ACAV100M: Automatic Curation of Large-Scale Datasets for Audio-Visual Video Representation Learning

Sangho Lee,* Jiwan Chung,* Youngjae Yu, Gunhee Kim
Seoul National University

Thomas Breuel, Gal Chechik
NVIDIA Research

Yale Song
Microsoft Research

https://acav100m.github.io

Abstract

The natural association between visual observations and their corresponding sound provides powerful self-supervisory signals for learning video representations, which makes the ever-growing amount of online videos an attractive source of training data. However, large portions of online videos contain irrelevant audio-visual signals because of edited/overdubbed audio, and models trained on such uncurated videos have shown to learn suboptimal representations. Therefore, existing self-supervised approaches rely on datasets with predetermined taxonomies of semantic concepts, where there is a high chance of audio-visual correspondence. Unfortunately, constructing such datasets require labor intensive manual annotation and/or verification, which severely limits the utility of online videos for large-scale learning. In this work, we present an automatic dataset curation approach based on subset optimization where the objective is to maximize the mutual information between audio and visual channels in videos. We demonstrate that our approach finds videos with high audio-visual correspondence and show that self-supervised models trained on our data achieve competitive performances compared to models trained on existing manually curated datasets. The most significant benefit of our approach is scalability: We release ACAV100M that contains 100 million videos with high audio-visual correspondence, ideal for self-supervised video representation learning.

1. Introduction

Our long-term objective is learning to recognize objects, actions, and sound in videos without the need for manual ground-truth labels. This is not only a theoretically interesting problem, since it mimics the development of au-
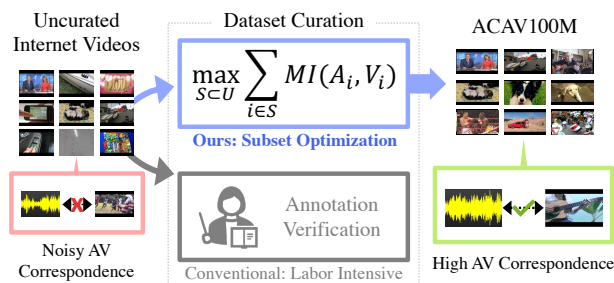
---
*Equal Contribution

Figure 1. We address the challenge of constructing a large-scale audio-visual dataset from uncurated Internet videos without relying on manual annotation or verification. We solve a constrained optimization problem that finds a subset maximizing the mutual information between audio and visual signals in videos. The result is a new 100M video dataset with high audio-visual correspondence, ideal for self-supervised video representation learning.

ditory and visual perception by infants [21], it is also of immense practical importance, since accurate manual labeling of audio-visual data is impractical. Compared to self-supervised learning on static images [50, 28, 25, 12], audio-visual inputs pose additional challenges: large portions of a video may contain no relevant information, and auditory and visual inputs may not always be in correspondence. Consequently, existing self-supervised methods on audio-visual data either start with datasets for which there is a high probability of audio-visual correspondence, or they learn audio-visual properties corresponding only to short-term statistical regularities. The necessary datasets are usually manually created or rely on domain-specific properties (e.g., [9, 20] and below). If we want to carry out self-supervised learning on full length (minutes, hours) of video without manually generating and/or selecting video clips, we need automated ways of curating such collections of audio/video clips from diverse collections of full length video.

We consider self-supervised learning from unlabeled

videos as a two-step process: (1) an automatic dataset curation process that generates short, relevant clips with useful self-supervisory signals, e.g., audio-visual correspondence, and (2) a self-supervised learning approach that operates on the collection of short clips. This paper focuses on step (1) and *not* on step (2), providing an automated way of taking a collection of general or domain-specific videos of arbitrary length and reducing it to a collection of shorter clips containing a high portion of relevant audio-video correspondences. The output of this step is a dataset, which can be used as input to existing self-supervised algorithms on audio-visual data [34, 3, 54], as well as the development of novel self-supervised techniques.

To achieve step (1), we assume access to a large collection of unconstrained videos and solve a subset selection problem with an information-theoretic measure of audio-visual correspondence as a selection criterion. Specifically, we find a subset that maximizes mutual information (MI) between audio and visual channels of videos. This is a necessary condition for self-supervised learning approaches that rely on audio-visual correspondence [17]. The main technical challenge we address is how to *efficiently* measure the audio-visual MI and find a subset that maximizes the MI in a scalable manner. Given that video processing is notoriously compute and storage intensive, we put a particular emphasis on scalability, i.e., we want an approach that can easily handle hundreds of millions of video clips.

MI estimation has a long history of research [53, 35], including the recent self-supervised approaches [50, 28, 12] that use noise contrastive estimation [23] as the learning objective. While it is tempting to use such approaches to estimate MI in our work, we quickly encounter the "chicken-and-egg" problem: to obtain such models for estimating audio-visual MI, we need a training dataset where we can reliably construct positive pairs with a high probability of audio-visual correspondence; but that is what we are set out to find in the first place! One might think that randomly chosen videos from the Internet could be sufficient, but this has shown to produce suboptimal representations [3]; our empirical results also show that self-supervised models indeed suffer from noisy real-world audio-visual correspondences.

In this work, we turn to a clustering-based solution that estimates the MI by measuring the agreement between two partitions of data [42, 67]. To circumvent the "chicken-and-egg" issue, we use off-the-shelf models as feature extractors and obtain multiple audio and visual clusters to estimate the MI. The use of off-the-shelf models is a standard practice in video dataset generation. Unlike existing approaches that use them as concept classifiers [8, 1, 43, 47, 11], here we use them as generic feature extractors. To avoid estimating the MI based on a restricted set of concepts the off-the-shelf models are trained on, we perform clustering over features computed across multiple layers (instead of just the penul-timate layers), which has been shown to provide general feature descriptors not tied to specific concepts [76].

To make our approach scalable, we avoid using memory-heavy components such as the Lloyd's algorithm [52] and instead use SGD [7] to perform K-means clustering. Further, we approximately solve the subset maximization objective with a mini-batch greedy method [13]. Through controlled experiments with ground-truth and noisy real-world correspondences, we show that our clustering-based approach is more robust to the real-world correspondence patterns, leading to superior empirical performances than the contrastive MI estimation approaches.

