Supplementary Material: Audio-Visual Instance Discrimination with Cross-Modal Agreement

A. Experimental setup

Architecture details The architecture details of the video and audio networks used in the analysis experiments are shown in Table 5, and those used for comparison to prior work is shown in Table 6.

Pre-training hyper-parameters Optimization and data augmentation hyper-parameters for AVID and CMA pre-training are provided in Table 3.

Action recognition hyper-parameters Optimization and data augmentation hyper-parameters for action recognition tasks are provided in Table 4.

Video pre-processing Video clips are extracted at 16 fps and augmented with standard techniques, namely random multi-scale cropping with 8% minimum area, random horizontal flipping and color and temporal jittering. Color jittering hyper-parameters are shown in Table 3 for pre-training and Table 4 for transfer into downstream tasks.

Audio pre-processing Audio signals are loaded at 24kHz, instead of 48kHz, because a large number of Audioset audio samples do not contain these high frequencies. The spectrogram is computed by taking the FFT on 20ms windows with either 10ms (§4, §5) or 20ms (§6) hop-size. We then convert the spectrogram to a log scale, and Z-normalize its intensity using mean and standard deviation values computed on the training set. We use volume and temporal jitering for data augmentation. Volume jittering is accomplished by multiplying the audio waveform by a constant factor randomly sampled between 0.9 and 1.1, and applied uniformly over time. Temporal jittering is done by randomly sampling the audio starting time within 0.5s of the video, and randomly selecting the total audio duration between 1.4s and 2.8s and rescaling back to the expected number of audio frames.

B. Longer AVID pre-training

To ensure that the benefits of CMA are not caused by longer training, we trained Cross-AVID for the same number of epochs as AVID+CMA. The Cross-AVID performance on Kinetics after 200 and 400 training epochs are shown in Table 1. Cross-AVID transfer performance seem to have already saturated after 200 epochs of pre-training.

Table 1: Top-1 accuracy of linear probing on Kinetics evaluated after 200 and 400 epochs of Cross-AVID training.

Method	block1	block2	block3	block4	Best
Cross-AVID (ep 200)	19.84	26.87	34.64	39.87	39.87
Cross-AVID (ep 400)	19.80	26.98	34.81	39.95	39.95

Table 2: Top-1 accuracy of linear probing of memory representations (video, audio and both concatenated).

Method	Video Mem	Audio Mem	Combined Mem
Cross-AVID	29.01±0.14	19.67±0.09	34.68±0.15
CMA	**34.00±0.25**	**21.98±0.11**	**38.91±0.14**

C. CMA calibration

To further study the benefits effect of the CMA procedure, we measured the classification performance of memory representations obtained with both AVID and CMA trained on the Kinetics dataset. We randomly split the 220K training samples, for which memory representations are available, into a train/validation set (70/30% ratio). We then train a linear classifier on the training set (using either video, audio or the concatenation of both, ConvNet is kept fixed), and evaluate the performance on the validation set. The train/validation splits are sampled 5 times and average performance is reported. The top-1 accuracies are shown in Table 2.

Table 3: Pre-training optimization hyper-parameters. CMA models are initialized by the AVID model obtained at epoch 200. bs batch size; lr learning rate; wd weight decay; ep number of epochs; es number of samples per epoch; msc - multi-scale cropping; hf - horizontal flip probability; bj/sj/cj/hj - brightness/saturation/contrast/hue jittering intensity.

Method	DB	bs	lr	wd	ep	es	msc	hf	bj	sj	cj	hj
AVID (§4)	Audioset	32	5e-4	1e-5	400	1e5	✓	0.5	0.4	0.4	0.4	0.2
AVID (§6)	Audioset	32	5e-4	1e-5	200	1.8e6	✓	0.5	0.4	0.4	0.4	0.2
AVID (§6)	Kinetics	32	2e-4	1e-5	300	2.4e5	✓	0.5	0.4	0.4	0.4	0.2
CMA (§5.3, §5.4)	Audioset	32	5e-4	1e-5	200	1e5	✓	0.5	0.4	0.4	0.4	0.2
CMA (§6)	Audioset	32	5e-4	1e-5	200	1.8e6	✓	0.5	0.4	0.4	0.4	0.2
CMA (§6)	Kinetics	32	2e-4	1e-5	300	2.4e5	✓	0.5	0.4	0.4	0.4	0.2

Table 4: Transfer learning optimization and data augmentation hyper-parameters. bs - batch size; lr - learning rate; wd - weight decay; ep - number of epochs; es - number of samples per epoch; gm - learning rate decay factor; mls - milestones for learning rate decay; msc - multi-scale cropping; hf - horizontal flip probability; bj/sj/cj/hj - brightness/saturation/contrast/hue jittering intensity.

