

AV-DATA2VEC: SELF-SUPERVISED LEARNING OF AUDIO-VISUAL SPEECH REPRESENTATIONS WITH CONTEXTUALIZED TARGET REPRESENTATIONS

Jiachen Lian¹, Alexei Baevski^{2*}, Wei-Ning Hsu^{3*}, Michael Auli³

¹ UC Berkeley ² Character.AI ³ FAIR, Meta

ABSTRACT

Self-supervision has shown great potential for audio-visual speech recognition by vastly reducing the amount of labeled data required to build good systems. However, existing methods are either not entirely end-to-end or do not train joint representations of both modalities. In this paper, we introduce AV-data2vec which addresses these challenges and builds audio-visual representations based on predicting contextualized representations which has been successful in the uni-modal case. The model uses a shared transformer encoder for both audio and video and can combine both modalities to improve speech recognition. Results on LRS3 show that AV-data2vec consistently outperforms existing methods under all settings with the same amount of data and model size.

1. INTRODUCTION

Both human speech production and perception are multimodal, producing acoustic and visual artifacts [1, 2]. Learning *audio-visual speech representations* helps to improve the robustness and accuracy of speech recognition in both noisy and clean settings [3, 4].

The state-of-the-art visual speech recognition (VSR) system relies on about 90K hours of transcribed training data [5]. However, annotating such large amounts of data for every language is simply infeasible which sparked large interest to learn from unlabeled data. AV-HuBERT [3] was the first self-supervised system to jointly learn speech representations from raw audio and video using masked-prediction. However, the training is not entirely end-to-end since the algorithm alternates between representation learning and creating targets using offline clustering. More recently, RAVen [6] introduced an end-to-end algorithm similar to data2vec [7] which trains separate encoder models for audio and visual data. However, separate encoders increase the number of model parameters, and their disjoint model design is also contradictory to the common understanding of the human perception system which is believed to fuse audio and vision early on [8]. Moreover, they do not push the limit of AVSR which tends to perform better than ASR [4, 9].

In this paper, we introduce AV-data2vec (Audio-Visual data2vec) to address these issues by extending data2vec [7] from the unimodal case to learn joint audio-visual representations (Figure 1). AV-data2vec encodes masked audio-visual data and performs a masked prediction task of contextualized targets based on the unmasked input data. Compared to prior work, training is fully end-to-end and there is a single encoder for both audio and vision that can be used to perform AVSR. Another difference to RAVen [6] is that target representations include features of varying granularity which is achieved by averaging the outputs of multiple layers instead of only predicting high-level features produced by the final layer. This enables a learning task over both low-level and high-level features.

* Equal Advising. Work done at Meta AI

AV-data2vec unifies ASR, VSR and AVSR within a single framework and achieves state-of-the-art performance under all settings with the same amount of data/model size.

2. RELATED WORK

2.1. Self-supervised Speech Representation Learning.

There has been much recent research on self-supervised speech representation learning which includes approaches that reconstruct a corrupted or incomplete form of the input using auto-encoding [10], auto-regressive based methods such as [11–13], and masked prediction based methods [14, 15]. There is also work on predicting the frame-wise targets outside of the model computational graph [16–18]. Related to the current paper is [7, 19] who directly regress contextualized targets created by a teacher model.

2.2. Speech Recognition With Visual Cues.

Visual-oriented speech recognition involves the task of *visual speech recognition* (VSR, also known as lip reading) and *audio-visual speech recognition* (AVSR). Earlier work [5, 20–26] started to train with transcribed video/audio-video data in a supervised manner. However, this required large amounts of labeled data of up to 90K hours [5]. There are some semi-supervised methods [23, 27] which significantly reduce the amount of labeled data, however, the performance is still far lower. Most recent advances in self-supervised audio-visual learning [3, 4, 6, 9] are not only more data-efficient but also achieve comparable or better speech recognition results. AV-HuBERT [3] is the first method that jointly learns the modality-agnostic speech representation from raw audio and video. u-HuBERT [9] generalizes AV-HuBERT to utilize both multimodal and unimodal data that is richer in the wild during pretraining. VATLM [28] extends AV-HuBERT by adding auxiliary speech-text tasks which use additional out-of-domain text and speech data. One problem with these approaches is that multi-stage iterative training with offline clustered labels is not end-to-end. RAVen [6] uses a student-teacher paradigm and is end-to-end, however, it uses separate encoders for each modality. This is less parameter-efficient and very different to the human speech perception mechanism [8].

3. METHOD

3.1. Background: data2vec

data2vec [7, 19] is a self-supervised framework that learns the representations from *contextualized* targets via masked prediction. Specifically, a student model encodes a masked version of the training example to predict a contextualized target representation encoded by a teacher model which is based on the unmasked version

