A CONTINUOUS PSEUDO-LABELING DETAILS

Algorithm 1: Audio-Visual Continuous Pseudo-Labeling (AV-SlimIPL)

Data: Labeled videos $L = \{(\boldsymbol{a}_i, \boldsymbol{v}_i), \boldsymbol{y}_i\}$ and unlabeled videos $U = \{(\boldsymbol{a}_j, \boldsymbol{v}_j)\}$
Result: Audio-visual model $\mathcal{M}_{\boldsymbol{\theta}}$
Optional step: Train $\mathcal{M}_{\boldsymbol{\theta}}$ on labeled audio-only data $L' = \{\boldsymbol{a}_i, \boldsymbol{y}_i\}$ and restart the optimizer;
1. Train $\mathcal{M}_{\boldsymbol{\theta}}$ on labeled audio-visual data $L = \{(\boldsymbol{a}_i, \boldsymbol{v}_i), \boldsymbol{y}_i\}$ with modality dropout $p_m = p_a$ until convergence, and restart the optimizer; ▷ `Train the seed model.`
2. Train $\mathcal{M}_{\boldsymbol{\theta}}$ on labeled audio-visual data $L = \{(\boldsymbol{a}_i, \boldsymbol{v}_i), \boldsymbol{y}_i\}$ with modality dropout $p'_m = p'_a$ for M steps ; ▷ `Begin CPL; "warm up" phase with new modality dropout.`
3. **while** *cache is not full at size* C **do**
    - Draw a random batch from $(\boldsymbol{a}, \boldsymbol{v}) \in U$;
    - Generate its PL $\hat{\boldsymbol{y}}$ by $\mathcal{M}_{\boldsymbol{\theta}}(\boldsymbol{a}, \boldsymbol{v})$ with greedy decoding;
    - Store $\{(\boldsymbol{a}, \boldsymbol{v}), \hat{\boldsymbol{y}}\}$ into the cache;
    - Train $\mathcal{M}_{\boldsymbol{\theta}}$ on L with augmentation and modality dropout $p'_m = p'_a$ for 1 update;

end
repeat
    4. Train $\mathcal{M}_{\boldsymbol{\theta}}$ on L with augmentation and modality dropout $p'_m = p'_a$ for N_L updates;
    5. **for** N_U *updates* **do**
        - Draw a random batch $B = \{(\boldsymbol{a}, \boldsymbol{v}), \hat{\boldsymbol{y}}\}$ from the cache;
        - With probability p, B is removed from the cache and replaced by a new unlabeled sample $(\boldsymbol{a}', \boldsymbol{v}') \in U$ and its PL $\hat{\boldsymbol{y}}'$ generated by current model state $\mathcal{M}_{\boldsymbol{\theta}}(\boldsymbol{a}, \boldsymbol{v})$;
        - Train $\mathcal{M}_{\boldsymbol{\theta}}$ on batch B with augmentation and modality dropout $p'_m = p'_a$ for 1 update;

    end
until *convergence*;

Algorithm 2: Audio-Visual Continuous Pseudo-Labeling (AV-EMA-PL)

Data: Labeled videos $L = \{(\boldsymbol{a}_i, \boldsymbol{v}_i), \boldsymbol{y}_i\}$ and unlabeled videos $U = \{(\boldsymbol{a}_j, \boldsymbol{v}_j)\}$
Result: Audio-visual model $\mathcal{M}_{\boldsymbol{\theta}}$
Optional step: Train $\mathcal{M}_{\boldsymbol{\theta}}$ on labeled audio-only data $L' = \{\boldsymbol{a}_i, \boldsymbol{y}_i\}$ and restart the optimizer;
1. Train $\mathcal{M}_{\boldsymbol{\theta}}$ on labeled audio-visual data $L = \{(\boldsymbol{a}_i, \boldsymbol{v}_i), \boldsymbol{y}_i\}$ with modality dropout $p_m = p_a$ until convergence, and restart the optimizer; ▷ `Train the seed model.`
2. Initialize $\mathcal{M}_{\boldsymbol{\phi}} = \mathcal{M}_{\boldsymbol{\theta}}$; ▷ `Copy teacher weights from student.`
for *M steps;* ▷ `Begin CPL; "warm up" phase with new modality dropout.`
do
    - Train $\mathcal{M}_{\boldsymbol{\theta}}$ on labeled audio-visual data $L = \{(\boldsymbol{a}_i, \boldsymbol{v}_i), \boldsymbol{y}_i\}$ w/ modality dropout $p'_m = p'_a$ for 1 step;
    - Update teacher weights: $\boldsymbol{\phi} \leftarrow \alpha\boldsymbol{\phi} + (1 - \alpha)\boldsymbol{\theta}$

