Supplementary Information: Visual Speech Recognition for Multiple Languages in the Wild

Pingchuan Ma[1] , Stavros Petridis[1,2] , Maja Pantic[1,2]

[1]*Imperial College London, Department of Computing, London, U.K.*
[2]*Meta AI, London, UK*

Table S1: Details of Audio-Visual Datasets used in this work. CM_{xx} and MT_{xx} denote the particular language parts of the CMU-MOSEAS and Multilingual TEDx datasets, respectively, where xx denotes the standard language codes, conforming to the ISO 639-1 standard.

Dataset	Transcription	Hours
Publicly Available Datasets		
LRW [1]	✓	157
LRS2 [2]	✓	223
LRS3 [3]	✓	438
CMLR [4]	✓	61
MT_{es} [5]	✓	71
MT_{it}[5]	✓	46
MT_{pt} [5]	✓	81
MT_{fr} [5]	✓	85
CM_{es} [6]	✓	16
CM_{pt} [6]	✓	18
CM_{fr} [6]	✓	15
AVSpeech [7]	✗	641
Non-Publicly Available Datasets		
MVLRS [2]	✓	730
LSVSR [8]	✓	3 886
YT-31k [9]	✓	31 000
YT-90k [10]	✓	90 000
$VoxCeleb2^{clean}$ [11]	✗	334

S1 Datasets Details

Details about the audio-visual datasets used in this study are presented in Table S1. It is clear that the non-publicly available datasets are one to two orders of magnitude larger than the publicly available ones.

S2 Architecture Details

The model consists of 4 modules, a front-end encoder (VSR encoder in Fig. 1c), a back-end encoder, a hybrid CTC and transformer decoder and two predictors. In particular, the encoder receives as input the raw images and maps them to visual speech representations which are fed to the back-end encoder. This is followed by a CTC and transformer decoder which generates the predicted characters. Finally, the features extracted from the middle position of the back-end encoder flow through two separate predictors to predict visual and acoustic speech representations from pre-trained VSR and ASR models, respectively.

The **front-end encoder** consists of a 3D convolutional layer with a kernel size of $5 \times 7 \times 7$ followed by a ResNet-18 [12, 13]. Let $B \times T \times H \times W$ be the input tensor to the visual front-end module, where B, T, H, and W correspond to batch size, number of frames, height and width, respectively. The visual features at the top of the residual blocks are aggregated along the spatial dimension by a global average pooling layer, resulting in a feature output of dimensions $B \times C \times T$, where C indicates the channel dimensionality. The Swish activation functions is used in all layers. The detailed architecture can be seen in Table S2.

The **back-end encoder** starts with a positional embedding module, followed by a stack of 12 conformer blocks. The positional embedding module is a linear layer, which projects the features from the output of ResNet-18 to a 256-dimensional space. The transformed features are further injected with relative position information [14]. In each conformer block, a feed-forward module, a self-attention module, a convolution module, and a second feed-forward module are stacked in order. Specifically, the feed-forward module is comprised of a linear layer, which projects the features to a higher 2048-dimensional space, followed by a Rectified Linear Unit (ReLU) activation function, a dropout layer with a probability of 0.1, and a second linear layer with output dimension of 256. Half-step residual connections are also used in each feed-forward module. The self-attention module is capable of modeling global dependencies among elements. The module maps the query and a set of key-value pairs through an attention map, which focuses on different parts of the

input. Instead of performing a single attention function, a multi-head mechanism is leveraged with different linear projections to a lower 64-dimensional space. The attention function is performed in parallel on each head and the outputs are concatenated into a 256-dimensional space and once again projected into the final values. The convolutional module, which excels at capturing local patterns efficiently, is composed of an 1D point-wise convolutional layer, Gated Linear Units (GLU) [15], an 1D depthwise convolutional layer, a batch normalisation layer, a swish activation layer, a 1D point-wise convolutional layer, and a layer normalisation layer. The combination of self-attention and convolution is capable of better capturing both local and global temporal information compared to the standard transformer architecture [16].