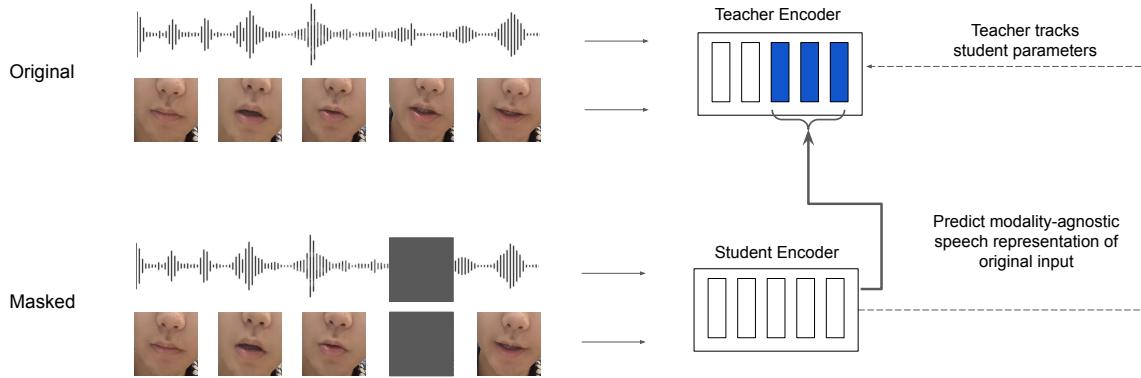


Fig. 1: AV-data2vec jointly encodes both audio and visual data to build audio-visual representations. The student model encodes a masked version of both audio and visual data and predicts a contextualized target representation created by a teacher model which is based on the unmasked version of the training sample. Target representations encode both high-level and low-level features from multiple layers of the teacher model.

of the sample. The teacher model weights are an exponentially moving average (EMA) of the student model weights. The original data2vec framework is designed for single-modality training.

3.2. Audio-Visual data2vec

We extend data2vec to multiple modalities and focus on speech and video inputs to create joint audio-visual representations (Fig. 1). Similar to data2vec, AV-data2vec has a *student encoder* and a *teacher encoder*, however, instead of processing a single modality, encoders can represent both audio and visual data. Both the student and teacher networks are composed of an audio encoder A , a video encoder V , a audio-visual fusion module F and a transformer encoder T .

Audio Encoder. Similar to [3], we encode the audio signal as log filterbanks. We then adopt a dense layer as audio encoder A that maps the U -frame log filterbank energy $X_A = [x_1, x_2, \dots, x_U]$ to acoustic features $M_A = [M_1, M_2, \dots, M_U] \in \mathbb{R}^{U \times D}$ of the same length: $M_A = A(X_A)$. The feature dimension D of the audio encoder matches the input dimension of the transformer encoder. The audio feature M_A is normalized per frame statistics for both pre-training and finetuning [3].

Video Encoder. We use the same video encoder V as AV-HuBERT which is a variant of ResNet-18 [3, 6, 23, 28] that replaces the first 2D convolutional layer [29] by a 3D convolutional layer with a kernel size [5, 7, 7] [30], followed by a batchnorm 3D layer [31], a PReLU layer [32] and a MaxPooling 3D layer with kernel size [1, 3, 3] and strides [1, 2, 2]. The visual features are then reshaped in order to be input to the subsequent 16-layer 2D convolutional layers [29]. An adaptive average pooling 2D layer is applied in the end to output a 1D tensor for each frame. Given a U -frame raw video signal $X_V = [x_1, x_2, \dots, x_U] \in \mathbb{R}^{U \times C \times H \times W}$, the visual encoder V maps X_V to 1D visual features $M_V = [M_1, M_2, \dots, M_U] \in \mathbb{R}^{U \times D}$: $M_V = V(X_V)$. Both the dimension D and number of frames T of visual encoder are the same as those of audio encoder. C , W , and H denote channel, weight and height of each video frame.

Audio-Visual Fusion. AV-data2vec accepts inputs that are either audio-only (a), video-only (v), or audio-video (av) for both student and teacher models. This leads to nine possible training tasks.¹ This compares to four learning tasks for RAVen [6] whose encoders can only encode a single modality each and which lacks the ability to jointly encode modalities. AV-HuBERT [3] can jointly encode modalities and uses *modality dropout* to randomly select the type of input. In initial experiments, we found it very beneficial to adjust the rate at which each input type is selected over time during training.

In this work, we propose a new modality scheduler that coordinates the nine different training tasks. We define the following parameters: p_A , p_V and p_{AV} , denoting the probability that audio/video/audio-video is selected as input modality respectively for either the student or the teacher.²

We designed a modality dropout scheduler for the student model where the rate at which modalities are dropout change over the time. The probabilities p_{AV} , $p_{V|AV}$ and $p_{A|AV}$ are annealed: given a starting and an ending value for a probability, we linearly anneal the probability over M_{anneal} steps. This results in p_A and p_V to be quadratically annealed over M_{anneal} steps.

The audio-visual fusion module is summarized in Eq. 1. Note that there are two independent audio-visual fusion modules for both the student model and the teacher model.

$$M = \begin{cases} M_A + M_V & \text{with probability } p_{AV} \\ M_A + \mathbf{0} & \text{with probability } p_A \\ M_V + \mathbf{0} & \text{with probability } p_V \end{cases} \quad (1)$$

If the input for both student and teacher model is audio-only data, then this is the same as data2vec [7] framework for audio (A-data2vec; Sec.5.4). See supplemental material for a better understanding as well as more details for modality scheduler.

Masking. Following [17, 33], we apply span masking on fused audio-visual features $M = [M_1, M_2, \dots, M_U] \in \mathbb{R}^{U \times D}$. We ran-

¹ $v \rightarrow a$, $av \rightarrow v$, $a \rightarrow a$, $v \rightarrow v$, $av \rightarrow v$, $a \rightarrow v$, $v \rightarrow av$, $av \rightarrow av$, $a \rightarrow av$, where \rightarrow denotes student-to-teacher prediction.