end
repeat
    4. Train $\mathcal{M}_{\boldsymbol{\theta}}$ on L with augmentation and modality dropout $p'_m = p'_a$ for N_L updates;
    5. **for** N_U *updates* **do**
        - Draw a random batch $(\boldsymbol{a}', \boldsymbol{v}') \in U$ and generate its PL $\hat{\boldsymbol{y}}'$ by $\mathcal{M}_{\boldsymbol{\phi}}(\boldsymbol{a}, \boldsymbol{v})$ with greedy decoding to form a batch $B = \{(\boldsymbol{a}, \boldsymbol{v}), \hat{\boldsymbol{y}}\}$;
        - Train $\mathcal{M}_{\boldsymbol{\theta}}$ on batch B with augmentation and modality dropout $p'_m = p'_a$ for 1 update;
        - Update teacher weights: $\boldsymbol{\phi} \leftarrow \alpha\boldsymbol{\phi} + (1 - \alpha)\boldsymbol{\theta}$

    end
until *convergence*;

AV-SlimIPL vs AV-EMA-PL. The pseudo-code for AV-SlimIPL and AV-EMA-PL is shown in Algorithm 1 and Algorithm 2 respectively. AV-SlimIPL requires a data structure for maintaining the cache. The fastest method for doing this is to store the unlabeled samples and pseudo-labels in CPU memory. While this is practical for the original SlimIPL (Likhomanenko et al., 2021a) which only trains with audio, this is infeasible for video with cache sizes $C > 100$ due to the large size of the video frames. For example, a 1s spectrogram contains $80 \times 100 = 8,000$ values, while 1s of single-channel video contains $96 \times 96 \times 25 = 230,400$ values. Instead of keeping the samples in memory, a workaround is to maintain a mapping of unlabeled samples IDs in the dataset to their PLs. However, this requires loading the unlabeled data twice for each training step on un-

labeled data: once for pseudo-labeling and once for training on the unlabeled sample[3]. Loading video frames is significantly more time-consuming than loading audio due to the larger file sizes. In comparison, AV-EMA-PL only requires loading the unlabeled data once since the teacher model is used to generate PLs each time the student model is trained on a batch of unlabeled data. This requires keeping two copies of the model parameters, however, we find this to be easier on the memory and CPU thread consumption at the expense of being slightly slower than AV-SlimIPL due to the re-generation of PLs at every iteration. Therefore, we focused on AV-EMA-PL for our AV-CPL experiments.

B Dataset Statistics

Table B1: Dataset statistics for labeled LRS3 and unlabeled VoxCeleb2-English videos.

Dataset	Split	# Samples	Audio duration (s)						
			Mean	Std.	Min.	25%	50%	75%	Max
LRS3	Test	1,321	2.3	1.3	0.6	1.4	1.9	2.8	6.2
LRS3	Validation	1,200	3.5	1.8	0.7	1.9	3.0	5.6	6.2
LRS3, 30h	Train	30,782	3.4	1.8	0.5	1.9	3.0	5.4	6.2
LRS3, 433h	Train	299,646	5.3	3.5	0.3	2.6	4.4	6.9	86.7
VoxCeleb2 1,326h	Train	628,418	7.6	3.8	0.4	4.9	6.3	9.0	22.5

Table B1 shows the number of samples and the length statistics for the sequences in the LRS3 and VoxCeleb2 dataset splits. The VoxCeleb2-English video split is provided by Shi et al. (2022a) according to an off-the-shelf English ASR model. We remove samples longer than 20s to ease the computational complexity.

C Optimization Details

We train all of the models up to 300k-400k steps on 8 A100 GPUs with 80GB of memory. Video samples with similar lengths are batched together such that the maximum number of frames is 5,680 frames (227s) per GPU. We apply SpecAugment (Park et al., 2019) during training to the input spectrograms with the following parameters: two frequency masks with frequency parameter $F = 30$, ten time masks with time mask parameter $T = 50$ and maximum time-mask ratio $p = 0.1$. When fine-tuning on the LRS3 30h training set, we use the same parameters except reduce the number of time masks to two since the videos are shorter on average. We use the AdaGrad optimizer (Duchi et al., 2011) with a learning rate of 0.03. The learning rate is warmed up for 64k steps and then held constant at 0.03 until 200k steps were reached. Then the learning rate is reduced by 2 every 50k updates if the WER does not improve on the validation set. Dropout and layer drop (Fan et al., 2020) are set to 0.1 during supervised training and CPL. Gradients are clipped to a maximum norm of 1.0. For AV-CPL and V-CPL experiments, we use $M = 5k$ warmup steps and $\alpha = 0.9999$. For the audio-only SlimIPL experiments, we use a cache size of 500 and $M = 20k$ warmup steps. Our implementation is in Jax (Bradbury et al., 2018).