The **decoder** is composed of an embedding module and a set of residual multi-head attention blocks. It takes as input the encoded sequence and the prefixes of the target sequence. First, the prefixes from index 1 to l - 1 are projected to embedding vectors, where l is the target length index. The absolute positional encoding [17] is also added to the embedding. Next, the embedding is fed to a stack of multi-head attention blocks. Each block consists of a self-attention module, an encoder-decoder attention module and a feed-forward module. Layer normalisation is added before each module. Specifically, the self-attention module is slightly different from the one in the encoder where future positions at its attention matrix are masked out, followed by an encoder-decoder attention, which helps the decoder to focus on the relevant part of the input. This attention receives the features from the previous self-attention module as Q and the features from the encoder as K and V ($K = V$). The features are further fed to a feed-forward module, which is the same as the one used in the encoder. Finally, a layer normalisation and a linear layer are added which predict the posterior distribution of the next generated token.

A **linear layer** with a softmax function, which maps the encoded features to the predicted character sequence is also used on top of the back-end encoder . This layer is trained with the Connectionist Temporal Classification (CTC) loss.

The **predictor** is a linear layer which takes as input the features at the middle block (6th) of the back-end encoder and predicts the corresponding audio/visual features from the pre-trained ASR/VSR models. Separate predictors are employed for each prediction task. Both the input and output dimensions of the linear layer are 256.

S3 Pre-trained VSR and ASR models

The pre-trained ASR and VSR models are shown in Fig. 1a and 1b, respectively. The pre-trained VSR model has exactly the same architecture as the full model described in section S2 but does not include any predictors. The pre-trained ASR model replaces the VSR encoder with an ASR encoder and its architecture can be seen in Fig. 1d and Table S3. It should be noted that these models are always trained on the same data as the full model. Then the pre-trained ASR/VSR encoders and some conformer layers are frozen and their internal representations are used as targets for the audio and visual predictors as shown in Fig. 1c. The performance of the pre-trained models for all languages can be seen in Tables S4, S5, S6 and S7.

S4 Hyperparameter Optimisation

The main hyperparameter that was found to have a significant impact on performance was the batch size. We observed that increasing the batch size from 8 to 16 led to reduced WER on the validation set of the LRS2 dataset (see Table S10). There is also one more hyper-parameter which controls the batch size based on the length of the sequences. In other words, if some sequences are too long then the batch is halved. We found that increasing this threshold from 150 to 220 frames also improved the performance. We could not increase these two hyper-parameters even further due to GPU memory constraints but it is likely that the WER will be reduced even more.

S5 Language Models

We train three monolingual transformer-based language model [18] for 50 epochs. The English language model is trained by combining the training sets of LibriSpeech (960 h) [19], pre-training and training sets of LRS2 [2] and LRS3 [3], TED-LIUM 3 [20], Voxforge (English) and Common Voice (English) [21], with a total of 166 million characters. The Mandarin language model is trained by combining the CMLR [4] and news2016zh, with a total of 153 million characters. The Spanish language model is trained by combining the Spanish corpus from Multilingual TEDx [5], Common Voice [21] and Multilingual LibriSpeech [22], with a total of 192 million characters. The Italian language model is trained by combining the Italian corpus from Multilingual TEDx [5], Common Voice [21] and Multilingual LibriSpeech [22], with a total of 252 million characters. The Portuguese language model is trained by combining the Portuguese corpus from Multilingual TEDx [5], Common Voice [21] and Multilingual LibriSpeech [22], with a total of 85 million characters. The French language model is trained by combining the French corpus from Multilingual TEDx [5], Common Voice [21] and Multilingual LibriSpeech [22], with a total of 945 million characters. In our work, we set λ and β from equation 3 to 0.1 and {English: 0.6, Mandarin: 0.3, Spanish: 0.4, Italian: 0.5, Portuguese: 0.3, French: 0.3}, respectively. The impact of the improved English language model on the validation set of the LRS2 dataset can be seen on Table S10.

