Supplementary Material for "Multimodal Clustering Networks for Self-supervised Learning from Unlabeled Videos"

This appendix is organized as follows:
A. Method description and comparisons.
B. Details on experimental setups.
C. Additional experiment results.
D. Quantitative experimental results for text-to-video retrieval and temporal action localization.

A. Method description and comparisons.

A.1. Salient features of the MCN method

To highlight various aspects of our proposed MCN method, we compare our method with another notable multimodal cluster method: XDC [2].

Goal of the model. While XDC's goal is to learn representations for each modality, with the MCN method, we try to learn a joint representation across modalities. The two approaches are hence complementary, given that they target different tasks. In addition, XDC aims to learn feature backbones from scratch since these feature backbones will be applied to single modality downstream tasks. In contrast, we start from pre-trained feature extractors and aim to learn projection heads across domains to derive a joint space from the three modalities.

Joint space of representation. Based on the formulation of XDC, pseudo-labels from one modality serve as prediction targets of another. Since the prediction target for the visual and audio instances are different, the model will not learn a joint space across modalities. The paper also proposed a CDC method where the prediction target of visual and audio instances are the same. However, it is not evaluated on multimodal tasks.

Combining contrastive learning. While XDC uses only the clustering loss, we combine multiple losses together. We find the contrastive loss to be crucial in multimodal tasks since it pulls the instances across modalities that co-occur together. In general, this supervision is crucial in most multimodal pre-training strategies.

Use of different modalities Since our goal is to learn a joint space across three modalities, our motivation for using audio is slightly different from XDC. XDC uses audio and video as complimentary learning signals for self-supervised prediction targets. On the other hand, we find audio as a modality that bridges the gap between video and text, since audio and video preserve fine-grained information. The text modality represents a more abstract concept, distilled from the audio signal using ASR. Hence, we find learning from the three modalities to be beneficial.

B. Experiment Details

B.1. Implementation details

We use an Adam optimizer [11] with a learning rate of $1\mathrm{e}{-4}$ and cosine learning rate schedule [16]. The model is trained for 30 epochs on four V100 GPUs over a period of about two days. Various hyperparameters in our experiments are set as follows: margin hyperparameter $\delta = 0.001$, and a batch size of B = 4096 video clips and cluster size is set to be 256.

B.2. Clustering metrics

To better evaluate our learned features, we use the k-means clustering algorithm and calculate various clustering metrics based on ground-truth labels on the CrossTask [22] and MiningYouTube [12] tasks. In this case, the number of clusters k, also corresponds to the number of possible steps assigned to the temporal action localization task for each video during test time.

We follow the evaluation protocol and notations used in [4] and report performance based on the following standard clustering metrics: *normalized mutual information* (NMI) [19], *adjusted rand index* (ARI) [10], and *accuracy* (Acc). These results are obtained after matching the estimated k-means pseudo-labels to the ground truth targets using the Kuhn–Munkres/Hungarian algorithm [13]. We also report the *mean entropy per cluster* :

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

where $\hat{y}$ corresponds to the psuedo-labels generated by clustering and y relates to the ground-truth labels. In this formulation $p(y|\hat{y}_k = k)$ denotes the distribution of ground-truth labels that fall in the generated clusters k, while $H(U)$ represents the entropy given as $-\sum_{i=1}^{|U|} P(i) \log(P(i))$. In ideal conditions, the perfect mean entropy will be zero.

