Leveraging Uni-Modal Self-Supervised Learning for Multimodal Audio-visual Speech Recognition

Anonymous ACL submission

Abstract

Training Transformer-based models demands a large amount of data, while obtaining parallel aligned and labelled data in multimodality is rather cost-demanding, especially for audio-visual speech recognition (AVSR). Thus it makes a lot of sense to make use of unlabelled uni-modal data. On the other side, although the effectiveness of large-scale self-supervised learning is well established in both audio and visual modalities, how to integrate those pre-trained models into a multimodal scenario remains underexplored. In this work, we successfully leverage uni-modal self-supervised learning to promote the multimodal AVSR. In particular, we first train audio and visual encoders on a large-scale uni-modal dataset, then we integrate components of both encoders into a larger multimodal framework which learns to recognize paired audio-visual data into characters through a combination of CTC and seq2seq decoding. We show that both components inherited from uni-modal self-supervised learning cooperate well, resulting in that the multimodal framework yields competitive results through fine-tuning. Our model is experimentally validated on both word-level and sentence-level AVSR tasks. Especially, even without an external language model, our proposed model raises the state-of-the-art performances on the widely accepted Lip Reading Sentences 2 (LRS2) dataset by a large margin, with a relative improvement of 30%.

1 Introduction

Audio-Visual Speech Recognition (AVSR) is a speech recognition task that leverages both an audio input of human voice and an aligned visual input of lip motions. It has been one of the successful application fields that involve multiple modalities in recent years. Due to the limited amount of labeled, multi-modal parallel data and the difficulty of recognition from the visual inputs (i.e., lip reading), it is a challenging task to tackle.

Existing AVSR models tend to use extra data to increase the performance of the system, in a form of inserting an extra supervised learning stage in the training process. For example, many existing methods rely on an extra sequence level classification to bootstrap its learning on visual features. Petridis et al. (2018); Zhang et al. (2019) train their visual front-end on LRW (Chung and Zisserman, 2016) before learning on the AVSR task. Afouras et al. (2018a,b) chunks the MV-LRS data (Chung and Zisserman, 2017) into pieces of words and pre-train the model through classification. VoxCeleb (Chung et al., 2018) are also used in Afouras et al. (2020) for the same purpose. Learning an effective visual front-end could still be notoriously hard, even with these extra supervised learning tasks. Sometimes curriculum learning is required to adapt the learned visual front-end into AVSR task (Afouras et al., 2018a). End-to-end learning of large-scale AVSR data hasn't been successful until recently (Ma et al., 2021).

Although self-supervised learning could enable leveraging unlabelled or even non-parallel data, it hasn't been adequately explored on this task. Shukla et al. (2020) is among the few attempts in this facet, in which it predicts lip motions from audio inputs. Their proposed learning schemes yield strong emotion recognition results but are relatively weak in speech recognition. Moreover, since in AVSR it is the lip shape and motions between frames rather than the objects in a single image that matters for recognizing speech contents, if pre-trained visual models tailored for tasks targeting at single frame images could work for AVSR remains unknown. In another scenario, self-supervised learning in uni-modality has been well established as a paradigm to learn general representations from unlabelled examples, such as in natural language processing (Brown et al., 2020; Devlin et al., 2018), speech recognition (Baevski et al., 2020), and computer vision (He et al., 2019;

Chen et al., 2020a; Grill et al., 2020).

In this work, we rely on a simple but effective approach, which is to utilize unlabelled uni-modal data by using pre-trained models that are trained in single-modality through self-supervised learning. Specifically, we use Baevski et al. (2020) pre-trained on the large LibriLight (Kahn et al., 2020) dataset as our audio front-end. For visual front-end, we found that it is not as straight-forward for it to leverage pre-trained models, as we have to substitute the first ResNet block in MoCo v2 (Chen et al., 2020b) by 3-D convolution layer and fine-tune it through LRW. In total, our approach doesn't require a curriculum learning stage, and the overall training time has been decreased.

Experimental results show that our new front-ends significantly outperform previous ones by a big margin in both audio-only and visual-only settings, and a new state-of-the-art has been achieved in the final AVSR setting. To our best knowledge, this is the first work that successfully applies uni-modal pre-trained models in the multi-modal setting of AVSR. We also ensure this research is reproducible by publishing our codes at `anonymized_url`.

2 Related Work

2.1 Audio-Visual Speech Recognition

The earliest work on AVSR could be dated back to around two decades ago, when Dupont and Luettin (2000) showed hand-crafted visual feature improves HMM-based ASR systems. The first modern AVSR system is proposed in Afouras et al. (2018a) where deep neural networks are used. The field has been rapidly developing since then. Most of the works are devoted into the architectural improvements, for example, Zhang et al. (2019) proposed temporal focal block and spatio-temporal fusion, and Lee et al. (2020a) explored to use cross-modality attentions with Transformer.

The other line of research focuses on a more diversified learning scheme to improve AVSR performance. Li et al. (2019) uses a cross-modal student-teacher training scheme. Paraskevopoulos et al. (2020) proposes a multi-task learning scheme by making the model to predict on both character and subword level. Self-supervised learning has also been explored in Shukla et al. (2020), where the cross-modality setting is utilized by predicting frames of videos from audio inputs.

The end-to-end learning of AVSR systems are first seen in Tao and Busso (2020), albeit in a much simpler dataset than LRS2. More recent work (Ma et al., 2021) has made end-to-end learning on LRS2 possible by using a Conformer acoustic model and a hybrid CTC/attention decoder.