We demonstrate our approach on a large collection of videos at an unprecedented scale: We process 140 million full-length videos (total duration 1,030 years) and produce a dataset of 100 million 10-second clips (31 years) with high audio-visual correspondence. We call this dataset ACAV100M (short for **a**utomatically **c**urated **a**udio-**v**isual dataset of **100M** videos). It is two orders of magnitude larger than the current largest video dataset used in the audio-visual learning literature, i.e., AudioSet [20] (8 months), and twice as large as the largest video dataset in the literature, i.e., HowTo100M [44] (15 years).

To evaluate the utility of our approach in self-supervised audio-visual representation learning, we produce datasets at varying scales and compare them with existing datasets of similar sizes that are frequently used in the audio-visual learning literature, i.e., Kinetics-Sounds [4] at 20K-scale, VGG-Sound [11] at 200K-scale, and AudioSet [20] at 2M-scale. Under the linear evaluation protocol with three downstream datasets, UCF101 [62], ESC-50 [56], and Kinetics-Sounds [4], we demonstrate that models pretrained on our datasets perform competitively or better than the ones pretrained on the baseline datasets, which were constructed with careful annotation or manual verification.

To summarize, our main contributions are: 1) We propose an information-theoretic subset optimization approach to finding a large-scale video dataset with a high portion of relevant audio-visual correspondences. 2) We evaluate different components of our pipeline via controlled experiments using both the ground-truth and the noisy real-world correspondence patterns. 3) We release ACAV100M, a large-scale open-domain dataset of 100M videos for future research in audio-visual representation learning.

2. Related Work

Large-Scale Data Curation. Several different types of audio-visual video datasets have been collected: (1) manually labeled, e.g., AudioSet [20], AVE [65], (2) domain specific, e.g., AVA ActiveSpeaker [58], AVA Speech [10], Greatest Hits [51], FAIR-Play [19], YouTube-ASMR-300K [75], and (3) unlabeled, unrestricted collections from consumer video sites, e.g., Flickr-SoundNet [5, 4].

AudioSet [20] contains about 2M clips corresponding to audio events retrieved from YouTube by keyword search; human raters verified the presence of audio events in the candidate videos. *Moments in Time* [46] contains over one million clips of diverse visual and auditory events; video clips were selected using keywords (verbs) and manually reviewed for high correspondence between the clips and the keywords. *HowTo100M* [44] contains 136M clips segmented from 1.22M narrated instructional web videos retrieved by text search from YouTube, with an additional filtering step based on metadata. *Web Videos and Text* (WVT) [64] contains 70M clips obtained by searching the web with keywords based on the Kinetics-700 [9] categories and retaining both the video and the associated text. Chen *et al.* [11] created a dataset of 200K clips for audio-visual research; clips were originally obtained by keyword search on YouTube and frames were classified with pretrained visual classifiers. Since keywords and visual classes do not perfectly correspond, such correspondences needed to be manually reviewed and corrected on randomly sampled clips in an iterative and interactive process.

We are building systems for learning audio-visual correspondence on diverse, unrestricted inputs. This requires large amounts of training data, making manual collection and labeling costly and impractical. Unlike previous dataset curation processes that involve costly human intervention, we introduce an automatic and scalable data curation pipeline for large-scale audio-visual datasets.

Subset Selection. Our work focuses on data subset selection; extensive prior work exists in supervised [66, 72, 61, 71], unsupervised [24, 73], and active learning settings [39, 60]. Different criteria for subset selection have been explored in the literature. *Submodular functions* naturally model notions of information, diversity and coverage [70], and can be optimized efficiently using greedy algorithms [45, 48]. *Geometric criteria* like the coreset [2] aim to approximate geometric extent measures over a large dataset with a relatively small subset.

Mutual-information (MI) between input feature values and/or labels has been used successfully [22, 40, 63] as a probablistically motivated criterion. We propose to use MI as an objective function for subset selection and make the following two unique contributions: First, we use MI to measure audio-visual correspondence within videos by formulating MI between the audio and visual features. Second, we apply MI for the large-scale video dataset curation problem. In case of clustering-based MI estimation, we demonstrate that optimizing MI objective with a greedy algorithm is a practical solution for building a large-scale pipeline.

3. Data Collection Pipeline

Our pipeline consists of four steps: (i) acquiring raw videos from the web and filtering them based on metadata, (ii) segmenting the videos into clips and extracting features with pretrained extractors, (iii) estimating mutual information (MI) between audio and visual representations, and (iv) selecting a subset of clips that maximizes the MI.

3.1. Obtaining Candidate Videos

We crawl YouTube to download videos with a wide variety of topics. Unlike previous work that use a carefully curated set of keywords [11], which could inadvertently introduce bias, we aim for capturing the natural distribution of topics present in the website. To ensure the diversity in topics, cultures and languages, we create combinations of search queries with diverse sets of keywords, locations, events, categories, etc., to obtain an initial video list.

Before downloading videos, we process the search results using metadata (provided by YouTube API) to filter out potentially low quality / low audio-visual correspondence videos. We use the duration to exclude videos shorter than 30 seconds (to avoid low quality videos) and longer than 600 seconds (to avoid large storage costs). We also exclude videos that contain selected keywords (in either title or description) or from certain categories – i.e., gaming, animation, screencast, and music videos – because most videos exhibit non-natural scenes (computer graphics) and/or low audio-visual correspondence. Finally, we detect language from the titles and descriptions using fastText [31, 32] and keep the ones that constitute a cumulative ratio of 0.9, resulting in eight languages (English, Spanish, Portuguese, Russian, Japanese, French, German, and Korean).

The result is 140 million full-length videos with a total duration of 1,030 years (median: 198 seconds). To minimize the storage cost we download 360p resolution videos; this still consumes 1.8 petabytes of storage. Handling such large-scale data requires a carefully designed data pipeline. We discuss our modularized pipeline below.

3.2. Segmentation & Feature Extraction

Clip Segmentation. To avoid redundant clips, we extract up to three 10-second clips from each full-length video. We do this by detecting shot boundaries (using the `scdet` filter in FFmpeg) and computing pairwise clip similarities based on the MPEG-7 video signatures (using the `signature` filter in FFmpeg). We then select up to 3 clips that give the minimum total pairwise scores using local search [30]. This gives us about 300M clips.