²In the actual implementation, either audio or video is selected conditioned on audio-video not being selected. More precisely: $p_A = p_{\bar{A}V} p_{A|\bar{A}V}$ and $p_V = p_{\bar{A}V} p_{V|\bar{A}V}$, where $p_{\bar{A}V} = 1 - p_{AV}$

domly select $r\%$ timesteps as starting indices to mask spans of length l . Note that if $M = M_A + M_V$, the masking is synchronously applied at the same time step for both audio and video, as illustrated in Fig.1.

Transformer Encoder. The transformer encoder T takes the masked fused audio-visual features \tilde{M} as input (cf. Eq. 1) and outputs the high-level speech representation $Z = T(\tilde{M}) = [z_1, z_2, \dots, z_U] \in \mathbb{R}^{U \times D}$.

3.3. Pretraining Objective

Targets. Similar to [7], AV-data2vec predicts contextualized targets encoding a time-step as well as information about the entire input. Targets are extracted from the representations encoded by the teacher encoder that takes the unmasked features as input. Following [7], we use the output of the FFN prior to the last residual connection in each block as target representation which is denoted as $\bar{Z} \in \mathbb{R}^{U \times D}$. We furthermore denote the target representation at the last k layer as $\bar{Z}^{(N-k+1)} \in \mathbb{R}^{U \times D}$, where N is the total number of transformer blocks, and k is the current block. We then average these representations over the last K blocks and apply instance normalization similar to [7] to derive the targets $Y = \text{IN}(\sum_{k=1}^K \bar{Z}^{(N-k+1)})$, where IN denotes instance normalization.

Loss. Denote outputs of Transformer encoder $Z = [z_1, z_2, \dots, z_U] \in \mathbb{R}^{U \times D}$ and contextualized targets $Y = [y_1, y_2, \dots, y_U] \in \mathbb{R}^{U \times D}$. We consider computing our loss for both masked time-steps and unmasked time-steps [6], depending on the input modality. Empirically we find that audio-only targets perform best (See supplemental materials) and in this setting we found it useful to predict audio targets when we have visual-only inputs even for unmasked time-steps. Whenever we have video as input, then we only predict targets for unmasked time-steps as the task is otherwise trivial. Specifically, if t is the frame index, I the set of masked indices, α and β are two weighting factors, then the loss is:

$$L_{\text{pretrain}} = \alpha \sum_{t \in I} \|z_t - y_t\|_2^2 + \beta \sum_{t \notin I} \|z_t - y_t\|_2^2 \quad (2)$$

Teacher Parameterization Given student encoder weights θ_S , the teacher weights θ_T are an exponentially moving average (EMA) similar to [7]:

$$\theta_T \leftarrow \tau \theta_T + (1 - \tau) \theta_S$$

where τ is a momentum parameter that is linearly increased over time $\tau^s \xrightarrow{\tau_{\text{anneal}}} \tau^e$, where τ^s , τ^e and τ_{anneal} denote the initial value, the ending value of EMA decay, and the EMA decay annealing steps.

3.4. Finetuning Objective

After pretraining, we initialize the encoder of an attention-based sequence-to-sequence (S2S) architecture [34] and finetune it on labeled data. We denote the text targets as $W = [W_1, W_2, \dots, W_S]$ for the current input representation $Z = [Z_1, Z_2, \dots, Z_T]$. We minimize a cross-entropy (CE) criterion: $L_{\text{S2S}} = -\sum_{t=1}^S \log(W_t | W_{<t}, Z)$

4. EXPERIMENTAL SETUP

4.1. Datasets and Preprocessing

LRS3 [35] is the largest publicly available labeled dataset for audio-visual speech recognition in English. It is split as follows: *pretrain* (403h), *trainval* (30h) and *test* (1h). We follow [3] to randomly select about 1h of data from *trainval* for validation.

Voxceleb2 [36] is a multilingual audio-visual dataset for speaker recognition without transcriptions. The original corpus contains more than 2442 hours of videos. We use the English-only part selected by [3] (1326 hours of videos).

Preprocessing. For audio feature extraction, we follow [3] and extract the 26-dimensional log filterbank energy with a stride of 10 ms from raw audio waveform. The original video track has a resolution of 224×224 with a frame rate of 25 fps. Following [3], we use dlib [37] to extract 68 facial key points for each video clip. We then crop a 96×96 region centered on the speakers mouth. During training, we randomly crop a 88×88 region from the whole region and flip it horizontally with probability 0.5. Following [3], we only take grayscale images. During testing, we use the 88×88 region centered on the mouth and no flipping is applied. The frame rate for both modality is 25 fps. As the original audio features have a frame rate of 100 fps, we stack them for 4 audio features.

4.2. Setup and Implementation Details

We consider two experimental setups in terms of amount of labeled data: *low-resource* and *high-resource*. We pretrain AV-data2vec with either LRS3 (433h) or English-only Voxceleb2 + LRS3 (1759h). In the low-resource setting, the model is finetuned on LRS3 *trainval* (30h) only and in the high-resource setting, the model is finetuned on the entire LRS3 training data (433h). Our methods are implemented in fairseq [38].

Hyper-parameters Tuning The performance of AV-data2vec is sensitive to hyper-parameters such as how many blocks to average for the target representations, settings for the modality scheduler, EMA scheduler as well as batch size and learning rate.