Component Name	Layer Type	Input Size	Output Size
Stem_1	Conv 3D, 5×7^2, 64	[B, 1, T_v, 88, 88]	[B, 64, T_v, 44, 44]
	3D Max Pooling, 1×3^2	[B, 64, T_v, 44, 44]	[B, 64, T_v, 22, 22]
Reshape	-	[B, 64, T_v, 22, 22]	[B×T_v, 64, 22, 22]
Residual Block_2	$\begin{bmatrix}\text{Conv 2D}, 3^2, 64\\ \text{Conv 2D}, 3^2, 64\end{bmatrix} \times 2$	[B×T_v, 64, 22, 22]	[B×T_v, 64, 22, 22]
Residual Block_3	$\begin{bmatrix}\text{Conv 2D}, 3^2, 128\\ \text{Conv 2D}, 3^2, 128\end{bmatrix} \times 2$	[B×T_v, 64, 22, 22]	[B×T_v, 128, 11, 11]
Residual Block_4	$\begin{bmatrix}\text{Conv 2D}, 3^2, 256\\ \text{Conv 2D}, 3^2, 256\end{bmatrix} \times 2$	[B×T_v, 128, 11, 11]	[B×T_v, 256, 6, 6]
Residual Block_5	$\begin{bmatrix}\text{Conv 2D}, 3^2, 512\\ \text{Conv 2D}, 3^2, 512\end{bmatrix} \times 2$	[B×T_v, 256, 6, 6]	[B×T_v, 512, 3, 3]
Aggregation	2D Global Average Pooling	[B×T_v, 512, 3, 3]	[B×T_v, 512, 1, 1]
Reshape	-	[B×T_v, 512, 1, 1]	[B, 512, T_v]

Table S2: The architecture of the front-end encoder of the VSR model. The filter shapes are denoted by {Temporal Size × Spatial Size^2, Channels} and {Spatial Size^2, Channels} for 3D convolutional and 2D convolutional Layers , respectively. The sizes correspond to [Batch Size, Channels, Sequence Length, Height, Width] and [Batch Size × Sequence Length, Channels, Height, Width], for 3D and 2D convolutional layers, respectively. T_v denotes the number of input frames.

Component Name	Layer Type	Input Size	Output Size
Stem_1	Conv 1D, 80, 64	[B, 1, T_a]	[B, 64, T_a//4]
Residual Block_2	$\begin{bmatrix}\text{Conv 1D}, 3, 64\\ \text{Conv 1D}, 3, 64\end{bmatrix} \times 2$	[B, 64, T_a//4]	[B, 64, T_a//4]
Residual Block_3	$\begin{bmatrix}\text{Conv 1D}, 3, 128\\ \text{Conv 1D}, 3, 128\end{bmatrix} \times 2$	[B, 64, T_a//4]	[B, 128, T_a//8]
Residual Block_4	$\begin{bmatrix}\text{Conv 1D}, 3, 256\\ \text{Conv 1D}, 3, 256\end{bmatrix} \times 2$	[B, 128, T_a//8]	[B, 256, T_a//16]
Residual Block_5	$\begin{bmatrix}\text{Conv 1D}, 3, 512\\ \text{Conv 1D}, 3, 512\end{bmatrix} \times 2$	[B, 256, T_a//16]	[B, 512, T_a//32]
Aggregation	1D Average Pooling, Stride 20	[B, 512, T_a//32]	[B, 512, T_a//640]

Table S3: The architecture of the front-end encoder of the ASR model. The filter shapes are denoted by {Temporal Size, Channels} for 1D Convolutional Layers, respectively. The sizes correspond to [Batch Size, Channels, Sequence Length]. T_a denotes the length of audio waveforms.

S6 Time Masking

We mask n consecutive frames with the mean frame of the video. The duration t_n is chosen from 0 to an upper bound $n_{\max}$ using a uniform distribution. Since there is a large variance in the video lengths of the LRS2 and LRS3 datasets, we set the number of masks proportional to the sequence length. Specifically, we use one mask per second, and for each mask, the maximum duration $n_{\max}$ is set to 0.4 seconds.

Table S4: Performance (Mean±Std) of the pre-trained ASR and VSR models on the LRS2 dataset.