Feature Extraction. To measure correspondence between audio and visual channels of the 300M clips, we need good feature representations. An ideal representation would capture a variety of important aspects from low-level details (e.g., texture and flow) to high-level concepts (e.g., semantic categories). However, such an oracle extractor is hard to obtain, and the sheer scale of data makes it impractical to learn optimal feature extractors end-to-end. Therefore,

we use the "off-the-shelf" pretrained models to extract features, i.e., SlowFast [15] pretrained on Kinetics-400 [33] and VGGish [27] pretrained on YouTube-8M [1] for visual and audio features, respectively.

3.3. Subset Selection via MI Maximization

Next, we select clips that exhibit strong correspondence between visual and audio channels. To this end, we estimate the mutual information (MI) between audio and visual signals. Computing the exact MI is infeasible because it requires estimating the joint distribution of high dimensional variables, but several approximate solutions do exist [68]. Here we implement and compare two approaches: a noise-contrastive estimator (NCE) [23], which measures MI in a continuous feature space, and a clustering-based estimator that computes MI in a discrete space via vector quantization. The former estimates MI for each video clip, while the latter estimates MI for a set of video clips. As we show later in our experiments, we find the clustering-based MI estimator to be more robust to real-world noise.

3.3.1 NCE-based MI Estimation

Contrastive approaches have become a popular way of estimating MI between different views of the data [50, 28]. We add linear projection heads over the precomputed audio/visual features and train them using the contrastive loss [12]. From a mini-batch $\{(v_i, a_i)\}_{i=1}^{N_b}$ where v_i and a_i are visual and audio features, respectively, we minimize

$$l(v_i, a_i) = -\log \frac{\exp(S(\mathbf{z}_i^v, \mathbf{z}_i^a)/\tau)}{\sum_{j=1}^{N_b} \exp(S(\mathbf{z}_i^v, \mathbf{z}_j^a)/\tau)}, \quad (1)$$

where $\mathbf{z}_i^v$ and $\mathbf{z}_i^a$ are embeddings from the linear projection heads, $S(\cdot, \cdot)$ measures the cosine similarity, and τ is a temperature term (we set $\tau = 0.1$). For each mini-batch we compute $l(v_i, a_i)$ and $l(a_i, v_i)$ to make the loss symmetric.

Once trained, we can directly use $S(\mathbf{z}^v, \mathbf{z}^a)$ to estimate audio-visual MI and find a subset by taking the top N candidates from a ranked list of video clips.

3.3.2 Clustering-based MI Estimation

MI Estimation. Clustering is one of the classical ways of estimating MI [42, 67]. Given two partitions of a dataset $\mathbf{X}$ w.r.t. audio and visual features, $\mathcal{A} = \{\mathbf{A}_1, \cdots, \mathbf{A}_{|\mathcal{A}|}\}$ and $\mathcal{V} = \{\mathbf{V}_1, \cdots, \mathbf{V}_{|\mathcal{V}|}\}$, we estimate their MI as:

$$\text{MI}(\mathcal{A}, \mathcal{V}) = \sum_{i=1}^{|\mathcal{A}|} \sum_{j=1}^{|\mathcal{V}|} \frac{|\mathbf{A}_i \cap \mathbf{V}_j|}{|\mathbf{X}|} \log \frac{|\mathbf{X}||\mathbf{A}_i \cap \mathbf{V}_j|}{|\mathbf{A}_i||\mathbf{V}_j|}. \quad (2)$$

This formulation estimates MI in a discrete (vector-quantized) space induced by clustering, and thus the quality

---

Algorithm 1: Batch Greedy Subset Selection

Input: initial dataset $\mathbf{D}$, MI estimator F, target
  subset size M, batch size b, selection size s
Output: $\mathbf{X} \subseteq \mathbf{D}, |\mathbf{X}| = M$
$\mathbf{X}_0 \leftarrow \emptyset, i \leftarrow 0$
while $|X_i| < M$ **do**
  Randomly sample $\mathbf{B} \subseteq \mathbf{D}\backslash\mathbf{X}_i, |\mathbf{B}| = b$
  $\mathbf{Y}_0 \leftarrow \emptyset, j \leftarrow 0$
  while $j < s$ **do**
    $x \leftarrow \text{argmax}_{x \in \mathbf{B}\backslash\mathbf{Y}_j} F(\mathbf{X}_i \cup \mathbf{Y}_j \cup \{x\})$
    $\mathbf{Y}_{j+1} \leftarrow \mathbf{Y}_j \cup \{x\}, j \leftarrow j + 1$
    if $|\mathbf{X}_i \cup \mathbf{Y}_j| = M$ **then** break
  end
  $\mathbf{X}_{i+1} \leftarrow \mathbf{X}_i \cup \mathbf{Y}_j, i \leftarrow i + 1$
end
$\mathbf{X} \leftarrow \mathbf{X}_i$
Return X

---

of clustering affects the quality of the estimator. A straightforward approach to obtaining $\mathcal{A}$ and $\mathcal{V}$ is to cluster videos using the output from the penultimate layers of the pretrained networks. However, this can introduce distributional bias specific to the datasets on which the networks are pretrained [76, 69]. To address this issue, we cluster samples over each output space induced by different layers of the networks. This allows the MI estimator to consider a wide range of abstract concepts, from low-level (such as textures) to high-level (such as object parts) [6].

