UNIVERSITY OF CALIFORNIA SAN DIEGO

Learning to see and hear without human supervision

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Intelligent Systems, Robotics & Control)

by

Pedro Miguel Maravilha Morgado

Committee in charge:

Professor Nuno Vasconcelos, Chair
Professor Nikolay A. Atanasov
Professor Gary Cottrell
Professor Kenneth Kreutz-Delgado
Professor Zhuowen Tu

2021

The dissertation of Pedro Miguel Maravilha Morgado is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

DEDICATION

To my parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This work would not be possible without the support of those around me.

First and foremost, I want to thank my PhD advisor Nuno Vasconcelos. Nuno took me as his PhD student, giving me the opportunity to pursue a career in research. I am genuinely grateful for the confidence he has put in me. I am fortunate to have him as my advisor and mentor. He taught me numerous lessons that reshaped my research mindset. He taught me to think about problems creatively and holistically. He has also taught me the value of perseverance in research, especially when progress is slow (the grass on the other side of the fence is not always greener!). Nuno was an exceptional mentor, and I am better because of him. I also want to thank the members of my doctoral committee, Professors Nikolay Atanasov, Gary Cottrell, Kenneth Kreutz-Delgado, and Zhuowen Tu, for their valuable feedback and support over the years. Gary Cottrell and Kenneth Kreutz-Delgado were also two of the best teachers I had as a student. Your lectures and dedication to your students is inspirational.

I am grateful to all mentors and collaborators I had over the years. Professors Margarida Silveira and Jorge Salvador Marques were my Master's thesis advisors at Instituto Superior Tecnico. Thank you for your patience and determination while introducing me to research for the first time. A shout-out to Professor Pedro Aguiar as well, which despite our limited interactions, planted the seed in my brain for pursuing doctoral studies. During graduate studies, I was fortunate to intern at Adobe with Oliver Wang and Timothy Langlois and Facebook with Ishan Misra. Your creativity, critical thinking, and research mindset have contributed enormously to my growth as a researcher. My initial Ph.D. research topic was built on the work of Mohammad Saberian and Jose Costa Pereira (two alumni from our lab). Their ground-breaking work was a source of motivation during the first few years of my PhD journey. I would also like to thank Jules Jaffes, Peter Franks, Eric Orenstein, Paul Roberts, and Kevin Le for tirelessly working on the Scripps plankton camera system. I will carry with me fond memories of this project, as well as the many sunsets spent at the Scripps TGIF with members of this crew. I was also fortunate

to collaborate with several younger PhD and Master students in the lab, Yi Li, Yunsheng Li, Gina Wu, John Ho, Amir Persekian, and Siddhant Jain. It has been a pleasure and an incredible learning experience for me to work with you. I hope the future will bring you the success you all deserve.

Beyond collaborators, many thanks to all lab mates, Jose Costa Pereira, Yingwei Li, Mandar Dixit, Zhaowei Cai, Bo Liu, Yunsheng Li, Yi Li, Pei Wang, John Ho, Gina Wu, Brandon Leung, Jiacheng Cheng, with whom I had the pleasure to spend countless hours. Thanks for taking the time to sit down with me, listen to me, share many meals, coffee and drinks with me, and be happy and frustrated with me. My PhD journey would not have been the same without you.

I also want to express my deepest gratitude to my family. To my grandma, Gracinda, whose unconditional encouragement makes me believe I can do anything. To my grandpa, Joaquim, whose tireless dedication to growing my brother and me, and his drive in the pursuit of a better life has inspired me beyond belief. To my Mom and Dad, which I owe life and much more. I wouldn't be here without your unconditional love, trust, and support. Thanks for teaching me the value of hard work and for your innumerable sacrifices. To my big brother for being my best friend and role model throughout all my life. Striving to follow your footsteps always propelled me to grow. To my close family, Paula, Armando, Bruno, Gisele, and my late granddad Arnaldo, I thank you for your love, nurture, and support. I have nothing but words of profound gratitude for raising me in the most supportive environment.

Last but not least, to Yea-Seul, my personal pâtissière and covid buddy, thank you for loving me. You have been a source of inspiration and joy in my life, and a source of kindness and comfort in stressful times. I am hopeful and excited for the next phase of our lives. Whatever life will bring us, I trust it will be worth it.