Method	Training Sets	Full Model	Pre-trained VSR model	Pre-trained ASR model
Ours	LRS2	**33.6±0.5**	33.4±0.3	4.0±0.4
Ours	LRW+LRS2	**29.5±0.4**	33.2±0.5	3.9±0.2
Ours	LRW+LRS2+LRS3	**27.6±0.2**	29.3±0.4	3.7±0.1
Ours	LRW+LRS2+LRS3+AVSpeech	**25.8±0.4**	29.3±0.4	3.7±0.1

Table S5: Performance (Mean± Std) of the pre-trained ASR and VSR models on the LRS3 dataset.

Method	Training Sets	Full Model	Pre-trained VSR model	Pre-trained ASR model
Ours	LRS3	**38.6±0.4**	38.7±0.5	2.3±0.1
Ours	LRW+LRS3	**35.8±0.5**	38.6±0.4	2.2±0.1
Ours	LRW+LRS2+LRS3	**34.9±0.2**	35.2±0.2	2.0±0.2
Ours	LRW+LRS2+LRS3+AVSpeech	**32.1±0.3**	35.2±0.2	2.0±0.2

Table S6: Performance (Mean±Std) of the pre-trained ASR and VSR models on the CMLR dataset.

Method	Training Sets	Full Model	Pre-trained VSR model	Pre-trained ASR model
Ours	CMLR	**9.1±0.05**	10.7±0.06	2.5±0.03
Ours	LRW+LRS2+LRS3+CMLR	**8.2±0.06**	9.0±0.05	2.2±0.03
Ours	LRW+LRS2+LRS3+AVSpeech+CMLR	**8.1±0.05**	8.9±0.08	2.2±0.03

Table S7: Performance (Mean±Std) of the pre-trained ASR and VSR models on the CMU-MOSEAS-Spanish (CM_{es}) dataset.

Method	Training Sets	Full Model	Pre-trained VSR model	Pre-trained ASR model
Ours	LRW+CM_{es}+MT_{es}	**51.5±0.8**	53.2±0.4	16.3±0.3
Ours	LRW+LRS2+LRS3+CM_{es}+MT_{es}	**47.4±0.2**	47.5±0.6	15.4±0.1
Ours	LRW+LRS2+LRS3+AVSpeech+CM_{es}+MT_{es}	**44.6±0.6**	45.3±0.4	15.4±0.1

S7 Loss Functions

To map input sequences $\mathbf{x} = [x_1, ..., x_T]$ such as audio or visual streams to corresponding target characters $\mathbf{y} = [y_1, ..., y_L]$, we consider a hybrid CTC/attention architecture [23] in this paper, where T, L are the lengths of the input sequence and target character sequence, respectively. The CTC loss assumes conditional independence between the output predictions and the estimated sequence posterior has the form of $P_{CTC}(\mathbf{y}|\mathbf{x}) \approx \prod_{t=1}^{T} p(y_t|\mathbf{x})$. The CTC loss from equation 5 is defined as follows:

$$\mathcal{L}_{CTC} = -\log P_{CTC}(\mathbf{y}|\mathbf{x}) \tag{S1}$$

An attention-based encoder-decoder model gets rid of this assumption by directly estimating the posterior on the basis of the chain rule and has a form of $P_{att}(\mathbf{y}|\mathbf{x}) \approx \prod_{l=1}^{L} p(y_l|y_{<l}, \mathbf{x})$. In this case the $\mathcal{L}_{att}$ from equation is:

$$\mathcal{L}_{att} = -\log P_{att}(\mathbf{y}|\mathbf{x}) \tag{S2}$$

The objective function of speech recognition is performed by a linear combination of the CTC loss and a cross-entropy loss as shown in equation 5. The α value used in this work is 0.1.

A grid search was performed for the parameters β_a and β_v used in the auxiliary loss (equation 3). The values that resulted in the best performance in the validation set of the LRS2 dataset are the following: $\beta_a = 0.4$ and $\beta_v = 0.4$. These values are used for all experiments.

Table S8: Results of curriculum learning experiments on the LRS2 dataset.