2.2 Self-Supervised Learning

Self-supervised learning has been chased in recent years since its ability to learn general representations of data through simple tasks that don't require labeling. Contrastive learning (Hadsell et al., 2006) has become the most impactful learning scheme in this field. In natural language processing, uni-or bi-directional language modelling (Brown et al., 2020; Devlin et al., 2018) have been used to significantly increase performances on various tasks. In audio speech processing, contrastive predictive coding (Baevski et al., 2020) has been proven to be powerful in speech recognition. In the visual domain, Earlier works create self-supervised tasks through image processing based methods, such as distortion (Gidaris et al., 2018),colorization (Zhang et al., 2016) and context prediction (Doersch et al., 2015). More recently, contrastive learning emerged as a paradigm of self-supervised learning, which results in a group of more expressive general visual representations, such as MoCo (He et al., 2019; Chen et al., 2020b), SimCLR (Chen et al., 2020a), BYOL (Grill et al., 2020), etc.

3 Architecture

The overall architecture of our model is shown in Fig. 1. The audio-visual model is comprised of four components, the front-ends and back-ends for both modalities, the fusion module, and the decoders.

3.1 Front-ends

Visual Front-end: Visual front-end serves as a component to capture the lip motion and reflect the lip position differences in its output representations. A naive way to apply pre-trained models in the visual front-end is to directly feed the RGB channels of each frame as input. However, since frames within a same clip in AVSR are largely similar in their contents while most pre-trained models in vision target at learning general representations reflecting the content of the whole image, this approach will result in similar outputs for all the frames, collapsing the informative lip position differences between frames.

To overcome the aforementioned problem while

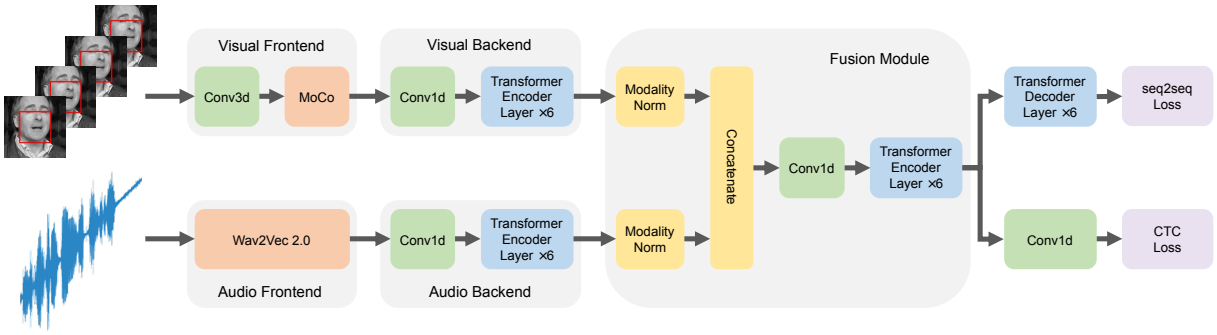


Figure 1: Overall architecture of our AVSR model.

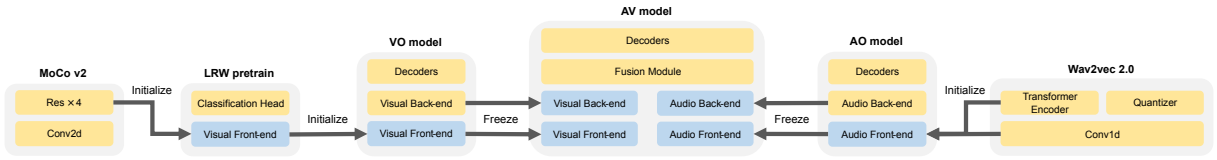


Figure 2: Training pipeline of the model. Yellow blocks represent new parameters that are randomly initialized, while Blue blocks represent parameters that are inherited from last training stage.

still being able to utilize the pre-trained model, we truncate the first convolutional layer in MoCo v2 (Chen et al., 2020b), which is pre-trained on ImageNet (Deng et al., 2009), and replace it by a layer of 3-D convolution. The outputs of 3-D convolution layer are intentionally made identical to the input of the first ResBlock in MoCo v2 (see Table 1), thus providing a compatible interface to transfer higher layers of MoCo v2 into this task. On the other hand, we also adopt the common practice to convert the RGB input image to gray-scale before feeding it into the model, as it prevents the model from learning chromatic aberration information.

Audio Front-end: The audio front-end is rather straight-forward. We use wav2vec 2.0 (Schneider et al., 2019) pre-trained on Libri-Light (Kahn et al., 2020), like it is normally used for ASR tasks, both the 1-D convolution layers and the stacked Transformer encoder layers are transferred into our audio front-end. The audio front-end takes as input raw audio wave of 16kHz, and produces one vector representation every 20ms. The audio feature dimensions are shown in Table 2.

3.2 Back-ends

Since the visual frames are in 25 FPS and the wav2vec 2.0 outputs are around 49 Hz[1], one should note that there is 2x difference in the frequency of frame-wise visual and audio representations at the output of their front-ends. In the back-end, we use 1-D convolution layers on the time dimension combined with Transformer encoder layers to provide single modality temporal modeling, as well as adjusting the features to have the same frequency.