Pretraining. Following [3, 28, 39], there are two options for transformer encoder: Base and Large. The number of blocks/embedding dimension/feed-forward dimension/attention heads in each transformer block are 12/768/3072/12 and 24/1024/4096/16 for Base and Large respectively. For masking, we set mask probability $r\% = 50\%$ and span length $l = 10$. For pretraining loss defined in Eq.2, we set $\alpha = 1$ and $\beta = 0$ if the input modality is audio-only or audio-video, and if the input modality is video-only, we set $\alpha = 1$ and $\beta = 1$. For student modality scheduler, we set $p_{AV} : 1 \xrightarrow{150k} 0.25, p_{V|\bar{AV}} : 1 \xrightarrow{150k} 1, p_{A|\bar{AV}} : 0 \xrightarrow{150k} 0$.

For teacher modality scheduler, we set the input as audio-only. We fixed these modality schedulers for all pretraining experiments. For BASE model with 433h pretraining, we set lr=5e-4, $\tau^s = 0.999$, $\tau^e = 0.99999$, $\tau_{\text{anneal}} = 100k$. The batch size is 20s per GPU. We set total number of updates as 1000k and the model is trained on 64 V100 for 4-5 days. For Base model with 1759h pretraining, we use almost the same settings with the exception that we double the effective batch size and train it for 2000k updates (8-10 days). For Large model with 433h pretraining, we still use the same settings

Table 1: Low-labeled Data Results. We pretrain AV-data2vec Large/Base with 433h/1759h of unlabeled data, and finetune on 30h of labeled data. The results of visual speech recognition (VSR), automatic speech recognition (ASR) and audio-visual speech recognition (AVSR) are shown. CE denotes cross-entropy, also applying to Table. 3. AV-data2vec achieves state-of-the-art results in all settings with same amount of data/model size.

Methods	Unlabeled AV data	Labeled Data	Encoder Size	Criterion	VSR	ASR	AVSR
<i>Self-supervised (Base Models)</i>							
AV-HuBERT [3]	433h	30h	103M	CE	51.8	4.9	4.7 ²
RAVen [6]	433h	30h	97M	CTC+CE	47.0	4.7	-
VATLM [28]	433h ¹	30h	103M	CE	48.0	-	3.6
AV-data2vec	433h	30h	103M	CE	45.2	4.4	4.2
AV-HuBERT [3, 4]	1759h	30h	103M	CE	46.1	4.6	4.0
RAVen [6]	1759h	30h	97M	CTC+CE	40.2	3.8	-
VATLM [28]	1759h ¹	30h	103M	CE	42.6	-	3.4
AV-data2vec	1759h	30h	103M	CE	37.8	3.7	3.3
<i>Self-supervised (Large Models)</i>							
AV-HuBERT [3]	433h	30h	325M	CE	44.8	4.5	4.2 ²
AV-data2vec	433h	30h	325M	CE	40.5	3.7	3.4
AV-HuBERT [3, 4]	1759h	30h	325M	CE	32.5	2.9	3.3
RAVen [6]	1759h	30h	671M	CTC+CE	33.1	2.6	-
VATLM [28]	1759h ¹	30h	325M	CE	31.6	-	2.7
AV-data2vec	1759h	30h	325M	CE	30.8	2.7	2.7

¹ VATLM uses additional 3846h audio, 452h audio-text and 600M text data

² We reproduced AV-HuBERT and report our AVSR results.

as Base model with the exception that we double the effective batch size and lr=2e-4. It takes around 6-7 days to finish. For Large model with 1759h pretraining, we use the same settings as Base model with 1759h pretraining with the exception that we set lr=2e-4. It takes around 10-12 days to finish training.

Finetuning. We consider two transformer decoders: Base and Large. The number of blocks/embedding dimension/feed-forward dimension/attention heads in each transformer block are 6/768/30 72/4 and 9/1024/4096/8 for Base and Large respectively. We use subword [40] for S2S targets. For ASR/VSR finetuning, the video or audio features are set as zero vectors respectively. For AVSR finetuning, both video and audio are taken as input and there is no modality dropout. For ASR finetuning, we use tri-stage learning rate scheduler and freeze the encoder for some steps [3]. The learning rate/total number of updates/warmup steps for 30h/433h are 1e-3/1e-3, 40k/60k, 10k/20k, 24k/48k respectively. Settings are the same for both Base and Large model.

For VSR finetuning, we use cosine learning rate scheduler and freeze the encoder for some steps [3]. The learning rate/total number of updates/warmup steps for 30h/433h are 1e-3/1e-3, 40k/120k, 2k/20k, 24k/48k respectively. Settings are the same for both Base and Large model. For AVSR finetuning, we use tri-stage learning rate scheduler and freeze the encoder for some steps [3]. The learning rate/total number of updates/warmup steps for 30h/433h are 1e-3/1e-3, 40k/60k, 10k/20k, 24k/48k respectively. Settings are the same for both Base and Large model. Since the AVSR results are not reported in [3] and are partially reported in [4], we reproduced AV-HuBERT and report our own AVSR results for the remaining settings, shown in Table.1 and Table.3.

Decoding. We tune the beam width in {5, 10, 25, 50, 100} and report the best number. We do not apply LM for decoding. For VSR, ASR and AVSR tasks, the input modalities are video-only, audio-only and audio-video respectively in both finetuning and decoding.

5. RESULTS

5.1. Low-labeled Data Setup

We first consider a low-labeled data setup using 30h of data for finetuning whose results are shown in Table. 1. For Base models, AV-data2vec consistently outperforms existing methods for both VSR and ASR. On AVSR, AV-data2vec achieves the best result except for the 433h pretraining setting, where AV-data2vec achieves 4.2 compared to 3.6 for VATLM. However, VATLM [28] uses additional audio and text data for their auxiliary pretraining tasks. AV-data2vec appears to benefit more from an increased amount of pretraining data (1759h vs. 433h) than other approaches.