Specifically, we use the feature spaces induced by the five convolutional blocks from each of the SlowFast and VGGish feature extractors. We then compute the average MI between *all pairs* of clusterings as our MI estimator. Let $\mathcal{CV}_{\mathbf{X}}^{(i)} = \{\mathbf{V}_1^{(i)}, \cdots, \mathbf{V}_{n_i}^{(i)}\}$ and $\mathcal{CA}_{\mathbf{X}}^{(i)} = \{\mathbf{A}_1^{(i)}, \cdots, \mathbf{A}_{m_i}^{(i)}\}$ denote the clustering results induced by the i-th convolutional block of the visual and audio feature extractors, respectively. We compute:

$$F(\mathbf{X}) = \sum_{(\mathcal{X}, \mathcal{Y}) \in \mathcal{C}_{\mathbf{X}}} \frac{\text{MI}(\mathcal{X}, \mathcal{Y})}{_{10}C_2}, \quad (3)$$

where $\mathcal{C}_{\mathbf{X}}$ denotes the combination of two elements from $\{\mathcal{CV}_{\mathbf{X}}^{(i)}\}_{i=1}^5 \cup \{\mathcal{CA}_{\mathbf{X}}^{(j)}\}_{j=1}^5$ and $_{10}C_2$ denotes the number of 2-combinations out of 10 elements, which equals to 45. This computes MI between layers from both within and across the extractors of different modalities (referred to as *combination* pairing scheme in Section 4.2).

Batch Greedy Subset Selection. Since the MI estimator $F(\cdot)$ is a function of $\mathbf{X}$, we can formulate an optimization problem where the goal is to find a subset $\mathbf{X}$ that maximizes $F(\mathbf{X})$. In general, finding a global solution to problems such as ours is NP-hard and thus greedy heuristic solutions are used instead [49]. However, they typically select one

sample in each iteration and re-evaluate the goodness function, e.g., $F(\cdot)$, on all the remaining candidates. This introduces a challenge to our setting because the time complexity is quadratic to the size of the population; this is clearly not scalable to 300 million instances.

Therefore, we approximate the typical greedy solution using the batch greedy algorithm [13], as shown in Algorithm 1. It randomly samples a batch $\mathbf{B}$ from the remaining pool of candidates, and searches for the next element to be included in the active solution set only within $\mathbf{B}$. This batch trick reduces the time complexity down to linear, i.e., $O(N \times |\mathbf{B}|)$, where N is the size of the input dataset. We demonstrate the efficacy of the algorithm in Section 4.

Stochastic Clustering. One missing piece in this pipeline is an *efficient* clustering algorithm scalable to hundreds of millions of instances. The most popular choice among various clustering methods is K-means clustering [74], which is a special case of mixture density estimation for isotropic normal and other densities. Typically, an expectation-maximization (EM) algorithm, such as Lloyd's [52], is used to find the cluster centers. Such algorithms require repeated computation of the distances of all samples from all k cluster centers, followed by cluster assignment, until convergence. Lloyd's algorithm updates cluster centers only after each pass through the entire dataset. But for very large datasets (like ours), a small subset usually contains enough information to obtain good estimates of the cluster centers, meaning that EM-style algorithms tend to take (perhaps too) many epochs to converge.

There are different strategies for addressing this issue, including random sampling and subsetting, but a straightforward approach is to replace EM algorithm with an SGD [41, 7, 59]. In such an approach, for large datasets, convergence rate and final accuracy of the cluster centers are determined not by the total dataset size, but by the learning rate schedule. A straightforward SGD update rule is to compute the nearest cluster centers for each sample in a batch and then update the cluster centers using a convex combination of the cluster centers and their nearest samples, weighting the samples with a learning rate λ and the cluster centers with $(1 - \lambda)$. However, mixture density estimators in general suffer from the problem that adding mixture components with zero probability does not change the mixture density; in practice, this means EM and SGD-based algorithms may end up with cluster centers that stop receiving updates at some point during the optimization.

We address this problem by estimating the mixture component utilization rate as the ratio of the total number of updates to the cluster center divided by the total number of estimation steps, and reinitializing cluster centers when that probability falls below $(1/k)^2$. In Section 4.2, we demonstrate that our mini-batch SGD update shows comparable accuracy to batch update in correspondence retrieval tasks.

4. Evaluation on Correspondence Retrieval

We systematically evaluate different components of our pipeline with synthetic correspondence-retrieval tasks, where we generate corresponding and non-corresponding pairs using CIFAR-10 [36], MNIST [38] and FSDD [29]. In each correspondence retrieval task, the goal is to discover the known corresponding samples among the non-corresponding pairs. To show the generality of the findings, we also experiment with Kinetics-Sounds [4] which exhibit real-world audio-visual correspondence.

4.1. Experimental Setting

Datasets We construct five datasets where each instance is a pair of samples with different correspondence types.

1/2) CIFAR10-Rotation/Flip. We use images from five randomly selected categories to construct a "positive pair" set, and use the rest for a "negative pair" set. For the positive set, we create pairs of images by sampling two different images from the same category (e.g., two images of a bird), and apply a geometric transformation to one of them; we apply either a 90° CCW rotation (CIFAR10-Rotation) or a horizontal flip (CIFAR10-Flip). The negative set follows the same process but each pair contains images from different categories. We categorize this type of correspondence as "Natural Class Correspondence" because pairings are made over natural semantic categories.

3/4) MNIST-CIFAR10/FSDD. We use images from five digit categories to construct a positive set and use the rest for a negative set. Different from above, correspondence is defined via an arbitrary class-level mapping, e.g., "digit 0" images map to the "car" images in CIFAR-10 or "digit 0" audio samples in FSDD. We take samples from the same categories to construct the positive set and samples from different categories for the negative set. We call these "Arbitrary Class Correspondence" to differentiate from above.

5) Kinetics-Sounds. Unlike the above datasets where the correspondence is defined over class categories, here the correspondence is defined at the *sample* level, i.e., a positive set contains pairs of audio and visual channels of the same video, and a negative set contains randomly permuted pairs. We do not utilize class labels to construct the dataset.