D Language Model Inference

We train a 4-gram word-level language model on the LRS3 text using KenLM (Heafield, 2011). We use the Flashlight beam-search decoder (Kahn et al., 2022) implemented in Torchaudio (Yang et al., 2022) to integrate the language model. The perplexity on the LRS3 test set using the language model trained on the 433h training set was 92.5 excluding Out-of-Vocabularies (OOV) and 94.0 including OOV. The perplexity on the LRS3 test set using the language model trained on the 30h training set was 112.2 excluding OOV and 122.4 including OOV. We use the LRS3 text to construct a lexicon file which contains 51,292 words. We tuned the LM weight among $\{0, 1, 2, 4, 8\}$ and word insertion penalty among $\{\pm 4, \pm 2, \pm 1, 0\}$ using grid search on the validation set and selected the LM weight of 2 and word insertion penalty of 0. We use a beam size of 1,500. We use the same LM decoding hyperparameters for all models.

[3]Smart data loading with proper pre-fetch is needed here.

E VALIDATION SET DISCUSSION

LRS3 (Afouras et al., 2018b) does not provide a validation set, therefore Shi et al. (2022a) randomly selected 1,200 samples (about 1h) from the 30h training set as the validation set. Several works since then have followed this setup (Haliassos et al., 2023; Zhu et al., 2023; Lian et al., 2023; Hsu et al., 2021a), however, so far no work has reported the performance of their final models on the validation set, except for an AV-HuBERT VSR ablation study (Shi et al., 2022a). We find it important to report the results on the validation set since the hyperparameters are tuned on the validation set with the test set held out until the final decoding. In most scenarios, the performance on the validation set is better than performance on the test set. However, for ASR, performance is better on the test set than on the validation set when using characters as the output units (Table G2). One interesting observation is that for VSR, the performance on the validation set is much better than the performance on the test set (Table F2), regardless of whether characters or subwords are used as the output units (Table G3). In some cases, the performance on the validation set is more than 20% absolute better than on the test set. Shi et al. (2022a) also report better VSR performance on the validation set compared to the test set by 9% absolute WER (Table D.1). Moreover, better performance on the validation set does not reliably indicate better performance on the test set. For example, the video-only V-CPL Base and Large models achieve 37.2% and 27.5% WER respectively on the validation set (Table F2) which is a significant difference, but they achieve 55.6% and 55.9% WER respectively on the test set, which is practically the same result. Upon further investigation, we found that the transcriptions for 1,044 of the 1,200 samples in the validation set are exact substrings of samples in the training set, while only 165 of the 1,321 samples in the test set are exact substrings of samples in the training set, which could potentially explain the discrepancy in performance on the sets and causes concern about over-fitting to particular sequences. Another reason could be that the test set may have more challenging visual conditions, for example, the test set may have faces shot at large angles, which would make VSR harder (Shillingford et al., 2019).

F AUDIO-ONLY AND VIDEO-ONLY CONTINUOUS PSEUDO-LABELING

Table F1: Comparison of audio-only semi-supervised methods: self-supervised learning and continuous pseudo-labeling. We reproduced SlimIPL (Likhomanenko et al., 2021a) on LRS3. The best results on the test set are bolded both with greedy decoding ("None") and with language model (LM) beam-search decoding (LM is trained either on 30h or 433h of LRS3 transcriptions). HuBERT (Hsu et al., 2021a) results presented by Shi et al. (2022a). All prior works use S2S encoder-decoder transformers w/ or w/o CTC loss except Ma et al. (2021b) used Conformer. RAVEn is from Haliassos et al. (2023).