For Large models, AV-data2vec achieves the best results except for the 1759h setting, where AV-data2vec gets 2.7 while RAVen gets 2.6. We attribute this in part due to RAVen having about double the model size due to their two encoder architecture. Overall, with the same amount of pretraining data, larger models result in better performance. However, the benefits of increased model capacity and more pretraining data begin to diminish as can be seen in the results of the largest setting (Large model, 1759h pretraining data).

5.2. High-labeled Data Setup

Results of the high-labeled data setting (433h) are shown in Table. 3. AV-data2vec achieves state-of-the-art VSR/ASR/AVSR results except for the largest setting (Large model, 1759h pretraining data): u-HuBERT [9] achieves the best VSR performance of 27.2, however, it uses an additional 452h of data for pretraining. VATLM [28] and u-HuBERT achieve the best AVSR results, however, VATLM uses additional 3846h audio, 452h audio-text and 600M text data, which gives it an advantage. In summary, AV-data2vec still achieves best results with the same amount of data/model size.

5.3. Comparison to RAVen

Similar to AV-data2vec, RAVen [6] also uses contextualized and continuous targets, however, it differs from AV-data2vec in several important aspects. RAVen does not create joint modality embeddings

Table 2: High-labeled Data Results. We pretrain Base/Large models with 433h/1759h of unlabeled data and finetune on 433h of labeled data. Results of supervised/semi-supervised work are also included. AV-data2vec achieves state-of-the-art results under most settings.

Methods	Unlabeled AV data	Labeled Data	Backbone	Encoder Size	Criterion	VSR	ASR	AVSR
<i>Supervised</i>								
Afouras et al. 2018 [20]	-	1519h	Transformer	-	CE	58.9	8.3	-
Xu et al. 2020 [21]	-	590h	RNN	-	CE	57.8	7.2	-
Shillingford et al. 2018 [22]	-	3886h	RNN	-	CTC	55.1	-	-
Ma et al. 2022 [23]	-	813h	Conformer	-	CTC+CE	34.7	-	-
Makino et al. 2019 [24]	-	31000	RNN	-	Transducer	33.6	4.8	4.5
Prajwal et al. 2022 [25]	-	2676h	Transformer	-	CE	30.7	-	-
Serdyuk et al. 2021 [26]	-	90000h	Transformer	-	Transducer	25.9	-	2.3
Serdyuk et al. 2022 [5]	-	90000h	Conformer	-	Transducer	17.0	-	1.6
<i>Semi-Supervised</i>								
Afouras et al. 2020 [27]	344h	433h	Jasper(CNN)	-	CTC+CE	59.8	-	-
Ma et al. 2022 [23]	641h	818h	Conformer	-	CTC+CE	31.5	-	-
<i>Self-supervised (Base Models)</i>								
AV-HuBERT [3]	433h	433h	Transformer	103M	CE	44.0	3.0	2.8 ²
RAVen [6]	433h	433h	Transformer	97M	CTC+CE	39.1	2.2	-
AV-data2vec	433h	433h	Transformer	103M	CE	39.0	2.0	1.8
AV-HuBERT [3]	1759h	433h	Transformer	103M	CE	34.8	2.0	1.8 ³
RAVen [6]	1759h	433h	Transformer	97M	CTC+CE	33.1	1.9	-
VATLM [28]	1759h ¹	433h	Transformer	103M	CE	34.2	-	1.7
AV-data2vec	1759h	433h	Transformer	103M	CE	32.9	1.7	1.4
<i>Self-supervised (Large Models)</i>								
AV-HuBERT [3]	433h	433h	Transformer	325M	CE	41.6	2.7	2.5 ²
AV-data2vec	433h	433h	Transformer	325M	CE	37.4	1.9	1.7
AV-HuBERT [3, 4]	1759h	433h	Transformer	325M	CE	28.6	1.3	1.4
RAVen [6]	1759h	433h	Transformer	671M	CTC+CE	28.2	1.4	-
VATLM [28]	1759h ¹	433h	Transformer	325M	CE	28.4	-	1.2
u-HuBERT [9]	1759h ¹	433h	Transformer	325M	CE	27.2	1.4	1.2
AV-data2vec	1759h	433h	Transformer	325M	CE	28.5	1.3	1.3

¹ VATLM uses additional 3846h audio, 452h audio-text and 600M text data, and u-HuBERT uses additional 452h audio data.

² We reproduced AV-HuBERT to report corresponding AVSR results.

and is not able to perform AVSR. Also, RAVen has different encoders for audio and video. For the Base model, each of the two encoders is half the size of AV-data2vec but collectively they have similar size. For the Large model, each RAVen encoder is the same size as AV-data2vec and thus the total size of RAVen in the Large size is about double of AV-data2vec. Next, the finetuning criterion for RAVen is joint CTC-Attention [41] while AV-data2vec adopts a sequence to sequence architecture inline with AV-HuBERT [3] and VATLM [28]. Finally, AV-data2vec empirically performs better as our results show.

5.4. Joint-modality vs. Audio-only Pretraining

Next, we compare joint audio-visual self-supervised learning to audio-only self-supervised learning. To do so, we pretrain an audio-only version of our model (A-data2vec), by simply removing visual features before they are fed to the transformer encoder; we do not use modality dropout. We train A-data2vec for 600K updates for all settings and adopt the same finetuning/decoding configurations as AV-data2vec. The ASR results (Figure 3) show that joint audio-visual pretraining outperforms audio-only pretraining in almost all settings. In the largest high-resource setting (Large, 1759h unlabeled data, 433h labeled data), performance saturates and the difference to audio-only pretraining is very small.