Chapter 2 is, in full, based on the material as it appears in the publication of "Audio-Visual Instance Discrimination with Cross-Modal Agreement", Pedro Morgado, Ishan Misra, Nuno

Vasconcelos, to appear in the Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. The dissertation author was the primary investigator and author of this material.

Chapter 3 is, in full, based on the material as it appears in the publication of "Robust Audio-Visual Instance Discrimination", Pedro Morgado, Nuno Vasconcelos, Ishan Misra, to appear in the Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. The dissertation author was the primary investigator and author of this material.

Chapter 4 is, in full, based on the material as it appears in the publication of "Learning Representations from Audio-Visual Spatial Alignment", Pedro Morgado, Yi Li, Nuno Vasconcelos, as appears in the Advances of Neural Information Processing Systems (NeurIPS), 2020.

Chapter 5 is, in full, based on the material as it appears in the publication of "Self-Supervised Generation of Spatial Audio for 360 Video", Pedro Morgado, Nuno Vasconcelos, Timothy Langlois and Oliver Wang, as appears in the Advances of Neural Information Processing Systems (NeurIPS), 2018. The dissertation author was the primary investigator and author of this material.

I gratefully acknowledge the support given by the Portuguese Foundation for Science and Technology (FCT) in the form of a Fellowship SFRH/BD/109135/2015. I also acknowledge support from the National Science Foundation (NSF) through several grants awarded to my advisor, which partially funded my work.

VITA

2011	B. S. in Electrical and Computer Engineering, Instituto Superior Técnico, Lisbon, Portugal
2012	M. S. in Electrical and Computer Engineering, Instituto Superior Técnico, Lisbon, Portugal
2021	Ph. D. in Electrical Engineering (Intelligent Systems, Robotics & Control), University of California San Diego

PUBLICATIONS

Pedro Morgado, Ishan Misra, Nuno Vasconcelos, "Robust Audio-Visual Instance Discrimination", *to appear in Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Pedro Morgado, Nuno Vasconcelos, Ishan Misra, "Audio-Visual Instance Discrimination with Cross-Modal Agreement", *to appear in Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Pedro Morgado, Yi Li, Nuno Vasconcelos, "Learning Representations from Audio-Visual Spatial Alignment", *in Advances of Neural Information Processing Systems (NeurIPS)*, 2020.

Pedro Morgado, Yunsheng Li, Jose Costa Pereira, Mohammad Saberian, Nuno Vasconcelos, "Deep Hashing with Hash-Consistent Large Margin Proxy Embeddings", *in International Journal of Computer Vision (IJCV)*, 2021.

Tz-Ying Wu, **Pedro Morgado**, Pei Wang, Chih-Hui Ho, and Nuno Vasconcelos, "Solving Long-tailed Recognition with Deep Realistic Taxonomic Classifier", *in Proceeding of European Conference on Computer Vision (ECCV)*, 2020.

Pedro Morgado and Nuno Vasconcelos, "NetTailor: Tuning the Architecture, Not Just the Weights", *in Proceeding of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Chih-Hui Ho, **Pedro Morgado**, Amir Persekian and Nuno Vasconcelos, "PIEs: Pose Invariant Embeddings", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Pedro Morgado, Nuno Vasconcelos, Timothy Langlois and Oliver Wang, "Self-Supervised Generation of Spatial Audio for 360° Video", *in Advances of Neural Information Processing Systems (NeurIPS)*, 2018.

Pedro Morgado and Nuno Vasconcelos, "Semantically Consistent Regularization for Zero-Shot Recognition", *in Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

ABSTRACT OF THE DISSERTATION

Learning to see and hear without human supervision

by

Pedro Miguel Maravilha Morgado

Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics & Control)

University of California San Diego, 2021

Professor Nuno Vasconcelos, Chair

Imagine the sound of waves. This sound may evoke the memories of days at the beach. A single sound serves as a bridge to connect multiple instances of a visual scene. It can group scenes that 'go together' and set apart the ones that do not. Co-occurring sensory signals can thus be used as a target to learn powerful representations for visual inputs without relying on costly human annotations.

In this thesis, I introduce effective self-supervised learning methods that curb the need for human supervision. I discuss several tasks that benefit from audio-visual learning, including representation learning for action and audio recognition, visually-driven sound source localization, and spatial sound generation. I introduce an effective contrastive learning framework that learns

audio-visual models by answering multiple-choice audio-visual association questions. I also discuss critical challenges we face when learning from audio supervision related to noisy audio-visual associations, and the lack of spatial grounding of sound signals in common videos.

