbf{h}^0$ which is also a $T/2$ length sequence.

Subsampled Framewise Addition (SFA): Inspired by the work of Jialu et al. [9], we first propose a simple parameterless approach as shown in Figure 1b that relies on the observation that performing subsampling on $\hat{\mathbf{v}}$ by a factor of 2 conveniently leads to both $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ to be of equal length. We can then perform framewise addition of both these sequences to generate $\mathbf{h}^0$. Mathematically, this generation can be written as: $\mathbf{h}^0_i = \hat{\mathbf{u}}_i \oplus \hat{\mathbf{v}}_{min(T,2\times i)} (\forall \;\; 1 \leq i \leq T/2)$ where $\mathbf{h}^0_i$ is the i^{th} entry in $\mathbf{h}^0$ and $\oplus$ is the elementwise addition of two d-dimensional vectors. As this is a parameterless operation, it does not add to the model size but there are two main drawbacks with this approach. First, we have a predetermined decision on which frame of the corresponding sequences are to be added. It would be beneficial to let the model determine this mapping. Second, the approach heavily relies on the lengths of the two sequences to have some linear relation, and for arbitrary lengths, it becomes harder to determine this mapping. It becomes exponentially harder if the lengths of both sequences differ by a large margin.

Cross Attention (CA): To address both the concerns, we introduce a cross-attention layer as shown in Figure 1c that uses the concept of attention to determine how each frame of $\hat{\mathbf{u}}$ wants to attend to the frames of $\hat{\mathbf{v}}$. Mathematically, this generation of $\mathbf{h}^0$ can be written as:

$$\mathbf{h}^0_i = \hat{\mathbf{u}}_i \oplus \mathbf{I}_i \quad \forall \quad 1 \leq i \leq T/2 \;\; \text{where} \;\; \mathbf{I} = \mathrm{MultiHeadAttention}(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{v}})$$

where $\mathrm{MultiHeadAttention}(Q, K, V)$ refers to the standard multi-headed attention proposed by Vaswani et al. [14] with Q, K and V denoting query, key and value respectively. This approach is free from any predetermined mapping of the frames of the corresponding sequences, as attention helps the model learn such mappings.

3 Experimental Setup

We run all our experiments on NVIDIA A100 GPUs using the ESPnet toolkit [15]. As is common practice, we add 3-way speed perturbation to both datasets before training. Unless specified otherwise, in all of our experiments, we train a conformer model with 12 encoder and 6 decoder layers. We use 256-dimensional tensors and four heads for attention computation. All the models are trained for 50 epochs with a batch size of 32, a learning rate of 1.0, and 4 gradient accumulation steps. The representations from SSL models are dumped beforehand to speed up the training. Throughout all of our experiments, we use three self-supervised models: (1) Wav2vec 2.0 BASE, (2) HuBERT BASE, and (3) HuBERT LARGE. All these models are only pre-trained with the SSL objective without any ASR fine-tuning. We report numbers on Librispeech-100 [11] and Tedlium2 [12] datasets that are publicly available.

4 Results

Table 1 compares word error rates (WER) between our proposed architectures (T1 to T5) and baselines (B1 and B2) on the Librispeech 100h [11] dataset. We report two baselines, out of which the Conformer baseline outperforms the Transformer baseline by a large margin. We show five variants of our architecture, labeled T1 through T5, each of which uses the conformer architecture but differs in either the choice of SSL model or the fusion method. We see that among Wav2vec and HuBERT, HuBERT representations consistently perform better. Furthermore, we find no added

Table 1: Comparison of Performance(WER) of our system with baselines on Librispeech-100 dataset [11]. Our experiments are labeled in the format X+Y+Z, indicating that the architecture X was modified to include representation from the Y SSL model by employing the Z approach.