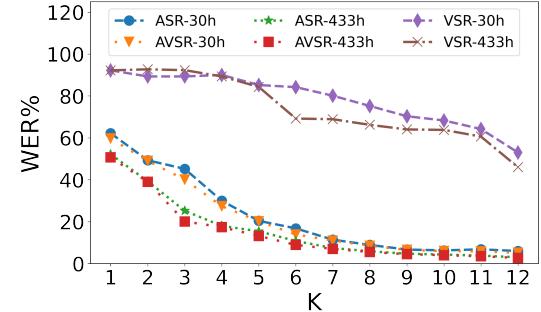


Fig. 2: Effect of averaging K blocks to create contextualized target representations. More blocks improve performance because targets become richer due to including both high-level and low-level features. Results are based on a Base model pretrained on 433h of unlabeled data and finetuned on 30h of labeled data.

5.5. Ablation1: Top-K target averaging

We first measure the impact of creating contextualized target representations based on multiple blocks ranging from the top block to the 12 blocks. For this experiment, we fix $p_{AV} = 0.5$, $p_A = p_V = 0.25$ for the student encoder which is the default schedule of [3] we set $p_A = 1$, $p_{AV} = p_V = 0$ for the teacher encoder as video contains more ambiguous information as targets, as mentioned in [6]. For

Table 3: High-labeled Data Results. We pretrain Base/Large models with 433h/1759h of unlabeled data and finetune on 433h of labeled data. Results of supervised/semi-supervised work are also included. AV-data2vec achieves state-of-the-art results under most settings.

Methods	Unlabeled AV data	Labeled Data	Backbone	Encoder Size	Criterion	VSR	ASR	AVSR
<i>Supervised</i>								
Afouras et al. 2018	-	1519h	Transformer	-	CE	58.9	8.3	-
Xu et al. 2020	-	590h	RNN	-	CE	57.8	7.2	-
Shillingford et al. 2018	-	3886h	RNN	-	CTC	55.1	-	-
Ma et al. 2022	-	813h	Conformer	-	CTC+CE	34.7	-	-
Makino et al. 2019	-	31000	RNN	-	Transducer	33.6	4.8	4.5
Prajwal et al. 2022	-	2676h	Transformer	-	CE	30.7	-	-
Serdyuk et al. 2021	-	90000h	Transformer	-	Transducer	25.9	-	2.3
Serdyuk et al. 2022	-	90000h	Conformer	-	Transducer	17.0	-	1.6
<i>Semi-Supervised</i>								
Afouras et al. 2020	344h	433h	Jasper(CNN)	-	CTC+CE	59.8	-	-
Ma et al. 2022	641h	818h	Conformer	-	CTC+CE	31.5	-	-
<i>Self-supervised (Base Models)</i>								
AV-HuBERT	433h	433h	Transformer	103M	CE	44.0	3.0	2.8 ²
RAVEn	433h	433h	Transformer	97M	CTC+CE	39.1	2.2	-
AV-data2vec	433h	433h	Transformer	103M	CE	39.0	2.0	1.8
AV-HuBERT	1759h	433h	Transformer	103M	CE	34.8	2.0	1.8 ³
RAVEn	1759h	433h	Transformer	97M	CTC+CE	33.1	1.9	-
VATLM	1759h ¹	433h	Transformer	103M	CE	34.2	-	1.7
AV-data2vec	1759h	433h	Transformer	103M	CE	32.9	1.7	1.4
<i>Self-supervised (Large Models)</i>								
AV-HuBERT	433h	433h	Transformer	325M	CE	41.6	2.7	2.5 ²
AV-data2vec	433h	433h	Transformer	325M	CE	37.4	1.9	1.7
AV-HuBERT	1759h	433h	Transformer	325M	CE	28.6	1.3	1.4
RAVEn	1759h	433h	Transformer	671M	CTC+CE	28.2	1.4	-
VATLM	1759h ¹	433h	Transformer	325M	CE	28.4	-	1.2
u-HuBERT	1759h ¹	433h	Transformer	325M	CE	27.2	1.4	1.2
AV-data2vec	1759h	433h	Transformer	325M	CE	28.5	1.3	1.3

¹ VATLM uses additional 3846h audio, 452h audio-text and 600M text data, and u-HuBERT uses additional 452h audio data.

² We reproduced AV-HuBERT to report corresponding AVSR results.

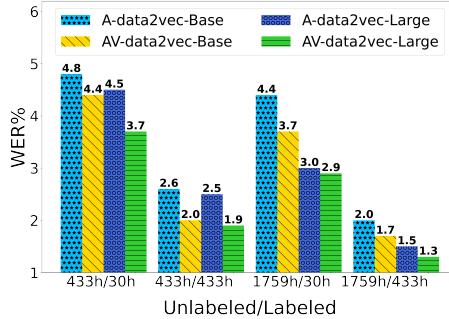


Fig. 3: AV-data2vec performs better than audio-only training (A-data2vec) in all ASR settings.

EMA, we set $\tau^s = 0.999$, $\tau^e = 0.99999$ and $\tau_{anneal} = 100k$. Fig.2 shows that averaging more blocks improves performance, inline with prior experiments for ASR, image recognition and natural language understanding [7]. We therefore generally use $K = 12$ for Base models and $K = 24$ for Large models.