Chapter 1

Introduction

Convolutional neural networks [106, 116] have become incredibly powerful over the last few years, leading to remarkable progress in predicting what objects are present in an image [78, 209, 189] and predicting each object's location [65, 66, 77, 24, 26], even at the pixel level [151, 125, 222, 188]. They can also predict remarkably well the location of different human body parts and their pose [52, 162, 25, 77]. These models, however, are trained using a supervised learning paradigm, where model parameters are trained to map each input (an image or video) to the desired output, which has been defined in advance by a human. To find the optimal parameters, deep learning models require large datasets of input-output relationships. For example, in human pose prediction, inputs are images of human subjects, and the desired outputs are the location of all body parts of all humans in each image. Hence, to collect such a dataset, human labelers had to annotate all body locations manually, which can be very time-consuming. This can be especially problematic in areas of computer vision that require expert annotations like medical imaging.

Furthermore, deep learning models often improve as the model size grows and are trained on larger datasets. While this can is good in practice, as it gives us the formula to build better models, it also increases our dependency on human annotations. Thus, this thesis focuses on making computer vision models more accessible by lowering the dependence on human annotations. Specifically, this thesis tackles the question of how to learn deep learning models without human annotations. This learning framework is often referred to as self-supervised learning.

1.1 Self-supervised learning

Self-supervised learning is an area of machine learning that seeks to learn models by solving pre-designed tasks, where the answer can be algorithmically defined without human input. Several tasks have been proposed over the years. One example shown here is to make

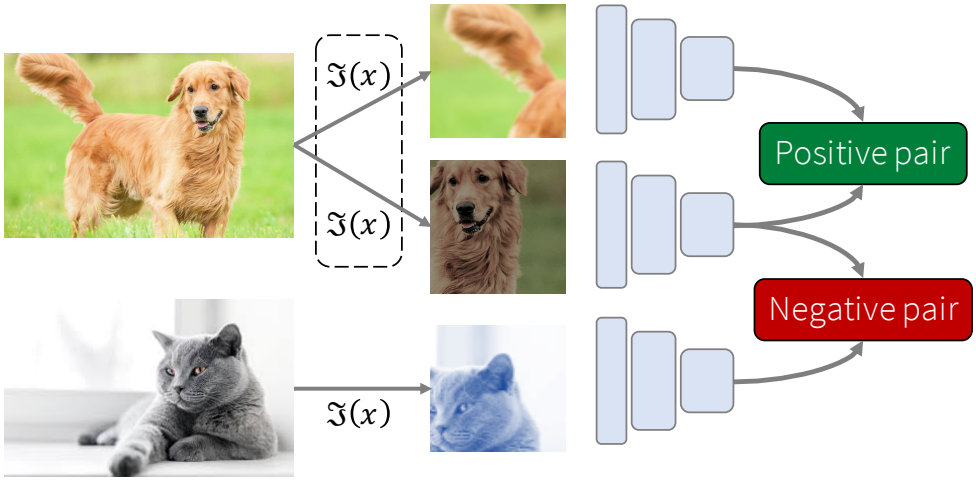


Figure 1.1: Contrastive learning

networks solve jigsaw puzzles [142]. The reasoning is, to predict the relative position of different tiles, the model needs to understand the contents of the underlying image. However, this and other tasks often don't work well in practice. Although models get very good at solving jigsaw puzzles, they are still weak when transferring the models to other tasks. In fact, designing reliable self-supervised learning tasks that generalize well to different tasks has proven hard, as most tasks cannot compete with supervised learning. Therefore, it is not surprising that the most significant progress in recent years has been the formulation of self-supervised learning as a supervised learning problem, where the labels come for free, by thinking of each image as a unique class. This task is called instance discrimination since the goal is to distinguish instances from each other [206, 47].