Method	Encoder Size	Criterion	Labeled Data	Unlabeled Data	PL Stage	LRS3 Val WER (%)			LRS3 Test WER (%)		
						None	LM 30h	LM 433h	None	LM 30h	LM 433h
Supervised											
RAVEn	328M	CTC+S2S	30h	-	-	-	-	-	**9.9**	-	-
SlimIPL (Ours)	256M	CTC	30h	-	-	20.0	15.4	10.8	11.1	8.0	**7.3**
E2E-Conformer	-	CTC+S2S	433h	-	-	-	-	-	2.3	-	-
RAVEn	328M	CTC+S2S	433h	-	-	-	-	-	**2.2**	-	-
SlimIPL (Ours)	256M	CTC	433h	-	-	3.4	3.1	2.1	2.5	2.2	**2.1**
Semi-Supervised											
HuBERT	300M	S2S	30h	433h	-	-	-	-	4.5	-	-
SlimIPL (Ours)	256M	CTC	30h	433h	PL LRS3	11.1	8.8	6.1	5.2	3.6	3.2
SlimIPL (Ours)	256M	CTC	30h	433h	+FT LRS3	9.9	8.3	6.5	**4.3**	3.2	**3.1**
HuBERT	300M	S2S	30h	1,759h	-	-	-	-	**3.2**	-	-
SlimIPL (Ours)	256M	CTC	30h	1,759h	PL (Vox+LRS3)	12.2	9.0	5.9	5.7	3.9	3.3
SlimIPL (Ours)	256M	CTC	30h	1,326h	PL Vox	12.2	9.4	6.2	6.3	4.3	3.9
SlimIPL (Ours)	256M	CTC	30h	1,759h	+PL LRS3	9.7	8.5	7.3	4.5	3.4	3.3
SlimIPL (Ours)	256M	CTC	30h	1,759h	+FT LRS3	9.4	8.4	7.4	3.8	3.2	**3.0**

In Table F1, we compare audio-based semi-supervised learning methods: HuBERT (Hsu et al., 2021a) as the SSL method and SlimIPL (Likhomanenko et al., 2021a) as the CPL method. We trained the SlimIPL method ourselves on LRS3 following the original model and hyperparameters. We report both the greedy and LM decoding results on both the LRS3 validation and test sets. We use the LRS3 30h training set as the labeled data, and either use the LRS3 433h training set as the unlabeled data or the combination of the LRS3 433h training data and VoxCeleb2 1,326h training data as unlabeled data. Comparing the supervised baselines, our model is able to match or outperform the reported state-of-the-art performance using a simple pipeline (encoder-only transformer with CTC loss compared to joint CTC and cross-entropy loss with a S2S encoder-decoder transformer).

Table F2: Comparison of video-only semi-supervised methods: self-supervised learning and continuous pseudo-labeling. AV-HuBERT (Shi et al., 2022a) results are for training with video-only and use S2S encoder-decoder transformers. The best results on the test set are bolded both with greedy decoding ("None") and with language model (LM) beam-search decoding (LM is trained either on 30h or 433h of LRS3 transcriptions).

Method	Model	Encoder Size	Criterion	LRS3 Labeled	VoxCeleb2 Unlabeled	LRS3 Val WER (%) None	LRS3 Val WER (%) LM 30h	LRS3 Val WER (%) LM 433h	LRS3 Test WER (%) None	LRS3 Test WER (%) LM 30h	LRS3 Test WER (%) LM 433h
Supervised											
AV-HuBERT	Base	103M	S2S	30h	-	-	-	-	**94.3**	-	-
V-CPL	Base	95M	CTC	30h	-	113.9	95.5	95.6	116.2	95.8	**96.1**
AV-HuBERT	Base	103M	S2S	433h	-	-	-	-	**60.3**	-	-
V-CPL	Base	95M	CTC	433h	-	64.1	55.2	50.6	73.6	65.1	**65.0**
AV-HuBERT	Large	325M	S2S	30h	-	-	-	-	**92.3**	-	-
V-CPL	Large	314M	CTC	30h	-	101.6	96.2	96.1	103.7	96.5	**96.6**
AV-HuBERT	Large	325M	S2S	433h	-	-	-	-	**62.3**	-	-
V-CPL	Large	314M	CTC	433h	-	41.3	35.7	32.4	66.0	61.1	**60.6**
*Semi-Supervised with **External ASR** Models*											
AV-HuBERT	Large	325M	S2S	433h	1,326h	-	-	-	**51.7**	-	-
*Semi-Supervised with Continuous Pseudo-Labeling (**Ours**)*											
V-CPL	Base	95M	CTC	433h	1,326h	51.3	43.3	37.2	**63.7**	56.4	**55.6**
V-CPL	Large	314M	CTC	433h	1,326h	37.1	31.5	27.5	**61.0**	55.9	**55.9**