Methods We compare our pipeline (both contrastive-based and clustering-based) to three ranking-based approaches. All the methods use the same precomputed features. For images, we use ResNet-50 [26] pretrained on ImageNet [14]. For videos, we use SlowFast [15] pretrained on Kinetics-400 [33] and VGGish [27] pretrained on YouTube-8M [1] for visual and audio features, respectively. For the ranking baselines, we apply PCA [55] to reduce the feature dimensionality to 64 and rank the instances based on three similarity metrics: inner product, cosine similarity, and (negative) l_2 distance. Because all our datasets have an

| Method | Natural Class Correspondence | | Arbitrary Class Correspondence | | Audio-Visual |
	CIFAR10-Rotation	CIFAR10-Flip	MNIST-CIFAR10	MNIST-FSDD	Kinetics-Sounds
Ranking-inner	87.872 ± 0.002	87.044 ± 0.001	63.076 ± 0.001	64.453 ± 0.003	52.558 ± 0.002
Ranking-cos	87.872 ± 0.002	87.044 ± 0.001	67.600 ± 0.002	61.893 ± 0.004	60.108 ± 0.001
Ranking-l_2	87.872 ± 0.002	87.044 ± 0.001	66.796 ± 0.001	62.933 ± 0.003	51.236 ± 0.001
Ours-Contrastive	$\mathbf{99.395} \pm 0.000$	$\mathbf{99.480} \pm 0.001$	73.252 ± 0.040	$\mathbf{73.733} \pm 0.027$	73.066 ± 0.036
Ours-Clustering	87.292 ± 0.014	87.248 ± 0.010	$\mathbf{77.224} \pm 0.009$	69.440 ± 0.049	$\mathbf{88.705} \pm 0.004$

Table 1. Correspondence retrieval results. We conduct a total of five runs and report the precision with the 99% confidence interval. We use the clustering pairing scheme which gives the highest score in each configuration: combination, except diagonal for Ranking-inner, Ranking-cos and Rank-l_2 on CIFAR10-Rotation and CIFAR10-Flip.

equal number of positive and negative instances, we simply select the top 50% instances as the retrieval result.

Protocol We split each dataset into train and test partitions of the same size. We conduct a total of five runs for each of the five datasets and report results on the test splits. We use train sets only for the contrastive estimator to train the projection heads. When constructing each dataset, we sample at most $n = 1000$ instances from each category of the source datasets. For the noise contrastive estimator, we train the linear projection heads for 100 epochs using the AMSGrad of Adam optimizer [57] with a learning rate of 2e-4. We randomly take one sample from each class to build a mini-batch for class-level correspondence datasets, and sample random $N_b = 10$ clips to build a mini-batch for the sample-level correspondence dataset. When applying our clustering-based method, we perform the SGD K-means clustering with the "ground-truth" number of centroids as the number of classes in each source dataset; we use the batch greedy algorithm with a batch size $b = 100$ and a selection size $s = 25$.

4.2. Ablation Results & Discussion

Table 1 shows that the two variants of our approach – contrastive and clustering – achieve overall higher precision rates than the ranking baselines. The contrastive approach performs well on the two datasets with the "natural class correspondence," conforming to the previous results that shows contrastive learning is robust to geometric transformations [12]. The clustering approach excels on Kinetics-Sounds that contains natural audio-visual correspondence, which is closer to our intended scenario. Therefore, we conduct various ablation studies on Kinetics-Sounds to validate different components of our clustering-based approach.

Multi-Layer Clustering. All the feature extractors that we use consist of five convolutional blocks. As discussed in Section 3.3.2, we cluster samples over each of the five output spaces to capture a wide range of abstract concepts. This raises a question: How should we combine audio-visual clusters for MI estimation? Table 2 compares the single-layer approaches to multi-layer approaches. Each of the single-layer approach estimates the audio-visual MI based on a single pair of clustering results. We can see that the

Layers	Method	Precision
Single	Layer1	50.820 ± 0.014
	Layer2	51.412 ± 0.011
	Layer3	52.659 ± 0.012
	Layer4	54.422 ± 0.012
	Layer5	58.418 ± 0.030
Multiple	Diagonal	71.450 ± 0.005
	Bipartite	76.969 ± 0.005
	Combination	$\mathbf{88.705} \pm 0.004$

Table 2. Correspondence retrieval results on Kinetics-Sounds with different clustering pairing schemes. We conduct a total of five runs and report the precision with the 99% confidence interval.

precision increases as we use clustering results from higher layers. However, all single-layer methods perform significantly worse than multi-layer variants.

We explore three options to select pairs of clusterings for MI estimation. `Diagonal` computes an average MI across all five single-layer scores (with L layers, this computes MI L times), `Bipartite` computes an average MI between all possible combinations of audio-visual clustering results (L^2 times), and `Combination` (**ours**) computes an average MI between all possible combinations of clustering results, regardless of modalities ($_{2L}C_2$ times). We observe that the performance increases with the number of connections as shown in the bottom rows of Table 2. This positive relationship suggests that the consensus between layers from the same extractor, as well as that across extractors, contributes to the clarity of correspondence signal. We further experimented with different layer weights for the `Combination` approach and found it to be robust to different weight distributions; we provide the results in the supplementary material.

Mini-Batch SGD K-means Clustering. We compared mini-batch SGD K-means [7] to the standard EM (Lloyd's) approach [52] and obtained very similar results on Kinetics-Sounds: 88.705 ± 0.004 (SGD) versus 88.732 ± 0.005 (EM). This shows that our SGD solution has negligible performance degradation while enjoying a significantly less memory requirement than the standard EM approach.

Batch Greedy Subset Selection. We explore how the use of mini-batches affects the quality of the selected sub-
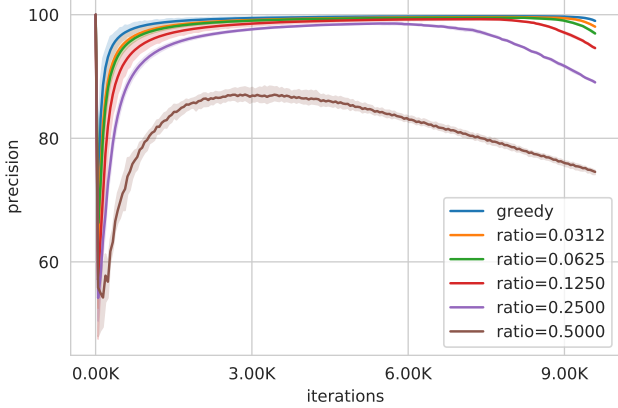
Figure 2. **Greedy vs. batch greedy algorithms** with varying selection-to-batch size ratios, s/b. The shaded regions show 99% confidence intervals obtained by five runs on Kinetics-Sounds. The batch greedy algorithm is robust when the ratio is $\leqslant 25\%$.
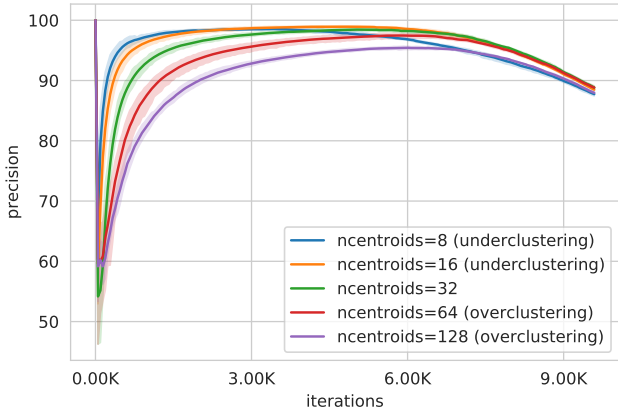