Visual Back-end: The incoming MoCo v2 output to the visual back-end has a feature dimension of 2048, at a frequency of 25 vectors per second. In the visual backend, we keep this frequency while reducing the feature size to 512. See Table 1. For positional encodings of the Transformer, we use fixed positional encoding in the form of sinusoidal functions.

Stage	Modules	Image sequence $(T_f \times 112^2 \times 1)$
Front-end	3-D convolution	$(T_f \times 28^2 \times 64)$
	MoCo v2	$(T_f \times 2048)$
Back-end	1-D convolution	$(T_f \times 512)$
	Transformer Encoder	$(T_f \times 512)$

Table 1: The feature dimension of visual stream. The dimensions of features are denoted by {temporal size × spatial size^2 × channels}. T_f denotes the number of visual frames.

Audio Back-end: In the audio back-end, the incoming wav2vec 2.0 outputs have a feature size of

[1]The odds are due to the larger receptive fields of wav2vec 2.0 1-D convolution layers, which we circumvent by properly prefixing and suffixing the audio sequence and truncate the trailing audio vector. Thus a perfect 1:2 ratio of visual frames and audio front-end outputs are ensured.

1024, at a frequency of 50 vectors per second. We downscale the frequency by setting the stride of 1-D convolution layer to 2. The Transformer encoder layers have the identical size to that of the visual back-end, while using a separate set of parameters. Table 2 shows a clearer picture of audio front- and back-end dimensions.

Stage	Modules	Audio waveform ($T_s \times 1$)
Front-end	wav2vec 2.0	($T_f \times 1024$)
Back-end	1-D convolution	($\frac{T_f}{2} \times 512$)
	Transformer Encoder	($\frac{T_f}{2} \times 512$)

Table 2: The feature dimension of audio stream. The dimensions of features are denoted by {temporal size × channels}. T_s and T_f denote the number of sampled audio input and audio frames, respectively.

3.3 Fusion Module

Features from both the audio and visual modalities are fused together in this section, forming vector representation of 1024 dimensions at a relatively low rate of 25 Hz. We use LayerNorm (Ba et al., 2016) separately on each of the modalities before concatenating them on the feature dimension. The LayerNorm is required since it avoids one modality overtaking the whole representation with larger variance. Similar 1-D convolution layers and a subsequent Transformer encoder block of 6 layers take the fused representations as input, and encode them for the two decoders.

3.4 Decoder

Following the setting of Petridis et al. (2018), there are two decoders trained simultaneously based on the same encoder output in the fusion module.

The first is a Transformer seq2seq decoder, a canonical Transformer decoder with 6 layers is used, and we perform teacher forcing at character level by using ground truth characters as input during training.

The second one is arguably a decoder since it yields character probabilities for each timestep and relies on the CTC loss in training. 4 extra 1-D convolution layers with ReLU activation are used on top of the last Transformer encoder layer output. We also include LayerNorm between each of the layers.

3.5 Loss Functions

In this work, we use a so called hybrid CTC/attention loss (Watanabe et al., 2017) for our training process. Let $\mathbf{x} = [x_1, \cdots, x_T]$ be the input frame sequence at the input of Transformer encoder in the fusion module and $\mathbf{y} = [y_1, \cdots, y_L]$ being the targets, where T and L denote the input and target lengths, respectively.

The CTC loss assumes conditional independence between each output prediction and has a form of

$$p_{\text{CTC}}(\mathbf{y}|\mathbf{x}) \approx \prod_{t=1}^{T} p(y_t|\mathbf{x}) \quad (1)$$

On the other hand, an auto-regressive decoder gets rid of this assumption by directly estimating the posterior on the basis of the chain rule, which has a form of

$$p_{\text{CE}}(\mathbf{y}|\mathbf{x}) = \prod_{l=1}^{L} p(y_l|y_{<l}, \mathbf{x}) \quad (2)$$

The overall objective function is computed as follows:

$$\mathcal{L} = \lambda \log p_{\text{CTC}}(\mathbf{y}|\mathbf{x}) + (1-\lambda) \log p_{\text{CE}}(\mathbf{y}|\mathbf{x}) \quad (3)$$

where λ controls the relative weight between CTC loss and seq2seq loss in the hybrid CTC/attention mechanisms. The weight is needed not only when integrating the two losses into one training loss, but also fusing the two predictions during decoding, which we will revisit in the following subsections.

3.6 Training Pipeline

The final AVSR model is achieved through a pipeline of training stages.

For audio modality, the audio front-end is first pre-trained through self-supervised learning, which is done by wav2vec 2.0. Then the audio back-end is trained through the audio-only (AO) setting, together with a dedicated decoder.

For the visual modality, we first pre-train the 3-D convolution layer and visual back-end through sequence classification at word level video clips in LRW data. After that, the visual front-end are inherited by the visual-only (VO) model, where dedicated visual back-end and decoder are used.

The final AVSR model can be trained after the audio-only and visual-only models have converged.

Due to computational constraints, we pre-compute the audio and visual back-end outputs, and only learn the parameters in the fusion model and decoder part in this final stage. A detailed visualization of our training pipeline is depicted in Figure 2.