Comparing the semi-supervised methods, we find that SlimIPL can exceed HuBERT's performance. With 30 hours of labeled data and 433h of LRS3 unlabeled data, SlimIPL achieves 3.1% WER compared to HuBERT's 4.5% WER. Although directly performing CPL on the combination of LRS3 and VoxCeleb2 unlabeled data performs well, we find that performing CPL first on VoxCeleb2 and then on LRS3, followed by fine-tuning on the 30h labeled data in LRS3 works better and alleviates the domain mismatch between the labeled and unlabeled data. After these rounds of training on a total amount of 1,759h of unlabeled data from LRS3 and VoxCeleb2, SlimIPL achieves 3.0% WER compared to HuBERT's 3.2% WER. These results show that audio-only CPL methods transfer well to new datasets and are competitive with SSL methods, even with a simpler pipeline.

In Table F2, we show the full results of video-only continuous pseudo-labeling (V-CPL), including results with the Base model and results on the validation set. Our Base models achieve similar performance to the Base video-only AV-HuBERT trained from scratch without self-supervised learning, although our models use only an encoder with beam-search decoding and a 4-gram LM instead of a S2S encoder and transformer decoder. Applying V-CPL to the Base model, the WER with LM decoding is improved to 55.6%, which is even better than the Large model (55.9%). However, the Large model's greedy decoding performance (63.7%) is better than the Base model's (61.0%).

G Ablation Studies

Table G1: Ablation study on video-only continuous pseudo-labeling λ (ratio of unsupervised to supervised updates). Experiments are conducted with LRS3 433h labeled video-only data and VoxCeleb2 1,326h unlabeled video-only data. We report greedy ("None") and beam-search decoding with a language model (LM) trained on 433h of LRS3 transcriptions.

Transformer	λ	LRS3 Val WER (%) No LM	LRS3 Val WER (%) LM 433h	LRS3 Test WER (%) No LM	LRS3 Test WER (%) LM 433h
Base	1 / 1	**51.3**	**37.2**	**63.7**	**55.6**
Base	3 / 1	82.1	94.3	99.4	86.3
Base	1 / 3	68.6	56.1	77.2	67.9
Large	1 / 1	37.1	27.5	61.0	**55.9**
Large	3 / 1	40.8	29.7	65.9	59.3
Large	1 / 3	**34.0**	**26.4**	**60.6**	56.0

In Table G1, we study $\lambda = N_U/N_L$, the ratio of the number of unsupervised to supervised updates during video-only CPL (V-CPL). We find a ratio of 1 / 1 to work the best in most cases. We therefore adopt this ratio for the video-only and audio-visual CPL experiments.

In Table G2, we compare different combinations of output tokens and strides for the supervised ASR models (Likhomanenko et al., 2021a). We follow Shi et al. (2022a) to construct unigram-based subwords with a vocabulary size of 1k (Kudo, 2018). We use 433h of labeled audio from

Table G2: Ablation study on token set (characters and subwords) vs stride for audio-only ASR using 433h of labeled LRS3 audio and Transformer-Base. We report greedy ("None") and beam-search decoding with a language model (LM) trained on 433h of LRS3 transcriptions.

Tokens	Stride	LRS3 Val WER (%)		LRS3 Test WER (%)	
		No LM	LM 433h	No LM	LM 433h
Characters	20ms	**5.3**	**2.4**	**3.2**	**2.3**
Characters	40ms	12.5	7.0	7.4	5.3
Subwords	20ms	**3.7**	**3.3**	**8.5**	**6.9**
Subwords	40ms	4.6	3.7	8.9	**6.9**

LRS3 and the Transformer-Base model. The audio encoder is a convolutional layer with a kernel width of 7. Prior work keeps the video's native stride of 40ms and stacks 4 audio spectrogram frames to match the video frame stride (Shi et al., 2022a). However, in Table G2, we show that performance is always better with a 20ms stride using either characters or subwords as the output token. The best performance is obtained with character tokens and 20ms stride.

Table G3: Ablation study on token set (characters and subwords) for VSR. Experiments are conducted with LRS3 433h labeled video-only data and VoxCeleb2 1,326h unlabeled video-only data. We report greedy ("None") and beam-search decoding with a language model (LM) trained on 30h or 433h of LRS3 transcriptions.