Method	Dev Clean	Dev Other	Test Clean	Test Other
B1: Transformer (Trans.)	10.1	25.8	10.4	26.4
B2: Conformer (Conf.)	7.9	21.4	8.4	22.0
T1: Conf. + w2v-BASE + SFA	5.9	16.2	6.4	16.4
T2: Conf. + w2v-BASE + CA	5.9	16.0	6.3	16.3
T3: Conf. + HuBERT-BASE + SFA	5.2	13.5	**5.4**	13.5
T4: Conf. + HuBERT-BASE + CA	**5.1**	**13.0**	**5.4**	**13.3**
T5: Conf. + (w2v-BASE, HuBERT-BASE) + CA	5.7	14.1	6.1	14.5

advantage in using both Wav2Vec and HuBERT representations. Irrespective of the SSL model chosen, cross-attention(CA) shows significant improvement over the sub-sampled framewise addition(SFA) approach.

5 Analysis

Faster Model Convergence: As shown in Figure 2, our approach outperforms the baseline best results within only ten epochs of training and continues to improve over the next few iterations. Moreover, the convergence is much faster in the early iterations, resulting in a reasonably good model within the first 5-10 epochs of training.

Efficacy of our approach: While directly incorporating HuBERT or Wav2vec as a frontend/Encoder is bound to perform better, our approach results in a model that is both faster in training and smaller in size at the cost of a slight degradation in performance. Furthermore, performing SSL feature extraction in the preprocessing stage adds a very minimal overhead [5] (approx 4 hrs for Librispeech-100 and 10 hrs for Tedlium2) compared to using SSL models throughout training.

Number of encoder layers: Table 2 compares the conformer baseline with four variations of our best system(Conf.+HuBERT-BASE+CA) which differ only in the number of encoder layers used. Even after reducing the number of encoder layers by 80%, we find that the model still performs much better despite having only half as many parameters and training time as the baseline model.

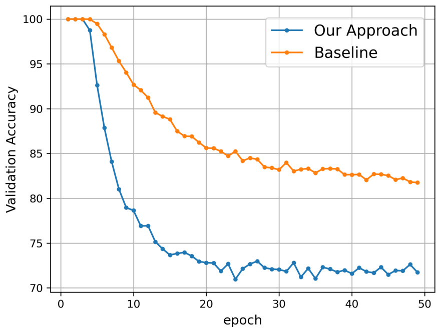


Figure 2: Comparison of epochwise model performance(WER) between baseline and our best setting(Conf.+HuBERT-BASE+CA) on the validation split of Librispeech-100 dataset.

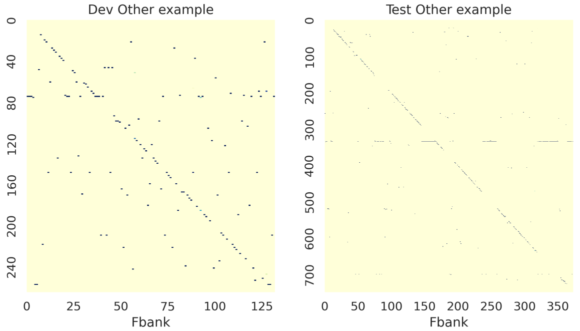


Figure 3: Visualization of attention scores for two samples from the Librispeech-100 dataset, one from Dev-Other split and Test-Other split respectively. We use the model from our best setting to obtain these scores.

[5] This process can be easily parallelized. In our experiments, we use sixty parallel jobs to perform feature extraction.

Table 2: Comparison of parameter count, training time and Performance(WER) of our architecture with baseline on Librispeech 100h dataset [11]. Conf.(E=x) represents conformer architecture with x Encoder layers.

Method	# of params	Train time	Dev		Test	
			Clean	Other	Clean	Other
Conf.(E=12)	30.6M	24h	7.9	21.4	8.4	22.0
Conf.(E=12) + HuBERT-Base + CA	31.0M	26h	5.1	13.0	5.4	13.3
Conf.(E=8) + HuBERT-Base + CA	24.7M	22h	**5.0**	12.9	**5.3**	13.0
Conf.(E=4) + HuBERT-Base + CA	18.4M	17h	5.4	**12.8**	5.7	**12.7**
Conf.(E=2) + HuBERT-Base + CA	15.2M	15h	5.5	**12.7**	5.5	12.8

Table 3: Comparison of Performance(WER) of our approach with baseline on the Dev and Test splits of Tedlium dataset [12]. We follow the same heuristics from Table 1 to label the experiments.