Video Length in Frames	WER on the Validation Set	WER on the Test Set
Base VSR Model		
0-100	65.1±0.2	52.7±0.8
0-150	54.0±0.7	44.2±0.5
0-300	46.0±0.6	36.3±0.4
0-450	43.6±0.5	34.3±0.5
0-600	42.4±0.4	33.7±0.4
VSR Model with Auxiliary Workers		
0-100	51.9±0.3	41.5±0.5
0-150	46.2±0.4	36.1±0.3
0-300	43.3±0.2	34.4±0.2
0-450	42.6±0.3	34.6±0.5
0-600	42.0±0.3	33.4±0.3

Table S9: Results of curriculum learning experiments on the LRS3 dataset.

Video Length in Frames	WER on the Test Set
Base VSR model	
0-100	75.2±0.4
0-150	53.3±0.7
0-300	43.0±0.4
0-450	39.9±0.6
0-600	38.7±0.5
VSR Model with Auxiliary Workers	
0-100	57.7±0.4
0-150	46.8±0.1
0-300	40.8±0.6
0-450	39.7±0.4
0-600	38.6±0.4

S8 Curriculum Learning

The end-to-end model was trained from scratch, resulting in poor performance on LRS2 and LRS3. This is likely due to the vast amount of very long utterances featured in LRS2 and LRS3, which makes learning from scratch especially challenging. We have found that the issue can be resolved by progressively training the end-to-end model, starting with short utterances and then using longer ones during training. This approach is commonly called curriculum learning (CL). In this paper, the model is initially trained with a subset of labelled training data, consisting of videos shorter than 100 frames. Then this model is used for initialisation when using utterances with up to 150 frames for training. This process is repeated for 3 more rounds where the length of training sequences is 300, 450, and 600 frames, respectively. Results for each round of curriculum learning can be seen in Tables S8 and S9.

Table S10: Investigation of the impact of hyperparameters and Language Model (LM) choices on the validation set of LRS2 dataset.

Method	WER
CM-seq2seq [24] - Baseline	47.7±0.5
+ Hyperparameter Optimisation	45.6±0.4
+ Improved LM	44.1±0.5

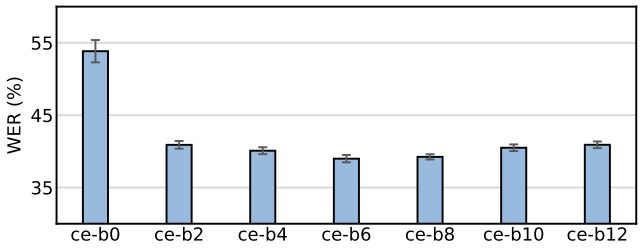

Figure S1: Performance of visual speech recognition on the LRS2 validation set as a function of the layer where the auxiliary loss is attached (see equation 3). "ce-b0" to "ce-b12" refer to the conformer layers from bottom to top.

S9 Additional Results

S9.1 Ablation Study on the Effect of Layer Position

We investigate the effect of the layer l where the auxiliary loss (equation 3) is attached. The position of layer varies from 0 to 12 at intervals of 2. Layer 6 was found to be the optimal level on the validation set of LRS2. Results are presented in Figure S1.

S9.2 Results on Spanish

Results on the Multilingual TEDx-Spanish dataset are shown in Table S11. We observe that our proposed approach results in a 5.6 % absolute reduction in the WER. A further reduction of 4.2 % can be achieved by using additional training data.

S9.3 Results on Italian

We manually clean the Italian corpus on Multilingual TEDx to exclude videos without visible speakers, result-

Table S11: Results on the Multilingual TEDx-Spanish (MT_{es}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [24]	LRW	CM_{es}+MT_{es}	244	66.4±0.8	65.2
Ours	LRW	CM_{es}+MT_{es}	244	**60.8±0.8**	**60.3**
Ours	LRW+LRS2+LRS3	CM_{es}+MT_{es}	905	**56.9±0.5**	**56.5**
Ours	LRW+LRS2+LRS3+AVSpeech	CM_{es}+MT_{es}	1 546	**56.6±0.3**	**56.3**

Table S12: Results on the Multilingual TEDx-Italian (MT_{it}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [24]	LRW	MT_{it}	203	71.5±0.4	70.9
Ours	LRW	MT_{it}	203	**65.9±0.5**	**65.2**
Ours	LRW+LRS2+LRS3	MT_{it}	864	**58.7±0.3**	**58.2**
Ours	LRW+LRS2+LRS3+AVSpeech	MT_{it}	1 505	**57.9±0.7**	**57.4**