Tokens	LRS3	VoxCeleb2	LRS3 Val WER (%)			LRS3 Test WER (%)		
	Labeled	Unlabeled	None	LM 30h	LM 433h	None	LM 30h	LM 433h
Characters	30h	-	113.9	**95.5**	**95.6**	116.2	95.8	96.1
Subwords	30h	-	**97.0**	**95.5**	95.7	**98.4**	**95.6**	**95.8**
Characters	433h	-	64.1	**55.2**	50.6	73.6	**65.1**	65.0
Subwords	433h	-	**57.2**	55.6	**45.3**	**70.7**	65.4	**64.9**
Characters	433h	1,326h	51.3	**43.3**	37.2	**63.7**	**56.4**	**55.6**
Subwords	433h	1,326h	**46.0**	45.2	**33.8**	65.4	61.0	60.6

In Table G3, we compare characters to subwords as the output unit for the video-only model. We use the video's native stride of 40ms. Although subwords achieve better performance when training purely on labeled data, characters achieve significantly better performance when performing pseudo-labeling with unlabeled data (55.6% vs 60.6%).

We proposed to pre-train the audio encoder for supervised AVSR according to the results in Table 2c. We show the full results of such pre-training for the Transformer-Base model trained on 433h of labeled data, including results on the validation set and results with greedy decoding in Table G4. We show the results of these experiments for the Large model on 433h in Table G5, as well as the Base model on 30h in Table G6 and the Large model on 30h in Table G7. We note that such pre-training becomes less necessary for the Large model on 433h since the ASR, AVSR, and VSR performance is nearly the same both with and without pre-training, which shows that it is easier to learn from both modalities given enough data and representational power.

H AV-CPL Full Results

We show the full results of AV-CPL using 433h and 30h labeled LRS3 data including results on the validation set and with greedy decoding in Table H1 and Table H2.

Table G4: AVSR modality pre-training ablation with labeled LRS3 433h and Transformer-Base. We report greedy ("no LM") and beam-search decoding ("w/ LM") with a language model (LM) trained on 433h of LRS3 transcriptions.

Train Mods.	PT	Mod. Drop	ASR WER (%) Val		ASR WER (%) Test		AVSR WER (%) Val		AVSR WER (%) Test		VSR WER (%) Val		VSR WER (%) Test	
			No LM	w/ LM	No LM	w/ LM	No LM	w/ LM	No LM	w/ LM	No LM	w/ LM	No LM	w/ LM
AV	-	N	25.9	13.5	24.5	15.4	**5.9**	**3.3**	**6.6**	**4.5**	**68.7**	**56.3**	**74.0**	**66.7**
AV	-	Y	**8.7**	**4.2**	**4.7**	**3.0**	14.3	7.8	13.3	9.6	75.7	66.4	81.0	74.6
A	-	-	**5.1**	**2.4**	**3.8**	**2.4**	95.6	94.1	95.7	94.8	99.9	99.9	99.8	99.9
V	-	-	99.9	99.9	99.9	99.9	**67.3**	**49.8**	**77.6**	**68.1**	**67.1**	**49.9**	**77.6**	**67.7**
AV	A	N	6.0	3.2	4.8	3.2	**3.6**	**1.9**	3.7	**2.5**	76.9	67.1	79.6	71.3
AV	A	Y	**5.6**	**2.9**	**3.6**	**2.6**	5.6	2.8	**3.4**	2.6	**70.7**	**60.7**	**74.9**	**67.0**
AV	V	N	93.9	92.4	89.7	85.9	**14.4**	**8.7**	**27.1**	**21.7**	**50.5**	**39.1**	**69.8**	**63.8**
AV	V	Y	**11.2**	**5.7**	**7.2**	**4.7**	19.5	12.3	29.3	23.5	56.5	47.2	71.6	66.7

Table G5: AVSR modality pre-training ablation with labeled LRS3 433h and Transformer-Large. We report greedy ("no LM") and beam-search decoding ("w/ LM") with a language model (LM) trained on 433h of LRS3 transcriptions.