1.1.1 Instance discrimination

Instance discrimination is a self-supervised learning technique which tasks models with predicting whether two views belong to the same instance or not. Since datasets we can have a million instances or more, traditional classifiers can struggle due to the number of instances/classes. Thus, instead of traditional classification, models are learned with a technique called contrastive learning, exemplified in Figure 1.1. Given one instance, we can apply two data transformations to

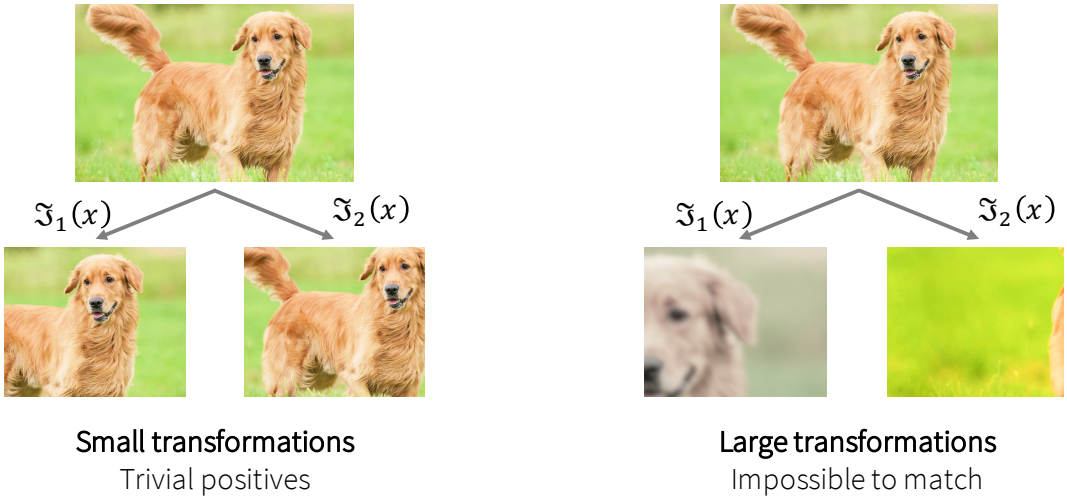


Figure 1.2: Visual transformations.

form a positive pair. We can also apply data transformations to two different instances to form negative pairs. With contrastive learning, the model is asked to distinguish positive from negative pairs.

At a high level, instance discrimination and contrastive learning encapsulate the main progress of self-supervised learning in computer vision. Note, however, that this framework relies heavily on data transformations to create positive pairs. If we apply small transformations, then finding positives is trivial, and the network does not need to learn refined representations. However, if we apply large transformations, then we change the input distribution significantly. That is, images no longer look like natural images (Figure 1.2). This thesis's contribution is to break the dependence of contrastive learning methods on data transformations to generate positive pairs. Instead of data transformations, we treat different data modalities, like audio and video, as a way to provide truly distinct but related views of the same instance (Figure 1.3). This allows us to push contrastive learning into the multimedia domain, enabling new multi-modal applications.

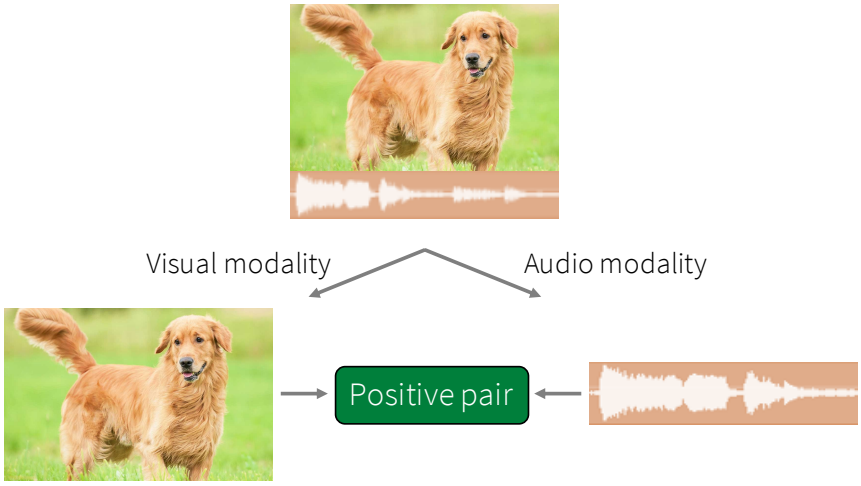


Figure 1.3: Audio-visual contrastive learning.

1.1.2 Challenges of audio-visual self-supervised learning

Audio-visual instance discrimination (Figure 1.3) can be used to learn strong models. However, while often correlated, audio and video sometimes are not informative of each other. Sometimes, people edit their videos to include background music or voice-over narrations. Other times, the video is naturally silent, *e.g.*, the video of a person doing yoga or meditating in silence. In these cases, establishing the correspondence between audio and video clips is impossible, and forcing these correspondences is detrimental for the learned representations. Also, contrastive learning relies on negatives randomly sampled from a dataset. As a result, negatives that are semantically similar to the base instance are often used as negatives. Since representation learning aims at learning an embedding function that yields similar outputs for similar instances, semantically similar negatives are also detrimental. Therefore, robust learning frameworks are necessary to tackle unreliable correspondences.