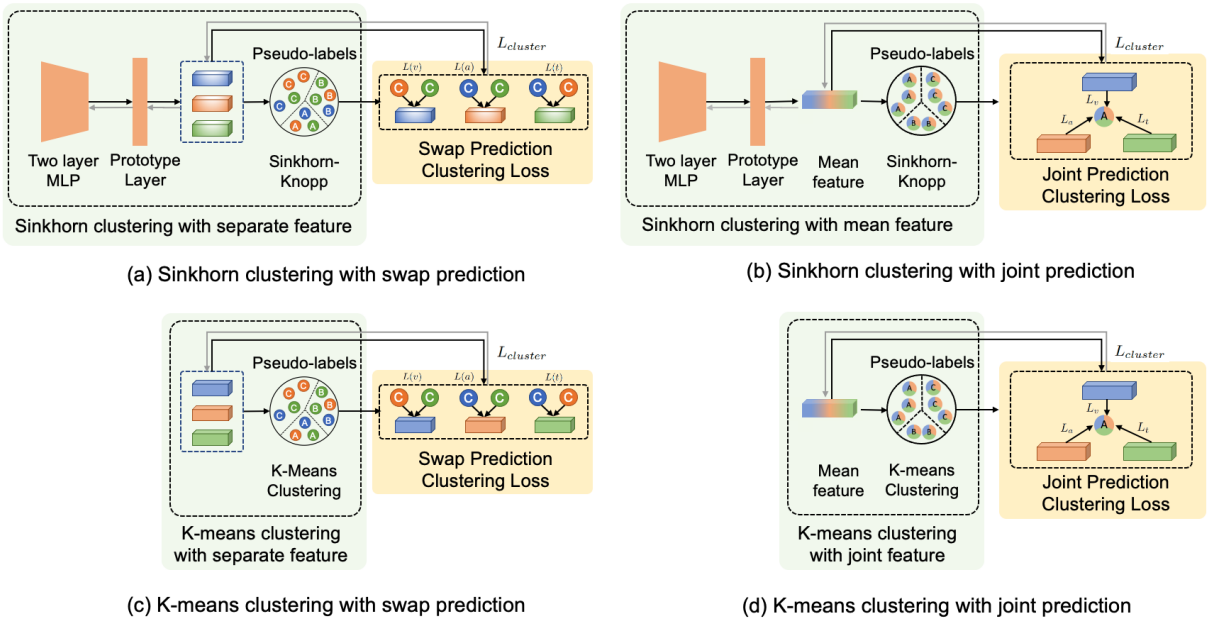

Figure A: Comparison of different clustering pipelines. We investigate different clustering pipelines in replace of the clustering loss in our main paper. (a) Performs a sinkhorn clustering folloing a swap prediction. The loss was calculated between the clustered features and pseudo labels. (b) Replaces the swap prediction to joint prediction by performing the clustering on the mean feature. The loss was calculated by the mean pseudo label and the projected feature in Figure 3a. (c) Performs K-means along with swap prediction. (d) Performs K-means on the mean features and performs joint prediction.

Method	NMI ↑	ARI ↑	Acc. ↑	$\langle \mathbf{H} \rangle \downarrow$	$\langle \mathbf{p}_{\max} \rangle \uparrow$
Random	0.4	0.4	8.6	2.3	25.5
Miech *et al.* [15]	72.9	45.4	59.8	0.44	79.5
MIL-NCE* [14]	73.1	46.8	60.6	0.37	77.9
MCN	**75.8**	**48.0**	**61.7**	**0.40**	**80.8**

Table A: Performance on various clustering metrics for the MiningYouTube task

We also report the the *mean maximal purity per cluster*,

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

In ideal conditions, the perfect mean purity will be 100%.

By using the various metrics described above, the clustering result on MiningYoucook dataset was shown in Table B. The overall results show a similar pattern with the experiment shown in the main paper using CrossTask dataset.

B.3. Clustering ablation explanation

In this ablation study, we investigate different clustering pipelines. Here, we break down our results into several categories and provide an analysis for various clustering methods, different prediction targets, and several kinds of pseudo-labels.

Clustering method. The goal of this analysis is to create various kinds of pseudo-labels as prediction targets. If a pseudo-label can be thought of as a certain semantic representation of a cluster, two instances that have the same pseudo-label, can then be considered as semantically similar. The K-means method follows the deep clustering [7] approach which utilizes K-means clustering to create pseudo labels as prediction targets. These targets are then used for single modality learning on ImageNet [18]. The Sinkhorn clustering method follows the SeLa [5] technique that utilized a trainable network to replace the K-means clustering for generating pseudo-labels. The method also applies an optimal transport sinkhorn algorithm [9] to guarantee uniform distribution over different cluster labels, which in turn prevents the learnable clustering network (2 layers MLP) from learning a degenerated solution. More details of this sinkhorn clustering approach can be found in [5, 8].

Prediction Target. We investigate two sources of pseudo-labels as prediction targets. In the first approach, the **swap** prediction utilizes a pseudo-label created from a different domain as a prediction target. As shown in the yellow box of Figure A (c), pseudo-labels from the audio (orange) and text (green) domains are used as prediction targets for the visual feature (blue). This mechanism is similar to XDC [2] except that we perform this approach on projected features a in common space. In the **joint** prediction method, a mean feature

Method	MMS	MMS + Clus	MMS + Clus + Recon
Aligned ↑	0.740	0.858	0.873
Misaligned ↓	0.327	0.279	0.260

Table B: Cosine similarity of aligned and misaligned instances.

from the features of three modalities is first computed as a multimodal feature representation. Later, its pseudo-label will be the prediction target for the three separate feature instances and will be used to guide the features to be close across modalities and semantics. As shown in Figure A (d), the pseudo-label of the mean feature is used as the prediction target for features of each of the three modalities.