Train Mods.	PT	Mod. Drop	ASR WER (%) Val		ASR WER (%) Test		AVSR WER (%) Val		AVSR WER (%) Test		VSR WER (%) Val		VSR WER (%) Test	
			No LM	w/ LM	No LM	w/ LM	No LM	w/ LM	No LM	w/ LM	No LM	w/ LM	No LM	w/ LM
AV	-	N	22.4	13.1	31.6	26.8	**4.7**	**2.4**	**4.1**	**3.1**	65.2	53.1	66.8	59.3
AV	-	Y	**6.2**	**3.2**	**4.2**	**2.7**	6.0	3.2	4.8	**3.1**	**56.4**	**45.9**	**65.0**	**58.6**
A	-	-	3.8	2.1	2.7	2.0	97.8	97.8	98.0	98.1	99.9	99.9	99.9	99.9
AV	A	N	5.8	**2.9**	4.4	3.3	**4.5**	**2.3**	**3.5**	**2.7**	78.6	71.0	79.9	73.7
AV	A	Y	**5.6**	3.0	**3.9**	**3.0**	5.4	2.9	3.7	3.0	**60.6**	**49.8**	**66.2**	**58.6**

Table G6: AVSR modality pre-training ablation with labeled LRS3 30h and Transformer-Base. We report greedy ("no LM") and beam-search decoding ("w/ LM") with a language model (LM) trained on 30h and 433h of LRS3 transcriptions.

Train Mods.	PT	Mod. Drop	ASR WER (%) Val			ASR WER (%) Test			AVSR WER (%) Val			AVSR WER (%) Test			VSR WER (%) Val			VSR WER (%) Test		
			No LM	30h	433h	No LM	30h	433h	No LM	30h	433h	No LM	30h	433h	No LM	30h	433h	No LM	30h	433h
AV	-	N	89.0	86.3	62.1	86.3	59.2	57.5	83.1	78.0	46.3	78.0	**46.1**	**44.2**	**99.1**	99.3	98.1	**99.3**	98.4	98.2
AV	-	Y	**52.4**	**38.0**	**32.0**	**40.4**	**27.2**	**25.9**	**74.5**	**65.2**	**63.1**	**64.6**	53.7	53.1	100.5	**95.0**	**95.0**	102.0	**95.3**	**95.5**
A	-	-	22.1	16.6	12.4	13.5	8.8	7.9	48.4	39.2	34.3	37.0	29.1	28.2	99.9	99.9	99.9	99.9	99.9	99.9
AV	A	N	31.9	25.1	20.4	21.2	15.5	14.7	**24.5**	**19.3**	**15.1**	**15.9**	**11.6**	**11.0**	**94.8**	**90.0**	**90.3**	**95.5**	**89.8**	**90.0**
AV	A	Y	**23.3**	**17.3**	**13.1**	**13.9**	**9.5**	**8.9**	29.1	22.2	17.6	18.0	12.3	11.4	99.7	93.3	93.4	101.4	94.1	94.3

Table G7: AVSR modality pre-training ablation with labeled LRS3 30h and Transformer-Large. We report greedy ("no LM") and beam-search decoding ("w/ LM") with a language model (LM) trained on 30h and 433h of LRS3 transcriptions.

Train Mods.	PT	Mod. Drop	ASR WER (%) Val			ASR WER (%) Test			AVSR WER (%) Val			AVSR WER (%) Test			VSR WER (%) Val			VSR WER (%) Test		
			No LM	30h	433h	No LM	30h	433h	No LM	30h	433h	No LM	30h	433h	No LM	30h	433h	No LM	30h	433h
AV	-	N	86.8	68.7	62.5	84.4	61.5	60.2	81.6	**56.8**	**45.9**	75.4	45.9	43.7	**98.6**	**97.3**	**97.2**	**98.7**	**97.1**	**97.1**
AV	-	Y	**38.7**	**30.9**	**26.4**	**27.0**	**20.7**	**19.5**	**67.1**	60.4	58.5	**50.2**	**43.6**	**43.0**	101.1	97.5	97.3	102.5	98.0	98.0
A	-	-	21.8	17.4	13.7	13.4	10.0	9.5	98.0	97.5	97.2	96.2	95.2	95.0	99.9	99.9	99.9	99.9	99.9	99.9
AV	A	N	25.0	20.0	16.4	16.0	12.0	11.6	**21.1**	**17.1**	**13.7**	**12.8**	**9.6**	**9.0**	96.0	90.0	90.3	95.5	88.7	88.6
AV	A	Y	**22.3**	**18.2**	**14.4**	**14.0**	**10.6**	**10.0**	22.1	18.1	14.4	13.4	10.1	9.7	**93.8**	**86.9**	**86.9**	**95.1**	**87.1**	**87.0**

Table H1: AV-CPL main results on LRS3 433h labeled videos reported on LRS3 val and test sets. The seed models use modality dropout $p_m = p_a = 0.5$. We report greedy ("no LM") and beam-search decoding ("w/ LM") with a language model (LM) trained on 433h of LRS3 transcriptions.