3.7 Decoding

Decoding is performed using joint CTC/attention one-pass decoding (Watanabe et al., 2017) with beam search. We apply shallow fusion to incorporate CTC and seq2seq predictions:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\hat{\mathcal{Y}}} \{\alpha \log p_{\text{CTC}}(\mathbf{y}|\mathbf{x}) + (1-\alpha)\log p_{\text{CE}}(\mathbf{y}|\mathbf{x})\} \tag{4}$$

where $\hat{\mathcal{Y}}$ denotes predictions set of target symbols, while α is the relative weight that tuned on validation set.

4 Experiments

In this section, we will first introduce the datasets and various settings we used in each component of our model. Then we will present results of audio-only, visual-only and audio-visual settings. We also present a breakdown of the relative contribution of every component through ablation study.

4.1 Dataset

We use the large-scale publicly AVSR dataset, the Lip Reading Sentences 2 (LRS2) (Chung et al., 2017) as our main testbed. During training, we also use the Lip Reading in the Wild (LRW) (Chung and Zisserman, 2016) as a word-level video classification task to pre-train our visual encoder.

LRS2 consists of 224 hours of aligned audio and videos, with a total of 144K clips from BBC videos, the clips are at a length of sentence level. The training data contains over 2M word instances and a vocabulary of over 40K. The dataset is very challenging as there are large variations in head pose, lighting conditions, genres and the number of speakers.

LRW is a word-level dataset, consisting of 157 hours of aligned audio and videos, totalling 489K video clips from BBC videos, each containing the utterance of a single word out of a vocabulary of 500. The videos have a fixed length of 29 frames, the target word occurring in the middle of the clip and surrounded by co-articulation. All of the videos are either frontal or near-frontal. In our experiment, we only use the visual modality from this dataset to train our visual front-end.

4.2 Experimental Settings

We use character level prediction with an output size of 40, consisting of the 26 characters in the alphabet, the 10 digits, the apostrophe, and special tokens for `[space]`, `[blank]` and `[EOS/SOS]`. Since the transcriptions of the datasets do not contain other punctuations, we do not include them in the vocabulary.

Our implementation is based on the Pytorch library (Paszke et al., 2019) and trained on four NVIDIA A100 GPUs with a total of 160GB memory. The network is trained using the Adam optimiser (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ and an initial learning rate of 10^{-4}. We use label smoothing with a weight set to 0.01, learning rate warm up and reduce on plateau. The relative weight in CTC loss and seq2seq loss λ is set to 0.2. When decoding, we set α to 0.1. The samples in the pre-train set are cropped by randomly sampling a continuous range of $1/3$ words of the whole utterances, in order to match the length of clips in the train set. Over-length samples are further truncated at 160 frames to reduce memory occupation.

Preprocessing: We detected and tracked 68 facial landmarks using dlib (King, 2009) for each video. To remove differences related to face rotation and scale, the faces are aligned to a neural reference frame using a similarity transformation following (Martinez et al., 2020). Interpolation and frame smoothing with a window width of 12 frames are used to deal with the frames that dlib fails to detect. Then a bounding box of 120×120 is used to crop the mouth ROIs. The cropped frame is further converted to gray-scale and normalized with respect to the overall mean and variance of the train set. Each raw audio waveform is normalized to zero mean and unit variance following (Baevski et al., 2020).

Data Augmentation: Following (Ma et al., 2021), random cropping with a size of 112×112 and horizontal flipping with a probability of 0.5 are performed consistently across all frames of a given image sequence when training visual-only and audio-visual models. For each audio waveform, additive noise is performed in the time domain following (Afouras et al., 2018a) during training audio-only and audio-visual models. Babble noise are added

to the audio stream with 5dB SNR and probability of $p_n = 0.25$. The babble noise is synthesized by mixing 20 different audio samples from LRS2.

Methods	WER
Visual-only	
LIBS (Zhao et al., 2020)	65.3
TM-CTC* (Afouras et al., 2018a)	54.7
Conv-seq2seq (Zhang et al., 2019)	51.7
TM-seq2seq* (Afouras et al., 2018a)	50.0
KD-TM (Ren et al., 2021)	49.2
LF-MMI TDNN* (Yu et al., 2020)	48.9
E2E Conformer* (Ma et al., 2021)	42.4
E2E Conformer** (Ma et al., 2021)	**37.9**
Our Model	43.2
Audio-only	
TM-CTC* (Afouras et al., 2018a)	10.1
TM-seq2seq* (Afouras et al., 2018a)	9.7
CTC/attention* (Petridis et al., 2018)	8.2
LF-MMI TDNN* (Yu et al., 2020)	6.7
E2E Conformer** (Ma et al., 2021)	3.9
Our Model	**2.7**
Audio-Visual	
TM-DCM (Lee et al., 2020b)	8.6
TM-seq2seq* (Afouras et al., 2018a)	8.5
TM-CTC* (Afouras et al., 2018a)	8.2
LF-MMI TDNN* (Yu et al., 2020)	5.9
E2E Conformer** (Ma et al., 2021)	3.7
Our Model	**2.6**

Table 3: Audio-only, visual-only and audio-visual results of word error rate (WER) tested on LRS2. Models with an * denote that results are using an external language model, which indicates an advantage over our model during evaluation. Models denoted with ** means that it uses a more powerful Transformer language model.

Evaluation: For all experiments, word error rate (WER) are reported which is defined as WER $= (S + D + I)/N$. The S, D and I in the formula denotes the number of substitutions, deletions and insertions respectively from the reference to the hypothesis, and N is the number of words in the inference. The babble noise added to the audio waveform during evaluation is generated using the same manner as training, while we set a different seed to avoid model fit to a specific generated noise. Decoding is performed using joint CTC/attention one-pass decoding (Watanabe et al., 2017) with beam width 5 (the values were determined on the held-out validation set of LRS2). We don't use an external language model in our experiments.