In addition to unreliable correspondences, another challenge is the fact that vanilla audio-visual instance discrimination only leverages global correspondences. In other words, models are asked to match audio signals to the whole video, instead of matching to the parts that contain audio sources. The lack of sound source localization can introduce ambiguities. For example,

consider the case of videos showing a car with the engine rumbling. While the association between the two modalities is good, most videos of cars also contain roads. Thus, from the model's perspective, which has no prior knowledge of the world, it is hard to distinguish what causes the rumbling sound – the car, the road, or both?

In the natural world, however, most animals possess spatial hearing, allowing them to perceive sound sources' spatial location. The spatial perception of audio signals can be replicated in 360 video content with spatial audio, as it captures all spatial information of both audio and visual signals. Therefore, the ability to localize sound sources can be used to differentiate between co-occurring objects, and as a result to learn better representations. I'll show that reasoning about the spatial location of sounds enhances representation learning.

Beyond representation learning, we also studied how well the spatial associations between audio and visual signals can be predicted, not just as a means to learn visual models but as an end in itself. In other words, we seek to generate the spatial components of audio given the 360-video content and mono (non-spatial) audio alone. Spatial audio generation can be a very practical task. Since many 360 video cameras only record mono, it would allow us to upgrade their audio into spatial sound and provide viewers with a more immersive experience. The question is, how can we accomplish this in a self-supervised manner, *i.e.*, without asking humans to manually label the spatial location of every sound in the video.

1.2 Contributions of the Thesis

In this thesis, we introduce a self-supervised framework for learning audio and visual representations from natural audio-visual associations, and tackle the main challenges mentioned above.

1.2.1 A strong framework for audio-visual instance discrimination

Chapter 2 presents a self-supervised learning approach to learn audiovisual representations from video and audio. Our method uses contrastive learning for cross-modal discrimination of video from audio and vice versa. We show that optimizing for cross-modal discrimination, rather than within-modal discrimination, is important to learn good representations from video and audio. With this simple but powerful insight, our method achieves state-of-the-art results when finetuned on action recognition tasks. While recent work in contrastive learning defines positive and negative samples as individual instances, we generalize this definition by exploring cross-modal agreement. We group together multiple instances as positives by measuring their similarity in both the video and the audio feature spaces. Cross-modal agreement creates better positive and negative sets, and allows us to calibrate visual similarities by seeking within-modal discrimination of positive instances.

1.2.2 Robust cross-modal instance discrimination

Chapter 3 presents a self-supervised learning method to learn audio and video representations. Prior work uses the natural correspondence between audio and video to define a standard cross-modal instance discrimination task, where a model is trained to match representations from the two modalities. However, the standard approach introduces two sources of training noise. First, audio-visual correspondences often produce faulty positives since the audio and video signals can be uninformative of each other. To limit the detrimental impact of faulty positives, we optimize a weighted contrastive learning loss, which down-weighs their contribution to the overall loss. Second, since self-supervised contrastive learning relies on random sampling of negative instances, instances that are semantically similar to the base instance are often used as faulty negatives. To alleviate the impact of faulty negatives, we propose to optimize an instance discrimination loss with a soft target distribution that estimates relationships between instances.

We validate our contributions through extensive experiments on action recognition tasks and show that they address the problems of audio-visual instance discrimination and improve transfer learning performance.

1.2.3 Towards increased spatial resolution of audio-visual associations

Chapter 4 introduces a novel self-supervised pretext task for learning representations from audio-visual content. Prior work on audio-visual representation learning leverages correspondences at the video level. Approaches based on audio-visual correspondence (AVC) predict whether audio and video clips originate from the same or different video instances. Audio-visual temporal synchronization (AVTS) further discriminates negative pairs originated from the same video instance but at different moments in time. While these approaches learn high-quality representations for downstream tasks such as action recognition, their training objectives disregard spatial cues naturally occurring in audio and visual signals. To learn from these spatial cues, we tasked a network to perform contrastive audio-visual spatial alignment of 360°video and spatial audio. The ability to perform spatial alignment is enhanced by reasoning over the full spatial content of the 360°video using a transformer architecture to combine representations from multiple viewpoints. The advantages of the proposed pretext task are demonstrated on a variety of audio and visual downstream tasks, including audio-visual correspondence, spatial alignment, action recognition and video semantic segmentation.