Label type. We have two kinds of labels: hard labels that represent discrete labels and soft labels that represent continuous, probabilistic labels. Since K-means assigns each instance to one of the centroids, it will only produce hard labels. The outputs from the Sinkhorn clustering are from a learnable network. We can use the softmax operator to transfer these outputs into probabilities over different labels (soft) or use the arg-max function to derive discrete labels (hard). When we perform soft-label prediction over the Sinkhorn pipeline as shown in (a), it will be similar to Swav [8], but we perform this over multiple modalities and treat the different modalities as a kind of data augmentation.

B.4. Dataset and computational resources used in each methods

To better compare between different methods and settings, we specify various datasets used to construct each of the baselines in Tables G and H. Methods with pre-trained feature extractors were trained on ImageNet (ImNet), Kinetics (K400), or Visual Genome (VG). Large-scale datasets such as HowTo100M (HT) and AudioSet (AS) are used for self-supervised pre-training. ActBERT [21] uses region features from a faster R-CNN, which is pre-trained on VG to better localize actions in CrossTask. We also include the computation resource and training time of each method. Note that methods [1, 14] with trainable backbones (TR) require 32 or more TPUs and usually perform better. For the reproduced *MIL-NCE method, we use code from [15] and apply the loss of [14] from their Github repo.

C. Additional experiments

C.1. Dealing with miss-alignment across modalities

To quantify the alignment discrepancy across modalities, we first consider the pairwise MMS loss for each modality combination: AT, AV, and VT (V: video, A: audio, T: text). The loss starts equally for all combinations from (16.3, 16.8, 16.4) and decreases to AT=2.4, AV=8.8, VT=10.8 (epoch 10). The AT loss is the lowest since the text was generated

		YouCook2			MSRVTT		
Method	Mod	R@1	R@5	R@10	R@1	R@5	R@10
MMS	T→V	7.4	20.0	29.3	8.8	23.2	32.2
MIL-NCE*	T→V	8.1	23.3	32.3	8.4	23.2	32.4
Ours	T→V	8.6	24.1	33.4	9.6	23.4	32.1
MIL-NCE* + audio	A→V	16.2	36.6	43.7	13.2	28.4	33.3
Ours	A→V	19.4	41.3	50.9	14.8	30.1	39.0
NCE	T→VA	14.5	32.1	39.2	8.8	24.1	33.7
MIL-NCE* + audio	T→VA	15.1	31.9	40.0	9.0	23.3	33.0
MMS	T→VA	16.1	33.9	43.7	9.5	23.3	32.9
Ours	T→VA	18.1	35.5	45.2	10.5	25.2	33.8

Table C: Comparison of retrieval across different modalitites.

from an ASR system, followed by AV since both signals are synchronized, which is relevant for object sounds like sizzling or chopping, and the largest gap can be found for VT pairs. Hence, introducing audio enables us to bridge this gap. We hypothesize that the clustering loss implicitly compensates for this misalignment. To show this effect, we sample V/A/T triplets from the YouCook2 dataset, generate misaligned instances by randomly replacing instances, and compare their cosine similarity to its mean multimodal embedding as in Eq.4 (see Tab. B, columns compare models from the ablation study). With the proposed clustering, aligned instances are closer to the mean embedding while misaligned are further away (as desired). Therefore, the clustering step in training could compensate/correct for the MMS loss, which always pulls together true instances, even if they are misaligned. With the proposed clustering, aligned instances are closer to the mean embedding while misaligned are further away, because the contrastive loss pulls every pair no matter the similarity between the instances. In the clustering step, for the aligned pairs, modalities will converge better while misaligned pairs will stay apart.

C.2. Ablation of modalities.