Table S13: Results on the Multilingual TEDx-Portuguese (MT_{pt}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [24]	LRW	CM_{pt}+MT_{pt}	256	70.2±0.3	69.7
Ours	LRW	CM_{pt}+MT_{pt}	256	**66.0±0.5**	**65.3**
Ours	LRW+LRS2+LRS3	CM_{pt}+MT_{pt}	917	**62.4±0.4**	**62.0**
Ours	LRW+LRS2+LRS3+AVSpeech	CM_{pt}+MT_{pt}	1 558	**62.1±0.6**	**61.5**

Table S14: Results on the CMU-MOSEAS-Portuguese (CM_{pt}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [24]	LRW	CM_{pt}+MT_{pt}	256	65.7±0.5	65.4
Ours	LRW	CM_{pt}+MT_{pt}	256	**57.2±0.7**	**56.6**
Ours	LRW+LRS2+LRS3	CM_{pt}+MT_{pt}	917	**53.1±0.2**	**52.8**
Ours	LRW+LRS2+LRS3+AVSpeech	CM_{pt}+MT_{pt}	1 558	**51.6±0.2**	**51.4**

ing in a total of 26387 videos (45.8 hours) for training, 252 videos (0.4 hours) for validation and 309 videos (0.5 hours) for testing. Results on the Multilingual TEDx-Italian dataset are shown in Table S12. Our proposed approach results in an absolute drop of 5.6 % in the WER. A further reduction of 8 % can be achieved by using additional training data.

S9.4 Results on Portuguese

We manually cleaned the Portuguese corpus on Multilingual TEDx to exclude videos where the speaker is not visible, resulting in a total of 52 395 videos (81.3 hours) for training, 532 videos (0.7 hours) for validation and 401 videos (0.6 hours) for testing. Results on the Multilingual TEDx-Portuguese dataset are shown in Table S13. We observe that our proposed approach results in a 4.2 % absolute reduction in the WER. A further reduction of 3.9 % can be achieved by using additional training data.

We divide the Portuguese corpus on CMU-MOSEAS [6] into 10 658 videos (17.8 hours) for training and 412 videos (0.7 hours) for testing, respectively. Results on the CMU-MOSEAS-Portuguese dataset are shown in Table S14. The proposed approach results in a 8.5 % absolute reduction in the WER. Using additional training data leads to a further reduction of 5.6 %.

Table S15: Results on the Multilingual TEDx-French (MT_{fr}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [24]	LRW	CM_{fr}+MT_{fr}	257	84.0±0.7	83.2
Ours	LRW	CM_{fr}+MT_{fr}	257	**74.6±0.6**	**73.4**
Ours	LRW+LRS2+LRS3	CM_{fr}+MT_{fr}	918	**67.0±0.3**	**66.7**
Ours	LRW+LRS2+LRS3+AVSpeech	CM_{fr}+MT_{fr}	1 559	**67.0±0.6**	**66.2**

Table S16: Results on the CMU-MOSEAS-French (CM_{fr}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [24]	LRW	CM_{fr}+MT_{fr}	257	79.9±0.4	79.6
Ours	LRW	CM_{fr}+MT_{fr}	257	**68.4±0.5**	**67.5**
Ours	LRW+LRS2+LRS3	CM_{fr}+MT_{fr}	918	**60.1±0.3**	**59.5**
Ours	LRW+LRS2+LRS3+AVSpeech	CM_{fr}+MT_{fr}	1 559	**59.1±0.5**	**58.3**

S9.5 Results on French

We manually cleaned the French corpus on Multilingual TEDx to exclude videos where the speaker is not visible, resulting in a total of 58 809 videos (84.9 hours) for training, 333 videos (0.4 hours) for validation and 235 videos (0.3 hours) for testing. Results on the Multilingual TEDx-French dataset are shown in Table S15. The proposed approach results in a 9.4 % absolute reduction in the WER. A further reduction of 7.6 % can be achieved by using additional training data.