5.6. Ablation2: Scaling

For Large models and the largest unlabeled data setting (1759h), we investigate the effect of batch size and learning rates. Table.4 shows

the performance of a few settings we explored: For 433h pretraining with Base model settings, increasing the batch size leads to plateauing performance. However, when the amount of pretraining data is increased to 1759h, larger batch size still leads to better performance for all tasks.

For the Large model with 433h of unlabeled data, we found that smaller learning rates ($< 5e-4$) improve performance; we find that $2e-4$ gives the best performance. When increasing the amount of pre-training data to 1759h, the largest batch size we considered (2560s) with learning rate $2e-4$ performs very well.

Configuration	30h Labeled Data			433h Labeled Data			
	unlabeled	bsz	lr	model	VSR	ASR	AVSR
433h	640s	5e-4		BASE	48.7	4.9	4.7
433h	1280s	5e-4		BASE	45.2	4.4	4.2
433h	2560s	5e-4		BASE	45.3	4.5	4.3
1759h	640s	5e-4		BASE	52.2	4.9	4.6
1759h	1280s	5e-4		BASE	44.2	4.2	4.0
1759h	2560s	5e-4		BASE	37.8	3.7	3.3
433h	1280s	5e-4		BASE	45.5	4.3	4.1
433h	1280s	3e-4		LARGE	43.7	4.0	3.8
433h	1280s	2e-4		LARGE	40.5	3.7	3.4
433h	1280s	1e-4		LARGE	41.2	3.9	3.8
1759h	2560s	2e-4		LARGE	30.8	2.7	2.7

Table 4: Ablation of batch size and learning rates for Base and Large models. bsz denotes batch size. Large models benefit more from smaller learning rates and larger amounts of unlabeled data benefits more from larger batch size.

6. CONCLUSION AND LIMITATIONS

We proposed AV-data2vec, a self-supervised framework to jointly learn audio-visual speech representations based on contextualized targets. AV-data2vec adopts a shared modality-agnostic transformer encoder which takes as input both audio and video data, both of which are fused early on, similar to the human speech perception system. AV-data2vec unifies ASR, VSR and AVSR within a single framework and achieves state-of-the-art performance under all settings with the same amount of data/model parameters. Despite of this, there are still several limitations.

Firstly, the current state-of-the-art self-supervised audio-visual speech recognition results are still inferior to supervised systems that rely on approximately 90K hours of labeled data [5]. Nevertheless, the self-supervised results for all of the current methods (AV-HuBERT [3], VATLM [28], RAVen [6], AV-data2vec) tend to reach saturation under high-resource and LARGE model settings. uHuBERT [9] and VATLM attempt to use additional single-modality data to enhance performance, but the gain is limited.

Secondly, our results are sensitive to hyper-parameters, such as modality scheduler. The training process for both data2vec [7] and AV-data2vec is not stable, which means that a good set of hyper-parameters can produce remarkable results. However, the optimal set of hyper-parameters may still be challenging to obtain. We believe that this high sensitivity is due to the fact that video data is much noisier than speech and contains less linguistic information. Furthermore, since our visual feature extractor, i.e., the ResNet-18, may not be capable of extracting sufficient useful information, the fused audio-visual feature may tend to be dominated by audio features. This sensitivity to the modality scheduler has also been observed in AV-HuBERT and RAVen. To address this issue, it would be beneficial to use a more powerful visual feature encoder such as Video Transformer that is adopted in VideoCLIP [42]. Additionally, implementing an information encoding monitoring method would provide better feedback for tuning the modality scheduler. It also worths to explore the audio-visual learning in the articulatory space [43–46] to introduce vocal tract signal as additional signal to supervise the learning.

7. ACKNOWLEDGEMENT

We thank Bernie Huang for fruitful discussions around transformer block normalization schemes. The purpose of this project is foremost to further the state of the art in audio-visual representation learning research.

8. REFERENCES

- [1] Randy L Diehl, Andrew J Lotto, Lori L Holt, et al., “Speech perception,” *Annual review of psychology*, vol. 55, no. 1, pp. 149–179, 2004.
- [2] Charles F Hockett and Charles D Hockett, “The origin of speech,” *Scientific American*, vol. 203, no. 3, pp. 88–97, 1960.
- [3] Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed, “Learning audio-visual speech representation by masked multimodal cluster prediction,” *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.
- [4] Bowen Shi, Wei-Ning Hsu, and Abdelrahman Mohamed, “Robust self-supervised audio-visual speech recognition,” *Interspeech*, 2022.
- [5] Dmitriy Serdyuk, Otavio Braga, and Olivier Siohan, “Transformer-based video front-ends for audio-visual speech recognition,” *Interspeech*, 2022.
- [6] Alexandros Haliassos, Pingchuan Ma, Rodrigo Mira, Stavros Petridis, and Maja Pantic, “Jointly learning visual and auditory speech representations from raw data,” *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.
- [7] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jitao Gu, and Michael Auli, “Data2vec: A general framework for self-supervised learning in speech, vision and language,” *International Conference on Machine Learning (ICML)*, 2022.
- [8] KP Green, “The use of auditory and visual information during phonetic processing: implications for theories of speech perception. campbell r, dodd b, burnham d, editors. hearing by eye ii: advances in the psychology of speechreading and auditory-visual speech,” 1998.
- [9] Wei-Ning Hsu and Bowen Shi, “u-hubert: Unified mixed-modal speech pretraining and zero-shot transfer to unlabeled modality,” in *Advances in Neural Information Processing Systems*, 2022.
- [10] Aaron Van Den Oord, Oriol Vinyals, et al., “Neural discrete representation learning,” *Advances in neural information processing systems*, vol. 30, 2017.
- [11] Yu-An Chung and James Glass, “Generative pre-training for speech with autoregressive predictive coding,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3497–3501.
- [12] Yu-An Chung, Hao Tang, and James Glass, “Vector-quantized autoregressive predictive coding,” *Interspeech*, 2020.
- [13] Shaoshi Ling, Yuzong Liu, Julian Salazar, and Katrin Kirchhoff, “Deep contextualized acoustic representations for semi-supervised speech recognition,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6429–6433.
- [14] Xianghu Yue and Haizhou Li, “Phonetically motivated self-supervised speech representation learning.,” in *Interspeech*, 2021, pp. 746–750.
- [15] Alexander H Liu, Yu-An Chung, and James Glass, “Non-autoregressive predictive coding for learning speech representations from local dependencies,” *Interspeech*, 2021.
- [16] Alexei Baevski, Michael Auli, and Abdelrahman Mohamed, “Effectiveness of self-supervised pre-training for speech recognition,” *arXiv preprint arXiv:1911.03912*, 2019.
- [17] 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,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [18] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei, “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