Figure 3. **Sensitivity analysis on the number of centroids**. We determine under/over-clustering based on the ground-truth number of class categories in Kinetics-Sounds ($c = 32$). The shaded regions show 99% confidence intervals over five runs.

sets. We compare the greedy algorithm and the batch greedy algorithm with a batch size $b = 160$ and varying selection sizes $s = \{5, 10, 20, 40, 80\}$. As shown in Figure 2, the performance gap between the greedy algorithm and the batch greedy algorithm is marginal (greedy: 98.970 vs. batch greedy with $(b, s) = (160, 5)$: 98.020), which validates our use of the batch greedy algorithm. While the batch size itself does not have a large impact on the subset quality, the ratio of selection size to batch size (s/b) highly affects the retrieval performance; the performance drops sharply as the ratio exceeds 0.25 in several (b, s) configurations. This is mainly dataset-dependent: by construction, there is a 50% chance that a sample will be a positive. We believe that the constructed dataset contains roughly 25% *easy positives*, i.e., videos with very high correspondence. When the selection ratio s/b does not exceed the easy positive ratio, the batch greedy algorithm finds those videos without introduc-

ing false positives, providing robustness. We found similar patterns with other ratios of $s/b > 25\%$.

Number of Centroids. We vary the number of centroids $k \in \{8, 16, 32, 64, 128\}$ to see how sensitive our approach is to the parameter. We apply the batch greedy algorithm with a batch size $b = 100$ and a selection size $s = 25$ on Kinetics-Sounds. Figure 3 shows that, although the final performance is similar across different number of centroids, they show different trends: underclustering ($k = \{8, 16\}$) shows high precision in early iterations while overclustering ($k = \{64, 128\}$) shows slower drop in the later stage.

5. Large-Scale Evaluation

We construct datasets at varying scales (20K, 200K, 2M) and compare them to existing datasets often used in the audio-visual learning literature: Kinetics-Sounds [4] (20K), VGG-Sound [11] (200K), and AudioSet [20] (2M). Note that all three datasets involve either human annotation [4, 20] or manual verification [11]. To demonstrate the scalable nature of our approach, we also generate datasets with 10M and 100M videos and evaluate their performance.

For the contrastive approach, we train linear projection heads on a batch size of 1024 from a randomly drawn set of 100M videos. Note that these additional videos are only used to train projection heads for MI estimation (Sec. 3.3.1), which is discarded once dataset curation is finished; all approaches use the same number of videos under the same evaluation protocol on all downstream tasks. We train the model for three epochs and rank the entire video set (300M) based on the cosine similarity [12]. We then take top $N \in \{20K, 200K, 2M\}$ ranked videos for the final dataset. For the clustering-based variant, we vary the number of clusters $C \in \{100, 200, 500, 1000, 2000\}$ for each size of the datasets.

5.1. Linear Evaluation on Downstream Tasks

To assess the quality of the datasets, we pretrain identical models on different datasets and evaluate their performance on downstream tasks. The idea is that if a model performed particularly better than the others, the dataset used to train that model must be superior to the other datasets. We pretrain audio-visual CNNs from scratch using the self-supervised objective of SimCLR [12]; we use 3D ResNet-50 [16] and ResNet-50 [26] as the visual and audio CNNs, respectively. We follow the linear evaluation protocol [12] by adding a linear classifier on top of the learned and frozen models. We test on three downstream tasks: visual action recognition on UCF101 [62], sound classification on ESC-50 [56], and audio-visual action recognition on Kinetics-Sounds [4] (we concatenate audio-visual features for the linear classifier). Note that the training procedures are identical for all the models except for the datasets used to train them. We report mean accuracy across the official splits of
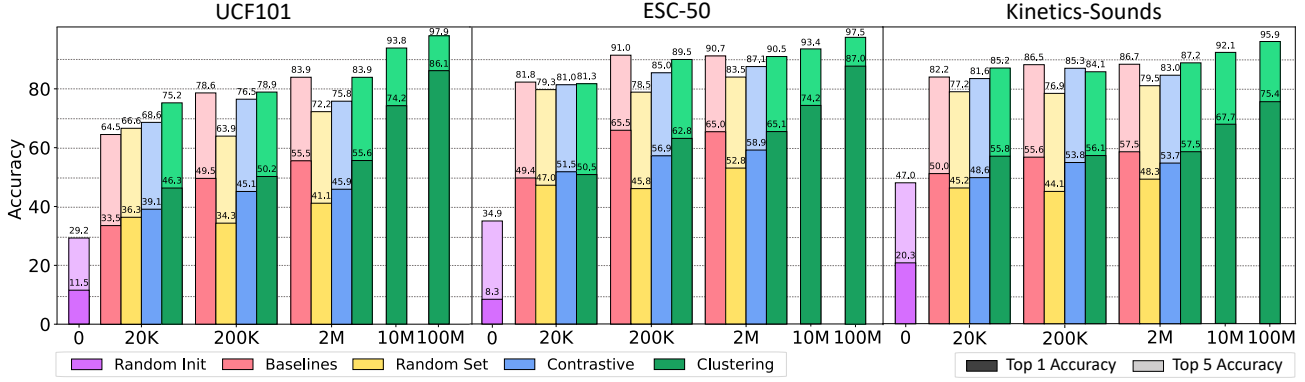
Figure 4. **Linear evaluation on downstream tasks**. The top-1/5 accuracy (%) of video classification on UCF101 [62], audio classification on ESC-50 [56] and audio-visual classification on Kinetics-Sounds (KS) [4]. We group the results by the downstream tasks and by the scale of the pretrain datasets. Baselines are Kinetics-Sounds [4] (20K), VGG-Sound [11] (200K), and AudioSet [20] (2M).