4.3 Results

We present results for all experiments in Table 3, reporting WERs on audio-only, visual-only and audio-visual models. Note that many of the models listed here are also using extra training data in different stages of training pipeline, such as MV-LRS (Chung and Zisserman, 2017), LRS3 (Afouras et al., 2018b), LibriSpeech (Panayotov et al., 2015) and LRW.

Audio-visual Setting: In the main audio-visual (AV) setting, the pre-train and train sets in LRS2 are used as train set in the final training stage. Our proposed audio-visual model achieves a WER of 2.6% without the help of an external language model, which improves by 1.1% over the current state-of-the-art (Ma et al., 2021). This is rather a big improvement, with a relative improvement of around 30%.

Audio-only Setting: The training data used for training audio-only model consists of 224 hours labelled data from LRS2, as well as the 60K hours unlabelled data from LibriLight (Kahn et al., 2020) that are indirectly used through inheriting wav2vec 2.0 parameters. Our model also achieves a WER of 2.7%, which reduces the WER of the current state-of-the-art (Ma et al., 2021) by 1.2%, indicating a relative improvement of 30%.

Visual-only Setting: The visual-only model uses labelled LRS2 data in its pre-train and train sets, the LRW for supervised pre-training, and indirectly using the 1.28M unlabelled images from ImageNet through MoCo v2. The visual-only model achieves a WER of 43.8%, lagging behind the current state-of-the-art (E2E Conformer) with 5.3%. Compared to E2E Conformer, the main difference is that a big Transformer language model is used during decoding, which itself brings a 4.5% difference compared with a normal RNN language model in their ablation study (Ma et al., 2021). The gap between our visual-only model and the E2E Conformer model with a RNN language model is 0.8%, which resides in a quite reasonable range. Additionally, we use a 6-layers Transformer encoder for temporal modelling instead of a 12-layers conformer encoder, which resulted in a smaller model size.

If we consider a fairer comparison by only looking at benchmarks without using an external lan-

guage model, the best-reported benchmark is Ren et al. (2021), which achieved a WER of 49.2%, lagging behind our model by 6.0%.

4.4 Ablation Studies

In this section, we investigate the impact of every individual building block by testing them in LRW, visual-only and audio-only settings.

MoCo v2 Contribution in Visual Word Classification: Results of visual word classification on LRW are shown in Table 4. We first train a model by replacing the ResNet-18 front-end in (Stafylakis and Tzimiropoulos, 2017) with a ResNet-50 front-end, matching the size of MoCo v2 but with fresh weights. This results in an absolute improvement of 2.1%. Then we initialize the ResNet-50 front-end with MoCo v2 weights and a further absolute improvement of 2.3% is observed, which implies that self-supervised learning is actually functioning in better represent the lip movement. Additionally, When Using 6 layers of Transformer encoder instead of TCN as back-end, we can observe another absolute improvement of 5.0%. We also noticed that using MoCo v2 front-end could significantly reduce the training time.

Method	Acc
Baseline(Stafylakis and Tzimiropoulos, 2017)	74.6%
+ ResNet-50 front-end	76.7%
+ MoCo v2 front-end	79.0%
+ Transformer encoder back-end	**85.0**%

Table 4: Ablation study on visual word classification performance on LRW.

Performance Breakdown in Audio-only Setting: Results of audio-only model on LRS2 are shown in Table 5. Starting from (Afouras et al., 2018a), we first train a model by replacing the STFT audio feature with a wav2vec 2.0 front-end pre-trained on LibriSpeech, resulting in an absolute improvement of 11.1%. Then we use another pre-trained model learned on an even larger unlabelled single modality dataset Libri-Light, and a further absolute improvement of 0.6% is observed. We further train the model with hybrid CTC/attention decoder during the training stage, which results in another absolute improvement of 0.9%.

Performance Breakdown in Visual-only Setting: Results of the visual-only model on LRS2 are shown in Table 6. Starting from (Afouras et al., 2018a), we first introduce end-to-end training by using a hybrid CTC/attention decoder (the front-end is still pre-trained through LRW), resulting in an absolute improvement of 16.0%. Then we initialize the front-end with MoCo v2 weights, a same end-to-end training manner results in a further absolute improvement of 5.8%.

Method	WER
Baseline(Afouras et al., 2018a)	15.3%
+ wav2vec 2.0 (LibriSpeech) encoder	4.2%
+ wav2vec 2.0 (LibriLight) encoder	3.6%
+ Hybrid CTC/attention	**2.7**%

Table 5: Ablation study on audio-only model performance on LRS2.

Method	WER
Baseline(Afouras et al., 2018a)	65.0%
+ Hybrid CTC/attention	49.0%
+ MoCo v2 front-end	**43.2**%

Table 6: Ablation study on visual-only model performance on LRS2.

Robustness under Noisy Inputs: To evaluate the model's tolerance to audio noise, we tested the performance of our model under babble noise with different SNR levels. Our audio-only and audio-visual models reach WERs of 32.5% and 24.5% when the SNR level is 0dB, respectively, which reduce the reported result in (Afouras et al., 2018a) by 25.5% and 9%[2]. When the SNR level rises to 5dB, our audio-only and audio-visual model obtain WERs of 6.8% and 6.3%.