1.2.4 Self-supervised generation of spatial audio

Chapter 5 introduces an approach to convert mono audio recorded by a 360°video camera into spatial audio, a representation of the distribution of sound over the full viewing sphere. Spatial audio is an important component of immersive 360°video viewing, but spatial audio microphones are still rare in current 360°video production. Our system consists of end-to-end

trainable neural networks that separate individual sound sources and localize them on the viewing sphere, conditioned on multi-modal analysis of audio and 360°video frames. We introduce several datasets, including one filmed ourselves, and one collected in-the-wild from YouTube, consisting of 360°videos uploaded with spatial audio. During training, ground-truth spatial audio serves as self-supervision and a mixed down mono track forms the input to our network. Using our approach, we show that it is possible to infer the spatial location of sound sources based only on 360°video and a mono audio track.

Chapter 2

Audio-visual instance discrimination

2.1 Introduction

Imagine the sound of waves. This sound can evoke the memory of many scenes - a beach, a pond, a river, *etc*. A single sound serves as a bridge to connect multiple sceneries. It can group visual scenes that 'go together', and set apart the ones that do not. We leverage this property of freely occurring audio to learn video representations in a self-supervised manner.

A common technique [148, 150, 105, 8] is to setup a verification task that requires predicting if an input pair of video and audio is 'correct' or not. A correct pair is an 'in-sync' video and audio and an incorrect pair can be constructed by using 'out-of-sync' audio [105] or audio from a different video [8]. However, a task that uses a *single* pair at a time misses a key opportunity to reason about the data distribution at large.

In our work, we propose a contrastive learning framework to learn cross-modal representations in a self-supervised manner by contrasting video representations against *multiple* audios at once (and vice versa). We leverage recent advances [206, 70, 190, 147] in contrastive learning to setup a Audio-Visual Instance Discrimination (AVID) task that learns a cross-modal similarity metric by grouping video and audio *instances* that co-occur. We show that the cross-modal discrimination task, *i.e*., predicting which audio matches a video, is more powerful than the within-modal discrimination task, predicting which video clips are from the same video. With this insight, our technique learns powerful visual representations that improve upon prior self-supervised methods on action recognition benchmarks like UCF-101 [182] and HMDB-51 [109].

We further identify important limitations of the AVID task and propose improvements that allow us to 1) reason about *multiple* instances and 2) optimize for visual similarity rather than just cross-modal similarity. We use Cross-Modal Agreement (CMA) to group together videos with high similarity in video and audio spaces. This grouping allows us to directly relate multiple videos as being semantically similar, and thus directly optimize for visual similarity in addition to cross-modal similarity. We show that CMA can identify semantically related videos, and

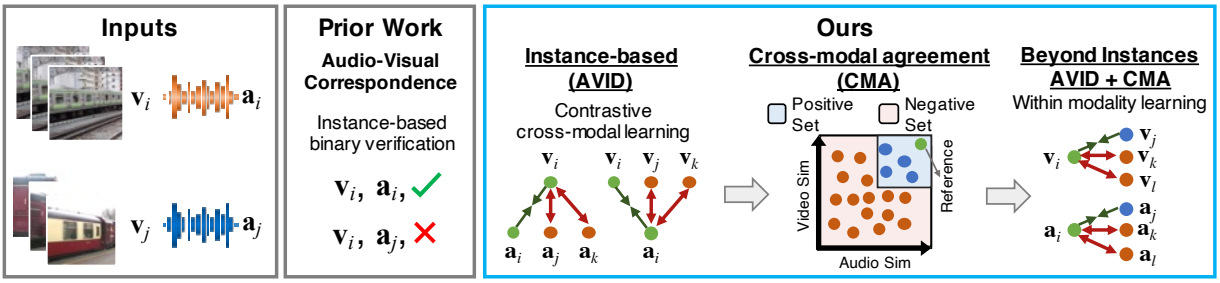


Figure 2.1: Comparison of audio-visual instance discrimination and cross-modal agreement to prior audio-video self-supervised methods.

that optimizing visual similarity among related videos significantly improves the learned visual representations. Specifically, CMA is shown to improve upon AVID on action recognition tasks such Kinetics [196], UCF-101 [182] and HMDB-51 [109] under both linear probing and full fine-tuning evaluation protocols.