UCF101 and ESC-50. We provide details of these experimental settings in the supplementary material.

Figure 4 shows that models pretrained on our dataset (green bars) achieve similar, or even slightly better, performances compared to the baseline datasets (pink bars) at 20K, 200K, and 2M scales. The significant gap between ours vs. random set (yellow bars) shows the improvement does not come from the initial pool we crawl (the 300M set) but rather come from higher portion of audio-visual correspondence in the resulting dataset. Our clustering approach to MI estimation (green bars) generally outperforms the contrastive approach (blue bars), suggesting its robustness to noisy real-world audio-visual correspondences. Finally, we report the results obtained from 10M and 100M datasets produced with our clustering-based MI estimation module (we omit the baseline results at these scales due to computational reasons). The significant performance boost from the 10M and 100M models reaffirms the importance of large-scale training. Considering our data curation process does not involve human intervention (i.e., no manual annotation and verification) this is a promising result showing the potential for large-scale self-supervised learning: one can obtain datasets of arbitrary scales and develop self-supervised models by leveraging high portion of audio-visual correspondences provided in the datasets.

5.2. Human Evaluation

We conduct a user study to assess the perceived presence/absence of audio-visual correspondence in video clips. We compare clips from four datasets: AudioSet [20], VGG-Sound [11], ours with clustering (2M scale, 1K clusters), and random (drawn from the 300M set). We prepare 100 randomly sampled clips from each of these datasets, for a total of 400 clips. We recruit 12 participants and present each with 100 clips (25 clips per dataset), and ask them whether audio and visual are corresponding or not. This provides us with 3 votes per video (we provide the details

Dataset	Majority Vote (%)	Fleiss' Kappa
AudioSet	65.66	0.4385
VGG-Sound	84.00	0.4634
Ours (2M)	69.00	0.5110
Random	44.00	0.6112

Table 3. **Human evaluation results** assessing the perceived audio-visual correspondence in videos from different datasets.

of the questionnaire in the supplementary material).

Table 3 shows the majority voting accuracy and inter-rater agreement (measured by Fleiss' Kappa [18]). Every dataset has Fleiss' Kappa greater than 0.4, verifying the reliability of the accuracy statistics [37]. Ours significantly improves audio-visual correspondence over a random subset (69% vs. 44%), and is even rated slightly higher than AudioSet. The annotation process for AudioSet has focused on audio events so we suspect that several of videos do not contain *visible* sound sources. There is still a significant gap between ours and VGG-Sound; we note that our process finds audio-visual correspondence without relying on manual verification as was done in VGG-Sound.

6. Conclusion

This work complements existing line of research on self-supervised representation learning with three main contributions: i) proposing an automatic and scalable data collection pipeline for audio-visual representation learning, ii) demonstrating that the MI-based subset selection can retrieve correspondence in both artificial and practical settings, and iii) releasing a large-scale open-domain video dataset consisting of 100M clips curated with our pipeline.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8M: A Large-Scale Video Classification Benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. Geometric Approximation via Coresets. *Combinatorial and Computational Geometry*, 52:1–30, 2005.

[3] Humam Alwassel, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-Supervised Learning by Cross-Modal Audio-Video Clustering. *arXiv preprint arXiv:1911.12667*, 2019.

[4] Relja Arandjelovic and Andrew Zisserman. Look, Listen and Learn. In *ICCV*, 2017.

[5] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Sound-Net: Learning Sound Representations from Unlabeled Video. In *NeurIPS*, 2016.

[6] David Bau, Bolei Zhou, Aude Oliva, and Antonio Torralba. Interpreting Deep Visual Representations via Network Dissection. *PAMI*, 41(9):2131–2145, 2019.

[7] Leon Bottou and Yoshua Bengio. Convergence Properties of the K-Means Algorithms. In *NeurIPS*, 1995.

[8] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *CVPR*, 2015.

[9] João Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A Short Note on the Kinetics-700 Human Action Dataset. *arXiv preprint arXiv:1907.06987*, 2019.

[10] Sourish Chaudhuri, Joseph Roth, Daniel P. W. Ellis, Andrew Gallagher, Liat Kaver, Radhika Marvin, Caroline Pantofaru, Nathan Reale, Loretta Guarino Reid, Kevin Wilson, and Zhonghua Xi. AVA-Speech: A Densely Labeled Dataset of Speech Activity in Movies. In *Interspeech*, 2018.

[11] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. VGGSound: A Large-Scale Audio-Visual Dataset. In *ICASSP*, 2020.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, 2020.

[13] Yuxin Chen and Andreas Krause. Near-optimal Batch Mode Active Learning and Adaptive Submodular Optimization. *ICML*, 2013.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[15] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. In *ICCV*, 2019.

[16] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal Residual Networks for Video Action Recognition. In *NeurIPS*, 2016.

[17] John W Fisher and Trevor Darrell. Probabalistic Models and Informative Subspaces for Audiovisual Correspondence. In *ECCV*, 2002.

[18] Joseph L Fleiss. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin*, 76(5):378, 1971.

[19] Ruohan Gao and Kristen Grauman. 2.5D Visual Sound. In *CVPR*, 2019.

[20] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio Set: An Ontology and Human-Labeled Dataset for Audio Events. In *ICASSP*, 2017.

[21] Eleanor Jack Gibson. *Principles of Perceptual Learning and Development*. Appleton-Century-Crofts, 1969.

[22] Yuhong Guo. Active Instance Sampling via Matrix Partition. In *NeurIPS*, 2010.

[23] Michael Gutmann and Aapo Hyvärinen. Noise-Contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models. In *AISTATS*, 2010.

[24] Sariel Har-Peled and Soham Mazumdar. On Coresets for K-Means and K-Median Clustering. In *STOC*, 2004.