Besides achieving significant improvement over the baseline model under babble noise environment, we further investigate the model performance under human noise environment. The human noise is extremely challenging cause the noise itself contains some words, while the model cannot easily distinguish which audio signal is the one to be recognized. We synthesize the human noise by randomly crop many 1 second signals from different audio samples in the LRS2 dataset. As shown in Fig. 3, we conduct experiments varying different levels of human noise, the models are trained using babble noise augmented audio. The WER increases greatly after the SNR level drops down under 0db. It is because the model may not be able to distinguish the two overlapped spoken words at a low

[2]Ma et al. (2021) also provides a performance under noisy inputs, however, we are not able to compare with them due to a lack of necessary details to generate the same noise.

SNR level.

And the overall performance under each SNR level is worse than babble noise, indicating that noise with specific information is harder than disorganized babble noise.

Modality	Model	0dB	5dB	clean
AO	Afouras et al. (2018a)	58.0%	-	10.5%
	Our model	**32.5%**	6.8%	**2.7%**
AV	Afouras et al. (2018a)	33.5%	-	9.4%
	Our model	**24.5%**	6.3%	**2.6%**

Table 7: Word error rate (WER) under different SNR levels. The noises are synthesized babble noises.

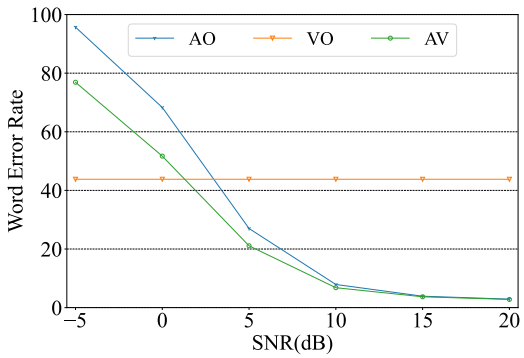


Figure 3: Word error rate (WER) under different SNR levels. The noises are human speech sampled from LRS2. AO: Audio-Only model, VO: Visual-Only model, AV:Audio-Visual model

Recognition under Low Resource: A significant benefit of using self-supervised pre-trained models is that only a small amount of labelled data is needed for training a model. To further investigate the models' performance in low resource environment, we use the 28 hours train set of LRS2 to train an audio-only and a visual-only model. The results are shown in Table 8. The audio-only model trained with 28 hours data achieves a WER of 3.4%, which is a little bit worse than the one trained with 224 hours data. The result indicates that for the audio-only model, the self-supervised model pre-trained on a large-scale single modality dataset can significantly reduce the demands of data. While the visual-only model trained with 28 hours data has a great gap with the one trained with 224 hours data, the reason can be that the visual-only model is harder to train and demands a larger amount of data.

Model	Training data (Hours)	WER (%)
audio-only	LRS2 (224)	2.7
	LRS2 train set (28)	3.4 (+0.7)
visual-only	LRS2 (224)	43.2
	LRS2 train set (28)	68.9 (+25.7)

Table 8: Performance of audio-only and visual-only models using different training data.

4.5 Discussion and Conclusion

In this work, we propose to utilize self-supervised learning for AVSR by simply incorporating the pre-trained model trained in massive unlabelled single modality data. Although the visual pre-trained models are not straight-forward to be transplanted into visual front-end, we still manage to integrate pre-trained models in both modalities for the AVSR task. Experimental results are impressive, resulting in a 30% relative improvement.

It's interesting to observe that self-supervised model in audio modality has an even larger improvement than that of the visual counterpart. We believe the reasons can be listed as follows:

- The training data scale of audio modality is significantly larger than that of visual modality, with the Libri-Light dataset used for pre-training wav2vec 2.0 consists of 60K hours audio signals, the ImageNet dataset, on the contrary, has only 1.28M images, roughly equivalent to 14 hours silent video under 25 FPS.

- The MoCo v2 model is pre-trained on images to better represent frame-level contents, while there are no pre-training steps to model the temporal correlation between frames. In contrast, the wav2vec 2.0 model is pre-trained on consistent audios, thus having a better temporal modelling ability.

As there has not emerged a dominating cross-modality self-supervised learning approach in the field of AVSR, in future work, we are going to explore two more directions in the self-supervised learning scenario based on this work. The first is utilizing the temporal correlations within the visual domain, while the other is the cross-modal correlations between the audio and visual modality. We hope this work could pave the way towards multimodality self-supervised learning, especially for various aspects in audio-visual speech recognition.

References

T. Afouras, J. Chung, A. Senior, O. Vinyals, and A. Zisserman. 2018a. Deep audio-visual speech recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence, pages 1–1.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2018b. Lrs3-ted: a large-scale dataset for visual speech recognition. arXiv preprint arXiv:1809.00496.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2020. Asr is all you need: Cross-modal distillation for lip reading. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2143–2147. IEEE.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. arXiv preprint arXiv:1607.06450.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. arXiv preprint arXiv:2006.11477.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020b. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. Voxceleb2: Deep speaker recognition. arXiv preprint arXiv:1806.05622.

Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2017. Lip reading sentences in the wild. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3444–3453. IEEE.

Joon Son Chung and Andrew Zisserman. 2016. Lip reading in the wild. In Asian conference on computer vision, pages 87–103. Springer.