2.2 Related work

Self-supervised learning is a well studied problem [145, 135, 130, 171, 39, 115]. Self-supervised methods often try to reconstruct the input data or impose constraints on the representation, such as sparsity [117, 145, 144], noise [195] or invariance [70, 166, 46, 91, 133, 22, 27, 28] to learn a useful and transferable feature representation. An emerging area of research uses the structural or domain-specific properties of visual data to algorithmically define 'pretext tasks'. Pretext tasks are generally not useful by themselves and are used as a proxy to learn semantic representations. They can use the spatial structure in images [44, 142, 64, 214], color [215, 114, 41], temporal information in videos [134, 118, 54, 93, 153, 143, 201, 72, 49] among other sources of 'self' or naturally available supervision. We propose an unsupervised learning technique that leverages the naturally available signal in video and audio alignment.

Representation Learning using Audio

Self-supervised learning can also make use of multiple modalities, rather than the visual data alone. As pointed out in [39, 97], co-occurring modalities such as audio can help learn powerful representations. For example, audio self-supervision has shown to be useful for sound source localization and separation [7, 175, 58, 60, 221, 220, 57], lip-speech synchronization [37] and visual representation learning [8, 105, 148] and audio spatialization [137].

Audio-Visual Correspondence (AVC)

AVC is a standard task [8, 105, 148, 7] used in audio-video cross-modal learning. This task tries to align the visual and audio inputs by solving a binary classification problem. However, most methods use only a single video and single audio at a time for learning. Thus, the model must reason about the distribution over multiple samples implicitly. In our work, we use a contrastive loss [70, 147, 190, 206] that opposes a large number of samples simultaneously. We show in §2.5 that our method performs better than recent methods that use AVC.

Contrastive Learning

Contrastive learning techniques use a contrastive loss [70] to learn representations either by predicting parts of the data [147, 80, 83], or discriminating between individual training instances [206, 46, 76, 133, 223, 53, 211, 84]. Contrastive learning has also been used for learning representations from video alone [72, 176]. Tian *et al*. [190] also use a contrastive approach, but propose to learn with a cross-modal objective applied to images and depth, video and flow. In contrast, our method learns visual representations using audio as cross-modal targets. Compared to [190], we present a new insight for audio-visual learning that optimizing cross-modal similarity is more beneficial than within-modal similarity. We also identify important limitations of cross-modal discrimination and present an approach that goes beyond instance discrimination by modeling Cross-Modal Agreement. This identifies groups of related videos and allows us

to optimize for within-modal similarity between related videos. The concurrently proposed [4] uses alternating optimization to find clusters in visual and audio feature spaces, *independently* and uses them to improve cross-modal features. While our CMA method bears a resemblance to theirs, we do not use alternating optimization and use *agreements* between the visual and audio representations to directly improve visual similarity rather than only cross-modal similarity. Finally, similar to our work, the concurrently proposed [74] also uses co-occurring modalities (optical flow and RGB) to expand the positive set. However, instead of mining positives based on an agreement between both modalities, [74] relies on the opposite modality alone.

Multi-view Learning

Multi-view learning aims to find common representations from multiple views of the same phenomenon, and has been widely used to provide learning signals in unsupervised and semi-supervised applications. Classical approaches can be broadly categorized in co-training procedures [21, 20, 200, 111, 127, 164, 74] that maximize the mutual agreement between views, multiple kernel learning procedures [112, 15, 104] which use kernels to model different views, and subspace learning procedures [43, 165] which seek to find the latent space that generates all views of the data. Multi-view data is an effective source of supervision for self-supervised representation learning. Examples include the motion and appearance of a video [190, 74], depth and appearance [216, 92], luminance and chrominance of an image [216, 190], or as in our work sound and video [8, 13, 150, 37].

2.3 Audio-Visual Instance Discrimination

We learn visual representations in a self-supervised manner from unconstrained video and audio by building upon recent advances in instance discrimination [206, 46, 129, 190] and contrastive learning [70, 69, 147].