We perform ablation experiments on the use of modalities in Table C. From these experiments we find audio information to be crucial in bridging the gap between video and text while learning a joint space across the three modalities. The improvement on MSR-VTT is not significant compared to Youcook2. We attribute this performance difference to the domain gap between the various datasets. Both HowTo100M and Youcook are based on instructional videos where the text modality has a strong correlation to the video and audio modalities. In HowTo100M, the text is based on ASR transcripts. In Youcook2 and MSR-VTT, the query texts are hand-annotated captions. While Youcook2 captions describe single cooking steps, MSR-VTT captions are general descriptions of the scene, with captions. These captions are

Method	UCF-101 Top-1	UCF-101 Top-5	HMDB Top-1	HMDB Top-5
Brattoli *et al*. [6]	37.6	62.5	26.9	49.8
MCN (ours)	33.0	62.3	20.9	48.4
MCN-actions (ours)	33.9	63.7	22.5	51.5

Table D: Zero-shot action recognition performance on the UCF-101 and HMDB datasets. MCN-actions is the MCN method, which has been "fine-tuned" on a subset of the HowTo100M dataset which contains action-related videos.

Method	Mod	Model	FT	YouCook2 R@1	YouCook2 R@5	YouCook2 R@10	YouCook2 Median R
Random		-	-	0.03	0.15	0.3	1678
Miech [15]	VT	R152+RX101	Y	8.2	24.5	35.3	24
MCN (ours)	VT	R152+RX101	Y	11.3	28.2	38.4	20
MCN (ours)	VAT	R152+RX101	Y	**28.2**	**53.0**	**63.7**	**5**

Table E: Comparison of text-to-video retrieval systems on finetune setting. FT indicates if it is finetuned on the downstream dataset.

Cluster size k	YouCook2 R@1	YouCook2 R@5	YouCook2 R@10	YouCook2 Median R	MSRVTT R@1	MSRVTT R@5	MSRVTT R@10	MSRVTT Median R
64	17.8	34.7	43.4	17	10.1	25.3	34.1	27
128	17.3	34.8	44.2	19	**10.5**	24.5	33.5	29
256	18.1	**35.5**	**45.2**	**16**	**10.5**	25.2	33.8	27
512	**18.3**	35.3	44.4	19	10.4	24.6	33.5	26.5
1024	17.9	34.6	43.5	17	9.4	**25.8**	**34.6**	**25**

Table F: Comparison of text-to-video retrieval systems on different number of cluster size in K-means

often not close to instructional ASR and also less related to what is being said in the audio.

C.3. Zero-Shot Action Recognition

We also test our method's performance for the downstream task of zero-shot action recognition. For these experiments, we follow the evaluation protocol of [6] and test on the full UCF-101 and HMDB datasets. We present the top-1 and top-5 accuracies on both datasets in Table D. Although MCN is trained using instructional videos, we find that the joint video/text space it learns is sufficient for the task of zero-shot action recognition. Furthermore, our method can be further improved by training on action-related videos; by removing various video categories - 'food and entertaining', 'computers and electronics', 'cars and other vehicles', 'home and garden', and 'health' and training on a subset of the HowTo100M dataset, we find MCN is able to achieve state-of-the-art Top-5 accuracy on both datasets. The baseline, [6], is a method designed specifically for zero-shot action recognition and is trained using labeled action videos from Kinetics-700, leading to strong top-1 accuracy.

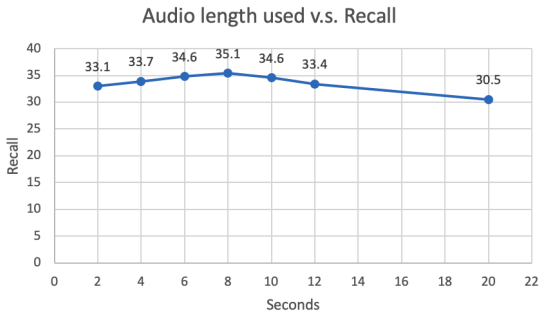

Figure B: Audio length used in inference on CrossTask.

C.4. CrossTask specific results.

We break down the consolidated performance result reported in the main paper on CrossTask and show results corresponding to each specific task in Table I. We observe that our model shows a very different yet often improved performance pattern, compared to the visual-only features used in [15] and [22]. We attribute this behavior to varying levels of information provided by the audio modality in each setting.

C.5. Finetune results

We show our model's performance on the finetune setting in Table E, which means we also train on an additional training set provided by the Youcook [20] dataset. Although the finetune setting, which requires ground-truth labels, isn't our main focus, we obtain significant improvement over the current baseline.

C.6. Different number of clusters

Table F shows the results using different number of cluster sizes for K-means. The result shows similar performance across different datasets and evaluation metrics.

C.7. Audio length used in inference.