Joon Son Chung and AP Zisserman. 2017. Lip reading in profile.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. 2015. Unsupervised visual representation learning by context prediction. In Proceedings of the IEEE international conference on computer vision, pages 1422–1430.

S. Dupont and J. Luettin. 2000. Audio-visual speech modeling for continuous speech recognition. IEEE Transactions on Multimedia, 2(3):141–151.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pages 1735–1742. IEEE.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B Girshick. 2019. Momentum contrast for unsupervised visual representation learning. corr abs/1911.05722 (2019). arXiv preprint arxiv:1911.05722.

Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. 2020. Libri-light: A benchmark for asr with limited or no supervision. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7669–7673. IEEE.

Davis E King. 2009. Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research, 10:1755–1758.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Yong-Hyeok Lee, Dong-Won Jang, Jae-Bin Kim, Rae-Hong Park, and Hyung-Min Park. 2020a. Audio–visual speech recognition based on dual cross-modality attentions with the transformer model. Applied Sciences, 10(20):7263.

Yong-Hyeok Lee, Dong-Won Jang, Jae-Bin Kim, Rae-Hong Park, and Hyung-Min Park. 2020b. Audio–visual speech recognition based on dual cross-modality attentions with the transformer model. Applied Sciences, 10(20):7263.

Wei Li, Sicheng Wang, Ming Lei, Sabato Marco Siniscalchi, and Chin-Hui Lee. 2019. Improving audio-visual speech recognition performance with cross-modal student-teacher training. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6560–6564. IEEE.

Pingchuan Ma, Stavros Petridis, and Maja Pantic. 2021. End-to-end audio-visual speech recognition with conformers. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7613–7617. IEEE.

Brais Martinez, Pingchuan Ma, Stavros Petridis, and Maja Pantic. 2020. Lipreading using temporal convolutional networks. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6319–6323. IEEE.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 5206–5210. IEEE.

Georgios Paraskevopoulos, Srinivas Parthasarathy, Aparna Khare, and Shiva Sundaram. 2020. Multiresolution and multimodal speech recognition with transformers. arXiv preprint arXiv:2004.14840.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32:8026–8037.

Stavros Petridis, Themos Stafylakis, Pingchuan Ma, Georgios Tzimiropoulos, and Maja Pantic. 2018. Audio-visual speech recognition with a hybrid ctc/attention architecture. In 2018 IEEE Spoken Language Technology Workshop (SLT), pages 513–520. IEEE.

Sucheng Ren, Yong Du, Jianming Lv, Guoqiang Han, and Shengfeng He. 2021. Learning from the master: Distilling cross-modal advanced knowledge for lip reading. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13325–13333.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. arXiv preprint arXiv:1904.05862.

Abhinav Shukla, Konstantinos Vougioukas, Pingchuan Ma, Stavros Petridis, and Maja Pantic. 2020. Visually guided self supervised learning of speech representations. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6299–6303. IEEE.

Themos Stafylakis and Georgios Tzimiropoulos. 2017. Combining residual networks with lstms for lipreading. arXiv preprint arXiv:1703.04105.

Fei Tao and Carlos Busso. 2020. End-to-end audiovisual speech recognition system with multitask learning. IEEE Transactions on Multimedia, 23:1–11.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. 2017. Hybrid ctc/attention architecture for end-to-end speech recognition. IEEE Journal of Selected Topics in Signal Processing, 11(8):1240–1253.

Jianwei Yu, Shi-Xiong Zhang, Jian Wu, Shahram Ghorbani, Bo Wu, Shiyin Kang, Shansong Liu, Xunying Liu, Helen Meng, and Dong Yu. 2020. Audio-visual recognition of overlapped speech for the lrs2 dataset. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6984–6988. IEEE.

Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful image colorization. In European conference on computer vision, pages 649–666. Springer.

Xingxuan Zhang, Feng Cheng, and Shilin Wang. 2019. Spatio-temporal fusion based convolutional sequence learning for lip reading. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 713–722.

Ya Zhao, Rui Xu, Xinchao Wang, Peng Hou, Haihong Tang, and Mingli Song. 2020. Hearing lips: Improving lip reading by distilling speech recognizers. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 6917–6924.

A Decoding Algorithm

Algorithm 1 Hybrid CTC/attention one-pass decoding adapted from (Watanabe et al., 2017). Notation: X is the speech input; L_{max} is the maximum length of the hypotheses to be searched, we set it to T; C is the decoded symbol sequence; `[b]` denotes `[blank]`.