Method	Dev	Test
Conformer (Conf.)	10.5	8.6
Conf. + w2v-Base + CA	9.6	9.3
Conf. + HuBERT-Base + CA	9.4	8.9
Conf. + HuBERT-Large + CA	**7.6**	**6.8**

Visualization of Attention: Figure 3 depicts the attention distribution generated by our Cross-Attention(CA) fusion layer for few examples. As can be seen, the layer has learned to predict the alignment between the two representations, indicating that approaches like Convolution or windowed attention that focuses on local attention can also be employed to achieve reasonable improvement.

Performance on unseen dataset: Table 3 compares the performance between baselines and different variations of our proposed architecture on the Tedlium dataset. We deliberately chose the SSL model that is not exposed to the Tedlium dataset to test the effectiveness of the generated representations on the unseen datasets. Because the Large model is exposed to a much more diverse dataset, we find it to be far more effective than the Base model. Furthermore, we find normalization to be a critical step for out-of-domain datasets.

6 Conclusion

In this work, we propose an end-to-end ASR architecture that integrates self-supervised model representations into the speech encoder. We accomplish this with a fusion layer, which can be as simple as framewise addition to a more complex cross-attention mechanism. To evaluate the efficacy of our approach, we conduct experiments on the Librispeech and Tedlium datasets, demonstrating significant reductions in word error rate (WER) compared to standard baseline models. Furthermore, we perform a thorough analysis of our approach, with a particular focus on the convergence rate, the impact of the number of parameters, and the information captured by the attention mechanism. Through these detailed analyses, we highlight the speed, efficiency, and scalability that our approach achieves in comparison to other baseline methods.

References

[1] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

[2] Maxime Burchi and Valentin Vielzeuf. Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition, 2021.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[4] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143891. URL `https://doi.org/10.1145/1143844.1143891`.

[5] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition, 2020.

[6] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units, 2021.

[7] Donghwa Kim and Pilsung Kang. Cross-modal distillation with audio–text fusion for fine-grained emotion classification using bert and wav2vec 2.0. *Neurocomputing*, 506:168–183, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2022.07.035. URL `https://www.sciencedirect.com/science/article/pii/S0925231222008931`.

[8] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning, 2017.

[9] Jialu Li, Vimal Manohar, Pooja Chitkara, Andros Tjandra, Michael Picheny, Frank Zhang, Xiaohui Zhang, and Yatharth Saraf. Accent-robust automatic speech recognition using supervised and unsupervised wav2vec embeddings, 2021.

[10] Ganesh S Mirishkar, Vishnu Vidyadhara Raju V, Meher Dinesh Naroju, Sudhamay Maity, Prakash Yalla, and Anil Kumar Vuppala. Iiith-cstd corpus: Crowdsourced strategies for the collection of a large-scale telugu speech corpus. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(7):1–26, 2023.

[11] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5206–5210. IEEE, 2015.

[12] Anthony Rousseau, Paul Deléglise, and Yannick Estève. Ted-lium: an automatic speech recognition dedicated corpus. In *Conference on Language Resources and Evaluation (LREC)*, pages 125–129, 2012.

[13] Kathleen M. Stafford, Christopher G. Fox, and David S. Clark. Long-range acoustic detection and localization of blue whale calls in the northeast Pacific Ocean. *The Journal of the Acoustical Society of America*, 104(6):3616–3625, 12 1998. ISSN 0001-4966. doi: 10.1121/1.423944. URL `https://doi.org/10.1121/1.423944`.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[15] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pages 2207–2211, 2018. doi: 10.21437/Interspeech.2018-1456. URL `http://dx.doi.org/10.21437/Interspeech.2018-1456`.

[16] Neil Zeghidour, Olivier Teboul, Félix de Chaumont Quitry, and Marco Tagliasacchi. Leaf: A learnable frontend for audio classification, 2021.

[17] Zihan Zhao, Yanfeng Wang, and Yu Wang. Multi-level fusion of wav2vec 2.0 and bert for multimodal emotion recognition, 2022.

[18] Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Incorporating bert into neural machine translation, 2020.