- [19] Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli, “Efficient self-supervised learning with contextualized target representations for vision, speech and language,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 1416–1429.
- [20] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman, “Deep audio-visual speech recognition,” *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [21] Bo Xu, Cheng Lu, Yandong Guo, and Jacob Wang, “Discriminative multi-modality speech recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14433–14442.
- [22] Brendan Shillingford, Yannis Assael, Matthew W Hoffman, Thomas Paine, Cian Hughes, Utsav Prabhu, Hank Liao, Hasim Sak, Kanishka Rao, Lorrayne Bennett, et al., “Large-scale visual speech recognition,” *Interspeech*, 2019.
- [23] Pingchuan Ma, Stavros Petridis, and Maja Pantic, “Visual speech recognition for multiple languages in the wild,” *Nature Machine Intelligence*, vol. 4, no. 11, pp. 930–939, oct 2022.
- [24] Takaki Makino, Hank Liao, Yannis Assael, Brendan Shillingford, Basilio Garcia, Otavio Braga, and Olivier Siohan, “Recurrent neural network transducer for audio-visual speech recognition,” in *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*. IEEE, 2019, pp. 905–912.
- [25] KR Prajwal, Triantafyllos Afouras, and Andrew Zisserman, “Sub-word level lip reading with visual attention,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- [26] Dmitriy Serdyuk, Otavio Braga, and Olivier Siohan, “Audio-visual speech recognition is worth 32x32x8 voxels,” *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 796–802, 2021.
- [27] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman, “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)*. IEEE, 2020, pp. 2143–2147.
- [28] Qiushi Zhu, Long Zhou, et al., “VatLM: Visual-audio-text pre-training with unified masked prediction for speech representation learning,” *IEEE Transactions on Multimedia*, pp. 1–11, 2023.
- [29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [30] Stavros Petridis, Themos Stafylakis, Pingchuan Ma, Georgios Tzimiropoulos, and Maja Pantic, “Audio-visual speech recognition with a hybrid ctc/attention architecture,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 513–520.
- [31] Sergey Ioffe and Christian Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*. PMLR, 2015, pp. 448–456.
- [32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.
- [33] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12449–12460, 2020.
- [34] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, “End-to-end attention-based large vocabulary speech recognition,” in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.
- [35] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman, “Lrs3-ted: a large-scale dataset for visual speech recognition,” *arXiv preprint arXiv:1809.00496*, 2018.
- [36] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, “Voxceleb2: Deep speaker recognition,” *Interspeech*, 2018.
- [37] Davis E King, “Dlib-ml: A machine learning toolkit,” *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.
- [38] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli, “fairseq: A fast, extensible toolkit for sequence modeling,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, June 2019, pp. 48–53.
- [39] Jacob Devlin, Ming-Wei Chang, et al., “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019, pp. 4171–4186.
- [40] Taku Kudo, “Subword regularization: Improving neural network translation models with multiple subword candidates,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.
- [41] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, “Hybrid ctc/attention architecture for end-to-end speech recognition,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.
- [42] Hu Xu, Gargi Ghosh, et al., “VideoCLIP: Contrastive pre-training for zero-shot video-text understanding,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Nov. 2021, pp. 6787–6800.
- [43] Jiachen Lian, Alan W Black, Louis Goldstein, and Gopala Krishna Anumanchipalli, “Deep Neural Convulsive Matrix Factorization for Articulatory Representation Decomposition,” in *Proc. Interspeech 2022*, 2022, pp. 4686–4690.
- [44] Jiachen Lian, Alan W Black, Yijing Lu, Louis Goldstein, Shinji Watanabe, and Gopala K Anumanchipalli, “Articulatory representation learning via joint factor analysis and neural matrix factorization,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [45] Peter Wu, Li-Wei Chen, Cheol Jun Cho, Shinji Watanabe, Louis Goldstein, Alan W Black, and Gopala K. Anumanchipalli, “Speaker-independent acoustic-to-articulatory speech inversion,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

- [46] Peter Wu, Tingle Li, Yijing Lu, Yubin Zhang, Jiachen Lian, Alan W Black, Louis Goldstein, Shinji Watanabe, and Gopala K. Anumanchipalli, “Deep Speech Synthesis from MRI-Based Articulatory Representations,” in *Proc. INTERSPEECH 2023*, 2023, pp. 5132–5136.