A Appendix

A.1 Pretrained model details

Here we provide additional information about the pretrained models we have used in this work.

Table A.1: **Details for audio encoder.** Architectural and pretraining details for XDC's audio encoder used for benchmarking.

Method	Input shape	Architecture	Pretrain dataset
XDC	$40 \times 1 \times 100$	Resnet-18	Kinetics-400

A.2 Implementation details

We train our method using the Sinkhorn-Knopp parameter $\lambda = 20$, an inverse quadratic clustering schedule with 100 clustering operations and 10 heads which we adopt from [6]. For evaluation, we report results for head 0 to compare against the ground-truth, as we found no significant difference in performance between heads. For the Gaussian distribution, we take the marginals to be from $\mathcal{N}(1, 0.1) * N/K$. For the clustering-heads, we use two-layer MLP-heads as in [8, 18]. The video inputs are 30 frame long clips sampled consecutively from 30fps videos and are resized such that the shorter side is 128 and during training a random crop of size 112 is extracted, no color-jittering is applied. Random horizontal flipping is applied to the video frames with probability 0.5, and then the channels of the video frames are Z-normalized using mean and standard deviation statistics computed across the dataset. The audio is processed as a $1 \times 257 \times 199$ image, by taking the log-mel bank features with 257 filters and 199 time-frames and for training, random volume jittering between 90% and 110% is applied to raw waveform, similar to [54]. For evaluation, a center-crop is taken instead for the video inputs and audio volume is not jittered. We use a mini-batch size of 16 on each of our 64 GPUs giving an effective batch size of 1024 for distributed training for 200 epochs. The initial learning rate is set to 0.01 which we linearly scale with the number of GPUs, after following a gradual warm-up schedule for the first 10 epochs [28]. For training on Kinetics-Sound and AVE, we initialize our model with a VGG-Sound pretrained backbone due to the small training set sizes ($N = 22$k and $N = 3328$). The clustering heads are re-initialized randomly. This ensures a more fair comparison as XDC, DPC and the supervised model are pretrained on Kinetics-400 with $N = 230$k and MIL-NCE on HowTo100M with $N = 100$M videos. We train on VGG-Sound for 200 epochs, which takes around 2 days.

A.3 Pair-based optimization for AV-Alignment

For aligning the visual and audio encoder, we use a greedy switching algorithm that starts from a feasible initial solution [23, 24, 62]. In particular, we consider 50000 potential pair switches with 5 randomized restarts and take the final permutation that yields the lowest cost.

A.4 Evaluation metrics details

The **normalized mutual information** (NMI) is calculated by the formula

$$\text{NMI} = \frac{\text{MI}(U, V)}{0.5H(U) + 0.5H(V)}, \tag{8}$$

where the Mutual information MI is given by $\text{MI}(U, V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} P(i, j) \log\left(\frac{P(i,j)}{P(i)P'(j)}\right)$, and H is the standard entropy, with $H(U) = -\sum_{i=1}^{|U|} P(i) \log(P(i))$. The NMI ranges from 0 (no mutual information) to 100%, which implies perfect correlation.

The rand index (RI) is given by $\text{RI} = \frac{a+b}{C}$, where a, b are the number of pairs of elements that are in the same/different set in the ground truth labelling and in the same/different set in the predicted clustering and C is the total number of such pairs. The a**djusted Rand index** (ARI) corrects for random assignments and is given by

$$\text{ARI} = \frac{\text{RI} - E[\text{RI}]}{\max(\text{RI}) - E[\text{RI}]}, \tag{9}$$

where the expected RI of a random label assignment is subtracted in both nominator and denominator. Due to the subtraction, the ARI varies from -1 to 1 with a value close to 0 implying random correlation and a value of 1 implying identical agreement.

The **mean entropy per cluster** is given by

$$\langle H \rangle = \frac{1}{K} \sum_{k \in K} H(p(y|\hat{y}_k = k)), \tag{10}$$

where $\hat{y}$ are unsupervisedly obtained clusters and $p(y|\hat{y}_k = k)$ is the distribution of ground-truth clusters for cluster k. Hence, the optimal number of this metric is 0 and a chance assignment yields $\langle H \rangle = -\log 1/K$.

Further, as we wish to understand the semantic purity compared to the ground truth labels of each cluster, so we additionally report the the **mean maximal purity per cluster**,

$$\langle p_{\max} \rangle = \frac{1}{K} \sum_{k \in K} \max(p(y|\hat{y}_k = k)), \tag{11}$$

which ranges from $\langle p_{\max} \rangle = 1/K$ (chance level) to perfect matching at $\langle p_{\max} \rangle = 100\%$.