We test the audio length needed for effective inference performance on CrossTask. As shown in Fig B, we find that using 8 seconds (4 seconds before and after) of audio leads to the best results. Given that some steps are very short (less than 3 seconds), this result also shows that using very long audio segments can distract the model from predicting a correct localization step.

D. Qualitative analysis

To further understand the proposed MCN model's improved performance, we also perform a qualitative analysis

Method	Mod	Model	Dataset	Com	Time	TR	YouCook2 R@1	YouCook2 R@5	YouCook2 R@10	MSRVTT R@1	MSRVTT R@5	MSRVTT R@10
Random		-		-			0.03	0.15	0.3	0.01	0.05	0.1
Miech [15]	VT	R152+RX101	HT+ImNet+K400	1 V100	1 day	N	6.1	17.3	24.8	7.2	19.2	28.0
MDR [3]	VT	R152+RX101	HT+ImNet+K400	1 V100	1 day	N	-	-	-	8.0	21.3	29.3
MIL-NCE* [14]	VT	R152+RX101	HT+ImNet+K400	4 V100	2 days	N	8.1	23.3	32.3	8.4	23.2	32.4
MCN (ours)	VAT	R152+RX101	HT+ImNet+K400	4 V100	2 days	N	**18.1**	35.5	45.2	**10.5**	**25.2**	**33.8**
MDR [3]	VT	R152	HT+ImNet+K400	1 V100	1 day	N	-	-	-	8.4	22.0	30.4
ActBERT [21]	VT	R101+Res3D	HT+VG+K400			N	9.6	26.7	38.0	8.6	23.4	33.1
SSB [17]	VT	R(2+1)D-34+R152	HT	8 V100	1 day	N	-	-	-	8.7	23.0	31.1
MMV FAC [1]	VAT	TSM-50x2	HT+AS	32 TPU	3 days	Y	11.7	33.4	45.4	9.3	23.0	31.1
MIL-NCE [14]	VT	I3D-G	HT	64 TPU	3 days	Y	11.4	30.6	42.0	9.4	22.0	30.0
MIL-NCE [14]	VT	S3D-G	HT	64 TPU	3 days	Y	15.1	**38.0**	**51.2**	9.9	24.0	32.4

Table G: Comparison of text-to-video retrieval systems. Mod indicates modality used, where V: video, A: audio, T: text. HT: HowTo100M. VG: Visual Genome. AS: AudioSet. Com stands for computational resource. Time indicates the training time. TR indicates if a trainable backbone is used or not.

Method	Mod	Model	Dataset	Com	Time	TR	CrossTask Recall	CrossTask IOD	CrossTask IOU	MYT Recall	MYT IOD	MYT IOU
CrossTask [22]	VT	R152+I3D	CrossTask			N	22.4	-	-	-	-	-
CrossTask [22]	VT	R152+I3D	CrossTask			N	31.6	-	-	-	-	-
Mining: GRU [12]	VT	TSN	MiningYouTube			N	-	-	-	-	14.5	7.8
Mining: MLP [12]	VT	TSN	MiningYouTube			N	-	-	-	-	19.2	9.8
Miech [15]	VT	R152+RX101	HT+ImNet+K400	1 V100	1 day	N	33.6	26.6	17.5	15.0	17.2	11.4
MIL-NCE* [14]	VT	R152+RX101	HT+ImNet+K400	4 V100	2 days	N	33.2	30.2	16.3	14.9	26.4	17.8
MCN (ours)	VAT	R152+RX101	HT+ImNet+K400	4 V100	2 days	N	35.1	**33.6**	**22.2**	**18.1**	**32.0**	**23.1**
ActBERT [21]	VT	R101+Res3D	HT+K400			N	37.1	-	-	-	-	-
ActBERT [21]	VT	+ Faster R-CNN	HT+VG+K400			N	**41.4**	-	-	-	-	-
MIL-NCE [14]	VT	I3D-G	HT	64 TPU	3 days	Y	36.4	-	-	-	-	-
MIL-NCE [14]	VT	S3D-G	HT	64 TPU	3 days	Y	40.5	-	-	-	-	-

Table H: Evaluation of temporal action localization systems.