[25] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*, 2020.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[27] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. CNN Architectures for Large-Scale Audio Classification. In *ICASSP*, 2017.

[28] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning Deep Representations by Mutual Information Estimation and Maximization. In *ICLR*, 2019.

[29] Zohar Jackson, César Souza, Jason Flaks, Yuxin Pan, Hereman Nicolas, and Adhish Thite. Free Spoken Digit Dataset: v1.0.8, Aug. 2018.

[30] David S Johnson, Christos H Papadimitriou, and Mihalis Yannakakis. How Easy Is Local Search? *Journal of computer and system sciences*, 37(1):79–100, 1988.

[31] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. FastText.zip: Compressing Text Classification Models. *arXiv preprint arXiv:1612.03651*, 2016.

[32] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of Tricks for Efficient Text Classification. In *EACL*, 2017.

[33] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The Kinetics Human Action Video Dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[34] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative Learning of Audio and Video Models from Self-Supervised Synchronization. In *NeurIPS*, 2018.

[35] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating Mutual Information. *Physical review E*, 69(6), 2004.

[36] Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009.

[37] J Richard Landis and Gary G Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, pages 159–174, 1977.

[38] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[39] David D Lewis and William A Gale. A Sequential Algorithm for Training Text Classifiers. In *SIGIR*, 1994.

[40] Xin Li and Yuhong Guo. Adaptive Active Learning for Image Classification. In *CVPR*, 2013.

[41] Thomas Martinetz and Klaus Schulten. A "Neural-Gas" Network Learns Topologies. In *ICANN*, 1991.

[42] Marina Meilă. Comparing Clusterings—An Information Based Distance. *Journal of multivariate analysis*, 98(5), 2007.

[43] N Michele Merler, Khoi-Nguyen C. Mac, Dhiraj Joshi, Quoc-Bao Nguyen, Stephen Hammer, John Kent, Jinjun Xiong, Minh N. Do, John R. Smith, and Rogerio Schmidt Feris. Automatic Curation of Sports Highlights Using Multimodal Excitement Features. *IEEE Trans Multimedia*, 21(5), 2019.

[44] 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.

[45] Michel Minoux. Accelerated Greedy Algorithms for Maximizing Submodular Set Functions. In *Optimization techniques*, pages 234–243. Springer, 1978.

[46] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, and Carl Vondrick. Moments in Time Dataset: One Million Videos for Event Understanding. *PAMI*, pages 1–8, 2019.

[47] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Science and Language*, 2019.

[48] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An Analysis of Approximations for Maximizing Submodular Set Functions–I. *Mathematical Programming*, 14(1):265–294, 1978.

[49] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An Analysis of Approximations for Maximizing Submodular Set Functions—I. *Mathematical programming*, 14(1):265–294, 1978.

[50] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018.

[51] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually Indicated Sounds. In *CVPR*, 2016.

[52] Stuart P. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[53] Liam Paninski. Estimation of Entropy and Mutual Information. *Neural computation*, 15(6):1191–1253, 2003.

[54] Mandela Patrick, Yuki M Asano, Ruth Fong, João F Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal Self-Supervision from Generalized Data Transformations. *arXiv preprint arXiv:2003.04298*, 2020.

[55] Karl Pearson. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[56] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *ACM-MM*, 2015.

[57] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the Convergence of Adam and Beyond. In *ICLR*, 2018.

[58] Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew Gallagher, Liat Kaver, Sharadh Ramaswamy, Arkadiusz Stopczynski, Cordelia Schmid, Zhonghua Xi, and Caroline Pantofaru. AVA-ActiveSpeaker: An Audio-Visual Dataset for Active Speaker Detection. *arXiv preprint arXiv:1901.01342*, 2019.

[59] David Sculley. Web-Scale K-Means Clustering. In *WWW*, 2010.

[60] Burr Settles. Active Learning Literature Survey. *Science*, 10(3):237–304, 1995.

[61] Yusuke Shinohara. A Submodular Optimization Approach to Sentence Set Selection. In *ICASSP*, 2014.

[62] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv preprint arXiv:1212.0402*, 2012.

[63] Jamshid Sourati, Murat Akcakaya, Jennifer G Dy, Todd K Leen, and Deniz Erdogmus. Classification Active Learning Based on Mutual Information. *Entropy*, 18(2):51, 2016.

[64] Jonathan C Stroud, David A Ross, Chen Sun, Jia Deng, Rahul Sukthankar, and Cordelia Schmid. Learning Video Representations from Textual Web Supervision. *arXiv preprint arXiv:2007.14937*, 2020.

[65] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-Visual Event Localization in Unconstrained Videos. In *ECCV*, 2018.

[66] Ivor W Tsang, James T Kwok, and Pak-Ming Cheung. Core Vector Machines: Fast SVM Training on Very Large Data Sets. *Journal of Machine Learning Research*, 6(Apr):363–392, 2005.

[67] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *Journal of Machine Learning Research*, 11, 2010.

[68] Janett Walters-Williams and Yan Li. Estimation of Mutual Information: A Survey. In *RSKT*, 2009.

[69] Mei Wang and Weihong Deng. Deep Visual Domain Adaptation: A Survey. *Neurocomputing*, 312:135–153, 2018.

[70] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in Data Subset Selection and Active Learning. In *ICML*, 2015.

[71] Kai Wei, Yuzong Liu, Katrin Kirchhoff, Chris Bartels, and Jeff Bilmes. Submodular Subset Selection for Large-Scale Speech Training Data. In *ICASSP*, 2014.

[72] Kai Wei, Yuzong Liu, Katrin Kirchhoff, and Jeff Bilmes. Using Document Summarization Techniques for Speech Data Subset Selection. In *NAACL*, 2013.

[73] Kai Wei, Yuzong Liu, Katrin Kirchhoff, and Jeff Bilmes. Unsupervised Submodular Subset Selection for Speech Data. In *ICASSP*, 2014.

[74] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 Algorithms in Data Mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[75] Karren Yang, Bryan Russell, and Justin Salamon. Telling Left From Right: Learning Spatial Correspondence of Sight and Sound. In *CVPR*, 2020.

[76] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How Transferable are Features in Deep Neural Networks? In *NeurIPS*, 2014.