A.5 Single modality degradation experiment details

We use the default input-sizes for each model, i.e. 112 for ours and the supervised model, 224 for MIL-NCE. Compression is implemented by nearest-neighbor downsampling and subsequently nearest-neighbor upsamling for speed. For this experiment only, we evaluate the performance on the smaller validation sets.

A.6 Further ablations

In Table A.2, we provide the results for varying the number of clusters K in our algorithm. We find that even when moving from the ground-truth number of classes ($K = 309$), to lower numbers ($K = 256$) or higher estimates ($K = 619, 1024$) our results remain stable with the NMI staying almost constant. While the ARI does drop for larger K, we also observe an increase in the purity of the clusters for a larger number of clusters from $\langle \mathbf{p}_{\max} \rangle = 38.0$ for $K = 309$ to $\langle \mathbf{p}_{\max} \rangle = 42.7$ for $K = 619$, which can be particularly useful when dividing the dataset into clusters and subsequently only obtaining human annotations for few examples per cluster.

Table A.2: **Varying K** in our method degrades performances only slightly, showing that our method is robust to various estimations of the ground-truth number of classes. Results on VGG-Sound.

Method	K	**NMI**	**ARI**	**Acc.**	$\langle \mathbf{H} \rangle$	$\langle \mathbf{p}_{\max} \rangle$
SeLaVi	309	56.7	22.5	32.3	2.4	38.0
SeLaVi	256	56.8	24.3	34.2	2.4	36.9
SeLaVi	619	56.9	16.8	23.0	2.2	42.7
SeLaVi	1024	55.1	16.3	9.6	2.1	42.2

A.7 Retrieval downstream task implementation details

We follow [78] in our evaluation protocol and use split 1 of UCF101 and HMDB-51. We uniformly sample 10 clips per video, and average the max-pooled features after the last residual block for each clip per video. We then utilize the averaged features from the validation set to query the videos in the training set. The cosine distance of representations between the query clip and all clips in the training set are computed and when the class of a test clip appears in the classes of k nearest training clips, it is considered to be correctly retrieved. R@k refers to the retrieval performance using k nearest neighbors.

A.8 Visual classification downstream task

Table A.3: **Representation learning downstream evaluation.** Self-supervised and fully-supervised trained methods on UCF101 and HMDB51 benchmarks. We follow the standard protocol and report the average top-1 accuracy over the official splits and show results for finetuning the whole network. Methods with † indicate the additional use of video titles and ASR generated text as supervision. Methods with * use ASR generated text.

Method	**Architecture**	**Pretrain Dataset**	**Top-1 Acc%**	
			UCF	HMDB
Full supervision [2]	R(2+1)D-18	ImageNet	82.8	46.7
Full supervision [2]	R(2+1)D-18	Kinetics-400	**93.1**	**63.6**
Weak supervision, CPD [49]†	3D-Resnet50	Kinetics-400	88.7	57.7
MotionPred [73]	C3D	Kinetics-400	61.2	33.4
RotNet3D [39]	3D-ResNet18	Kinetics-600	62.9	33.7
ST-Puzzle [42]	3D-ResNet18	Kinetics-400	65.8	33.7
ClipOrder [78]	R(2+1)D-18	Kinetics-400	72.4	30.9
DPC [30]	3D-ResNet34	Kinetics-400	75.7	35.7
CBT [66]	S3D	Kinetics-600	79.5	44.6
Multisensory [58]	3D-ResNet18	Kinetics-400	82.1	-
XDC [2]	R(2+1)D-18	Kinetics-400	84.2	47.1
AVTS [43]	MC3-18	Kinetics-400	85.8	56.9
AV Sync+RotNet [76]	AVSlowFast	Kinetics-400	87.0	54.6
GDT [60]	R(2+1)D-18	Kinetics-400	88.7	57.8
SeLaVi	R(2+1)D-18	Kinetics-400	83.1	47.1
SeLaVi	R(2+1)D-18	VGG-Sound	87.7	53.1

In Table A.3 we show the performance of our method on two common visual-only video feature representation benchmarks, UCF-101 [65] and HMDB-51 [44]. Note that, as is the standard in this evaluation, we use our visual encoder as initialization and fine-tune the whole network on the target down-stream task. In particular, we follow the finetuning schedule of the one of the current state-of-the-art methods [60]. We find that we achieve competitive performance when trained on VGG-Sound, even surpassing XDC, despite our method using only a spatial resolution of 112×112 and not 224×224.