Model	Mod. Drop	VoxCeleb2 Unlabeled	ASR WER (%) Val		ASR WER (%) Test		AVSR WER (%) Val		AVSR WER (%) Test		VSR WER (%) Val		VSR WER (%) Test	
	$p'_m = p'_a$		No LM	w/ LM	No LM	w/ LM	No LM	w/ LM	No LM	w/ LM	No LM	w/ LM	No LM	w/ LM
Base	Seed	-	**5.6**	**2.9**	3.6	2.6	**5.6**	**2.8**	3.4	2.6	70.7	60.7	74.9	67.0
Base	0.1	1,326h	9.9	4.9	4.8	3.3	9.2	4.9	4.6	3.0	**44.5**	**30.9**	**55.8**	**48.4**
Base	0.5	1,326h	7.0	3.4	**3.5**	**2.2**	6.3	3.1	**3.2**	**2.0**	61.6	46.8	64.9	55.7
Large	Seed	-	6.2	3.2	4.2	2.7	6.0	3.2	4.8	3.1	56.4	45.9	65.0	58.6
Large	0.1	1,326h	8.8	4.6	4.9	3.4	8.0	4.1	4.4	3.0	**33.3**	**24.1**	**51.0**	**45.3**
Large	0.5	1,326h	**5.1**	**3.0**	**3.0**	**2.3**	**4.8**	**2.8**	**3.2**	**2.2**	46.0	34.7	54.3	47.4

Table H2: AV-CPL main results on LRS3 30h labeled videos reported on LRS3 val and test sets. The seed models use modality dropout $p_m = p_a = 0.5$ while the AV-CPL models use modality dropout $p'_m = p'_a = 0.1$. We report greedy ("no LM") and beam-search decoding with a language model (LM) trained on 30h and 433h of LRS3 transcriptions.

Model	Unlabeled Hours	PL Stage	ASR WER (%)						AVSR WER (%)						VSR WER (%)					
			Val			Test			Val			Test			Val			Test		
			No LM	30h	433h	No LM	30h	433h	No LM	30h	433h	No LM	30h	433h	No LM	30h	433h	No LM	30h	433h
Base	-	Seed	23.3	17.3	13.1	13.9	9.5	8.9	29.1	22.2	17.6	18.0	12.3	11.4	99.7	93.3	93.4	101.4	94.1	94.3
Base	433h	PL LRS3	29.3	27.9	18.4	16.9	14.8	13.2	32.9	28.5	22.9	22.2	18.0	17.0	71.2	61.4	58.9	74.5	66.6	66.4
Base	1,759h	PL (Vox + LRS3)	**22.0**	**17.9**	**10.6**	**14.2**	**10.7**	**9.5**	**20.8**	**17.0**	**11.1**	**12.8**	**9.0**	**8.2**	71.2	62.5	59.9	70.5	63.2	62.9
Base	1,326h	PL Vox	28.4	24.6	18.0	20.6	15.1	14.4	30.6	24.8	20.9	19.2	14.3	13.6	74.1	66.3	65.0	71.7	63.5	63.3
Base	1,759h	+ PL LRS3	24.7	20.6	13.4	15.2	12.0	10.8	26.6	22.9	18.3	17.9	14.4	13.6	**61.6**	**51.9**	**49.1**	**65.3**	**57.3**	**57.3**
Large	-	Seed	22.3	18.2	14.4	14.0	10.6	10.0	22.1	18.1	14.4	13.4	10.1	9.7	93.8	86.9	86.9	95.1	87.1	87.0
Large	433h	PL LRS3	19.0	16.1	11.3	12.9	10.2	9.5	18.9	16.0	11.7	12.2	9.6	9.0	58.2	50.1	47.2	67.8	61.4	61.3
Large	1,759h	PL (Vox + LRS3)	**17.4**	**13.9**	**8.9**	**9.8**	**7.1**	**6.6**	**17.6**	**14.3**	**9.0**	**9.5**	**7.4**	**6.4**	66.4	60.1	57.5	68.1	62.5	62.4
Large	1,326h	PL Vox	33.1	19.7	20.8	19.7	16.2	15.4	28.6	16.6	17.1	16.6	12.8	12.2	74.1	70.2	64.3	70.2	63.2	63.1
Large	1,759h	+PL LRS3	19.6	17.2	11.5	13.1	10.8	10.0	20.3	20.3	12.7	13.3	13.1	10.4	**54.9**	**48.1**	**43.3**	**63.1**	**57.5**	**56.7**