	Make Kimchi Rice	Pickle Cucumber	Make Banana Ice Cream	Grill Steak	Jack Up Car	Make Jello Shots	Change Tire	Make Lemonade	Add Oil to Car	Make Latte	Build Shelves	Make Taco Salad	Make French Toast	Make Irish Coffee	Make Strawberry Cake	Make Pancakes	Make Meringue	Make Fish Curry	Average
CrossTask [22]	13.3	18.0	23.4	23.1	16.9	16.5	30.7	21.6	4.6	19.5	35.3	10.0	32.3	13.8	29.5	37.6	43.0	13.3	22.4
Supervised [22]	19.1	25.3	38.0	37.5	25.7	28.2	**54.3**	25.8	18.3	31.2	**47.7**	12.0	**39.5**	23.4	30.9	41.1	**53.4**	17.3	31.6
Miech *et al.* [15]	**33.5**	27.1	36.6	**37.9**	24.1	**35.6**	32.7	35.1	**30.7**	28.5	43.2	19.8	34.7	33.6	**40.4**	41.6	41.9	27.4	33.6
MCN	25.5	**31.1**	**39.7**	32.7	**35.4**	36.8	29.0	**40.0**	28.4	**33.8**	45.7	**27.5**	36.1	**34.9**	39.6	**42.6**	43.0	**29.1**	**35.1**

Table I: **Action step localization** results on CrossTask.

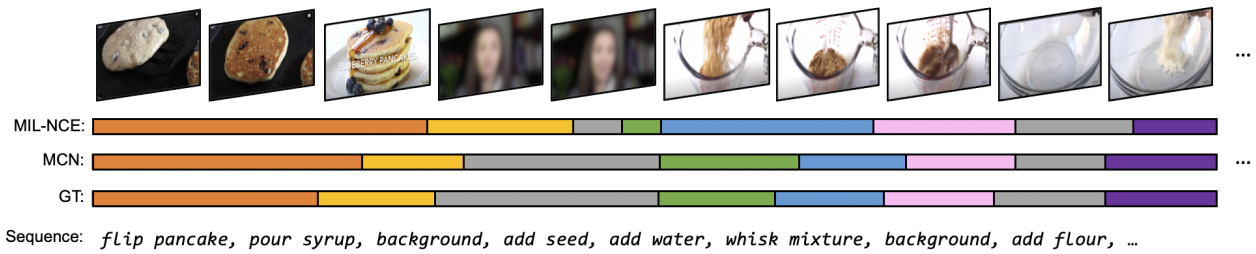

Figure C: Temporal action localization example from the first minute of the video "Vegan Blueberry Quinoa Pancakes" in the MiningYouTube dataset. Given the video and the action step sequence, the goal is to align the step temporal boundaries.

Figure D: Text-to-video retrieval examples. The retrieved video clips show a similar pattern.

with the model's temporal action localization results on the MiningYoutube task. One interesting observation is shown in Figure C. We observed that our model performs well in distinguishing action steps from the background scenes. We attribute this improvement to the proposed clustering component, which we observe has separated the background frames from various action classes. Background class instances are often placed as outliers with respect to the various action step clusters. In Figure D, we show more examples on text-to-video retrieval. The retrieved video segments show similar semantics.

References

[1] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelovic, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *NeurIPS*, 2020. 3, 5

[2] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020. 1, 2

[3] Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. Noise estimation using density estimation for self-supervised multimodal learning. In *AAAI*, 2021. 5

[4] Yuki Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*, 2020. 1

[5] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020. 2

[6] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *CVPR*, 2020. 4

[7] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 2

[8] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2, 3

[9] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013. 2

[10] Lawrence Hubert and Phipps Arabie. Comparing partitions. In *booktitle of classification*, volume 2, pages 193–218. Springer, 1985. 1

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[12] Hilde Kuehne, Ahsan Iqbal, Alexander Richard, and Juergen Gall. Mining youtube-a dataset for learning fine-grained action concepts from webly supervised video data. In *CVPR*, 2019. 1, 5

[13] Harold W Kuhn. The hungarian method for the assignment problem. In *Naval research logistics quarterly*, volume 2, pages 83–97. Wiley Online Library, 1955. 1

[14] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 2, 3, 5

[15] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 2, 3, 4, 5

[16] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 1

[17] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, João Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *ICLR*, 2021. 5

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. In *International booktitle of computer vision*, volume 115, pages 211–252. Springer, 2015. 2

[19] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. In *booktitle of machine learning research*, volume 3, pages 583–617, 2002. 1

[20] Luowei Zhou, Xu Chenliang, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 4

[21] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, pages 8746–8755, 2020. 3, 5

[22] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019. 1, 4, 5