DB	input size	bs	lr	wd	ep	es	gm	mls
Kinetics (§4, §5)	16×112^2	32	1e-4	0.	20	1e4	0.3	8,12,15,18
UCF (§6)	8×224^2	32	1e-4	0.	160	1e4	0.3	60,100,140
UCF (§6)	32×224^2	16	1e-4	0.	80	1e4	0.3	30,50,70
HMDB (§6)	8×224^2	32	1e-4	0.	250	3.4e3	0.3	75,150,200
HMDB (§6)	32×224^2	16	1e-4	0.	100	3.4e3	0.3	30,60,80

DB	msc	hf	bj	sj	cj	hj
Kinetics (§4, §5)	✓	0.5	0.	0.	0.	0.
UCF (§6)	✓	0.5	0.4	0.4	0.4	0.2
HMDB (§6)	✓	0.5	1.	1.	1.	0.2

Table 5: Architecture details of R(2+1)D video network and Conv2D audio network for analysis experiments (§4, §5.3, §5.4). The video network is based of R(2+1)D convolutions, and the audio on 2D convolutions. Both video and audio networks use ReLU activations and batch normalization at each layer. X_s spatial activation size, X_t temporal activation size, X_f frequency activation size, C number of channels, K_s spatial kernel size, K_t temporal kernel size, K_f frequency kernel size, S_s spatial stride, S_t temporal stride, S_f frequency stride.

Video Network

Layer	X_s	X_t	C	K_s	K_t	S_s	S_t
`video`	112	16	3	-	-	-	-
`conv1`	56	16	64	7	3	2	1
`block2.1`	56	16	64	3	3	1	1
`block2.2`	56	16	64	3	3	1	1
`block3.1`	28	8	128	3	3	2	2
`block3.2`	28	8	128	3	3	1	1
`block4.1`	14	4	256	3	3	2	2
`block4.2`	14	4	256	3	3	1	1
`block5.1`	7	2	512	3	3	2	2
`block5.2`	7	2	512	3	3	1	1
`max pool`	1	1	512	7	2	1	1
`fc1`	-	-	512	-	-	-	-
`fc2`	-	-	512	-	-	-	-
`fc3`	-	-	128	-	-	-	-

Audio Network

Layer	X_f	X_t	C	K_f	K_t	S_f	S_t
`audio`	129	100	1	-	-	-	-
`conv1`	65	50	64	7	7	2	2
`block2.1`	65	50	64	3	3	1	1
`block2.2`	65	50	64	3	3	1	1
`block3.1`	33	25	128	3	3	2	2
`block3.2`	33	25	128	3	3	1	1
`block4.1`	17	13	256	3	3	2	2
`block4.2`	17	13	256	3	3	1	1
`block5.1`	17	13	512	3	3	1	1
`block5.2`	17	13	512	3	3	1	1
`max pool`	1	1	512	17	13	1	1
`fc1`	-	-	512	-	-	-	-
`fc2`	-	-	512	-	-	-	-
`fc3`	-	-	128	-	-	-	-

Table 6: Architecture details of R(2+1)D video network and Conv2D audio network for comparison to prior work (§6). The video network is based of R(2+1)D convolutions, and the audio on 2D convolutions. Both video and audio networks use ReLU activations and batch normalization at each layer. X_s spatial activation size, X_t temporal activation size, X_f frequency activation size, C number of channels, K_s spatial kernel size, K_t temporal kernel size, K_f frequency kernel size, S_s spatial stride, S_t temporal stride, S_f frequency stride.

Video Network

Layer	X_s	X_t	C	K_s	K_t	S_s	S_t
video	224	8	3	-	-	-	-
conv1	112	8	64	7	3	2	1
max-pool	56	8	64	3	1	2	1
block2.1.1	56	8	64	3	3	1	1
block2.1.2	56	8	64	3	3	1	1
block2.2.1	56	8	64	3	3	1	1
block2.2.2	56	8	64	3	3	1	1
block3.1.1	28	4	128	3	3	2	2
block3.1.2	28	4	128	3	3	1	1
block3.2.1	28	4	128	3	3	1	1
block3.2.2	28	4	128	3	3	1	1
block4.1.1	14	2	256	3	3	2	2
block4.1.2	14	2	256	3	3	1	1
block4.2.1	14	2	256	3	3	1	1
block4.2.2	14	2	256	3	3	1	1
block5.1.1	7	1	512	3	3	2	2
block5.1.2	7	1	512	3	3	1	1
block5.2.1	7	1	512	3	3	1	1
block5.2.2	7	1	512	3	3	1	1
max-pool	1	1	512	7	2	1	1
fc1	-	-	512	-	-	-	-
fc2	-	-	512	-	-	-	-
fc3	-	-	128	-	-	-	-

Audio Network

Layer	X_f	X_t	C	K_f	K_t	S_f	S_t
audio	257	200	1	-	-	-	-
conv1	129	100	64	7	7	2	2
block2.1	65	50	64	3	3	2	2
block2.2	65	50	64	3	3	1	1
block3.1	33	25	128	3	3	2	2
block3.2	33	25	128	3	3	1	1
block4.1	17	13	256	3	3	2	2
block4.2	17	13	256	3	3	1	1
block5.1	17	13	512	3	3	1	1
block5.2	17	13	512	3	3	1	1
max pool	1	1	512	17	13	1	1
fc1	-	-	512	-	-	-	-
fc2	-	-	512	-	-	-	-
fc3	-	-	128	-	-	-	-