We divide the French corpus on CMU-MOSEAS [6] into 8 880 videos (15.3 hours) for training and 513 videos (0.8 hours) for testing, respectively. Results on the CMU-MOSEAS-French dataset are shown in Table S16. We observe that our proposed approach results in a 11.5 % absolute reduction in the WER. Furthermore, as expected, the performance is improved by a large margin of 9.3 % when additional training data is included.

References

[1] Chung, J. S. & Zisserman, A. Lip reading in the wild. In *Proceedings of the 13th Asian Conference on Computer Vision*, vol. 10112, 87–103 (2016).

[2] Chung, J. S., Senior, A., Vinyals, O. & Zisserman, A. Lip reading sentences in the wild. In *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3444–3453 (2017).

[3] Afouras, T., Chung, J. S. & Zisserman, A. LRS3-TED: a large-scale dataset for visual speech recognition. Preprint at `https://arxiv.org/abs/1809.00496` (2018).

[4] Zhao, Y., Xu, R. & Song, M. A cascade sequence-to-sequence model for chinese mandarin lip reading. In *Proceedings of the 1st ACM International Conference on Multimedia in Asia*, 1–6 (2019).

[5] Salesky, E. *et al.* The Multilingual TEDx Corpus for Speech Recognition and Translation. In *Proceedings of the 22nd Annual Conference of International Speech Communication Association*, 3655–3659 (2021).

[6] Zadeh, A. B. *et al.* CMU-MOSEAS: A multimodal language dataset for spanish, portuguese, german and french. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 1801–1812 (2020).

[7] Ephrat, A. *et al.* Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. *ACM Transactions on Graphics* **37**, 112:1–112:11 (2018).

[8] Shillingford, B. *et al.* Large-scale visual speech recognition. In *Proceedings of the 20th Annual Conference of International Speech Communication Association*, 4135–4139 (2019).

[9] Makino, T. *et al.* Recurrent neural network transducer for audio-visual speech recognition. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 905–912 (2019).

[10] Serdyuk, D., Braga, O. & Siohan, O. Audio-visual speech recognition is worth 32×32×8 voxels. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 796–802 (2021).

[11] Afouras, T., Chung, J. S. & Zisserman, A. ASR is all you need: Cross-modal distillation for lip reading. In *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing*, 2143–2147 (2020).

[12] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the 29th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–778 (2016).

[13] Stafylakis, T. & Tzimiropoulos, G. Combining residual networks with LSTMs for lipreading. In *Proceedings of the 18th Annual Conference of International Speech Communication Association*, vol. 9, 3652–3656 (2017).

[14] Dai, Z. *et al.* Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 2978–2988 (2019).

[15] Dauphin, Y. N., Fan, A., Auli, M. & Grangier, D. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 933–941 (2017).

[16] Gulati, A. *et al.* Conformer: Convolution-augmented transformer for speech recognition. In *Proceedings of the 21st Annual Conference of International Speech Communication Association*, 5036–5040 (2020).

[17] Vaswani, A. *et al.* Attention is all you need. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 5998–6008 (2017).

[18] Irie, K., Zeyer, A., Schlüter, R. & Ney, H. Language modeling with deep transformers. In *Proceedings of the 20th Annual Conference of International Speech Communication Association*, 3905–3909 (2019).

[19] Panayotov, V., Chen, G., Povey, D. & Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In *Proceedings of the 40th IEEE International Conference on Acoustics, Speech and Signal Processing*, 5206–5210 (2015).

[20] Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N. A. & Estève, Y. TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *International Conference on Speech and Computer*, vol. 11096, 198–208 (2018).

[21] Ardila, R. *et al.* Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, 4218–4222 (2020).

[22] Pratap, V., Xu, Q., Sriram, A., Synnaeve, G. & Collobert, R. MLS: A large-scale multilingual dataset for speech research. In *Proceedings of the 21st Annual Conference of International Speech Communication Association*, 2757–2761 (2020).

[23] Watanabe, S., Hori, T., Kim, S., Hershey, J. R. & Hayashi, T. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing* **11**, 1240–1253 (2017).

[24] Ma, P., Petridis, S. & Pantic, M. End-to-end audio-visual speech recognition with conformers. In *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing*, 7613–7617 (2021).