Input: X, L_{max}
Output: C

$\Omega_0 = \{\texttt{[SOS]}\}$
$\hat{\Omega} = \emptyset$
$\gamma_0^{(b)}(\texttt{[SOS]}) = 1$
for $t = 1, \cdots, T$ **do**
  $\gamma_t^{(n)}(\texttt{[SOS]}) = 0$
  $\gamma_t^{(b)}(\texttt{[SOS]}) = \prod_{\tau=1}^{t} \gamma_{\tau-1}^{(b)}(\texttt{[SOS]}) \cdot p(z_\tau = \texttt{[b]}|X)$
end for
for $l = 1 \cdots L_{max}$ **do**
  $\Omega_l = \emptyset$
  while $\Omega_{l-1} \neq \emptyset$ **do**
    $g = \text{HEAD}(\Omega_{l-1})$
    DEQUEUE(Ω_{l-1})
    for each $c \in \mathcal{U}$ **do**
      $h = g \cdot c$
      if $c = \texttt{[EOS]}$ **then**
        $\log p_{\text{ctc}}(h|X) = \log\{\gamma_T^{(n)}(g) + \gamma_T^{(b)}(g)\}$
      else
        if $g = \texttt{[SOS]}$ **then**
          $\gamma_1^{(n)}(h) = p(z_1 = c|X)$
        else
          $\gamma_1^{(n)}(h) = 0$
        end if
        $\gamma_1^{(b)}(h) = 0$
        $\Psi = \gamma_1^{(n)}(h)$
        for $t = 2 \cdots T$ **do**
          if $last(g) = c$ **then**
            $\Phi = \gamma_{t-1}^{(b)}(g)$
          else
            $\Phi = \gamma_{t-1}^{(b)}(g) + \gamma_{t-1}^{(n)}(g)$
          end if
          $\gamma_t^{(n)}(h) = (\gamma_{t-1}^{(n)}(h) + \Phi) p(z_t = c|X)$
          $\gamma_t^{(b)}(h) = (\gamma_{t-1}^{(b)}(h) + \gamma_{t-1}^{(n)}(h)) p(z_t = \texttt{[b]}|X)$
          $\Psi = \Psi + \Phi \cdot p(z_t = c|X)$
        end for
        $\log p_{\text{ctc}}(h|X) = \log(\Psi)$
      end if
      $\log p(h|X) = \alpha \log p_{\text{ctc}}(h|X) + (1-\alpha) \log p_{\text{att}}(h|X)$
      if $c = \texttt{[EOS]}$ **then**
        ENQUEUE$(\hat{\Omega}, h)$
      else
        ENQUEUE(Ω_l, h)
      end if
    end for
  end while
  $\Omega_l = \text{TOPK}(\Omega_l, W)$
end for
return $\arg\max_{C \in \hat{\Omega}} \log p(C|X)$

Algorithm 1 describes the hybrid CTC/attention decoding procedure. The CTC prefix probability is defined as the cumulative probability of all label sequences that have h as their prefix:

$$p_{\text{ctc}}(h|X) = \sum_{v \in (\mathcal{U})^+} p_{\text{ctc}}(h \cdot v|X) \quad (5)$$

where v denotes all possible symbol sequences except the empty. The CTC probability can be computed by keeping the forward hypothesis probabilities $\gamma_t^{(n)}$ and $\gamma_t^{(b)}$, where the superscripts (n) and (b) represents all CTC paths end with a non-`[blank]` or `[blank]` symbol, respectively.

The decoding algorithm is also a beam search with width W and hyperparameter α control the relative weight given to CTC and attention decoding. $\mathcal{U}$ is a set of symbols excluding `[blank]`, and a same token is used to represent `[SOS]` and `[EOS]` in our implementation.

B Decoding Examples

AO: WHATEVER YOU ASK
AV: WHATEVER YOU ARE
AO: TRAVEL THREE MILES URBER WEST AND YOU DO GET MORE FOR YOUR MONEY HERE
AV: TRAVEL THREE MILES FURTHER WEST AND YOU DO GET MORE FOR YOUR MONEY HERE
AO: IT COULD BE YOUR PASSPORT FOR A SMALL FORTUNE
AV: IT COULD BE YOUR PASSPORT TO A SMALL FORTUNE
AO: WHAT TO THINK FOR THEMSELVES
AV: NOT TO THINK FOR THEMSELVES
AO: NOT THE SUBJECT MATTERING
AV: NOT FOR SUBJECT MATTER
AO: I WOULDN'T SAY I'M THE STAR
AV: I WOULDN'T SAY I'M A STAR
AO: CRISPAS PUDDING THAT NOBODY REALLY LIKES
AV: CHRISTMAS PUDDING THAT NOBODY REALLY LIKES
AO: BUT AT THE SAME TIME
AV: AT THE SAME TIME
AO: BEING ~~ON~~ MY OWN
AV: BEING MY OWN
AO: ~~SO~~ AT ONE POINT
AV: AT ONE POINT

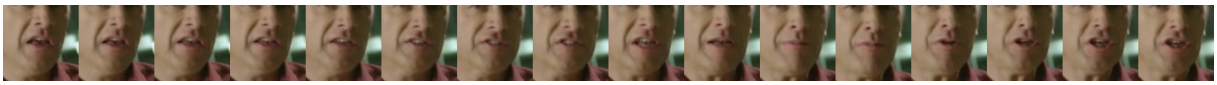
Table 9: AO (audio-only) and AV (audio-visual) decoding examples. Underline denotes substitutions and insertions error; Strikethrough denotes deletions error.

Table 9 is examples of sentences that audio-only model fails to predict while audio-visual model

(a) Landmarks detected by dlib. Green dots are 68 landmarks, frames without landmarks are ones that dlib fail to detect.

(b) Landmarks after linear interpolation.

(c) Faces smoothed with a window width of 12 and aligned to a neural reference frame using a similarity transformation.



(d) Mouth ROIs cropped using a bounding box of 120×120.

Figure 4: Preprocessing example to illustrate the process to generate mouth ROIs.

correctly predicts. The visual modality enhances the model from a wide range of error cases.

C Preprocessing Example

The input images are sampled at 25 FPS and resized to 224×224 pixels. We crop a 120×120 mouth ROI from each frame. Fig. 4 shows the process to generate.