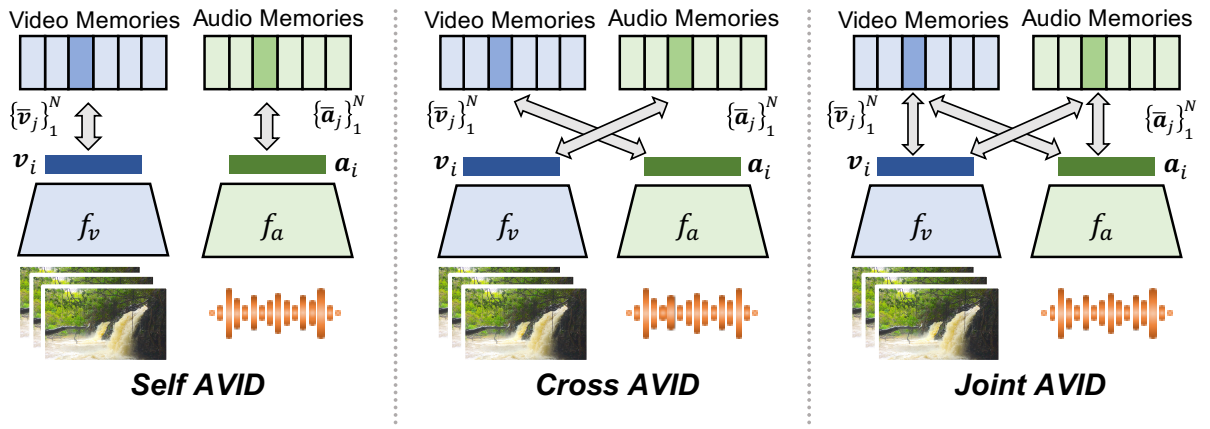


Figure 2.2: Variants of the AVID task.

2.3.1 Goal and Intuition.

Consider a dataset of N samples (instances) $\mathcal{S} = \{s_i\}_{i=1}^N$ where each instance s_i is a video s_i^v with a corresponding audio s_i^a. The goal of Audio-Visual Instance Discrimination (AVID) is to learn visual and audio representations $(\mathbf{v}_i, \mathbf{a}_i)$ from the training instances s_i. The learned representations are optimized for 'instance discrimination' [46, 206, 129], *i.e.*, must be discriminative of s_i itself as opposed to other instances s_j in the training set. Prior work [46, 206] shows that such a discriminative objective among instances learns semantic representations that capture similarities between the instances.

To accomplish this, two neural networks extract unit norm feature vectors $\mathbf{v}_i = f_v(s_i^v)$ and $\mathbf{a}_i = f_a(s_i^a)$ from the video and audio independently. Slow moving (exponential moving average) representations for both video and audio features $\{(\bar{\mathbf{v}}_i, \bar{\mathbf{a}}_i)\}_{i=1}^N$ are maintained as 'memory features' and used as targets for contrastive learning. The AVID task learns representations $(\mathbf{v}_i, \mathbf{a}_i)$ that are more similar to the memory features of the instance $(\bar{\mathbf{v}}_i, \bar{\mathbf{a}}_i)$ as opposed to memory features of other instances $(\bar{\mathbf{v}}_j, \bar{\mathbf{a}}_j)$, $j \neq i$. However, unlike previous approaches [206, 46] defined on a single modality (but similar to [190]), AVID uses multiple modalities, and thus can assume multiple forms as depicted in Figure 2.2.

1. **Self-AVID** requires instance discrimination within the same modality - $\mathbf{v}_i$ to $\bar{\mathbf{v}}_i$ and $\mathbf{a}_i$ to $\bar{\mathbf{a}}_i$.

This is equivalent to prior work [46, 206] independently applied to the two modalities.

2. **Cross-AVID** optimizes for cross-modal discrimination, *i.e.*, the visual representation $\mathbf{v}_i$ is required to discriminate the accompanying audio memory $\bar{\mathbf{a}}_i$ and vice-versa.

3. **Joint-AVID** combines the Self-AVID and Cross-AVID objectives.

It is not immediately obvious what the relative advantages, if any, of these variants are. In §2.3.3, we provide an in-depth empirical study of the impact of these choices on the quality of the learned representations. We now describe the training procedure in detail.

2.3.2 AVID training procedure.

AVID is trained using a contrastive learning framework [70, 69], where instance representations are contrasted to those of other (negative) samples.

While various loss functions have been defined for contrastive learning [147, 173], we focus on noise contrastive estimation (NCE) [69]. Let $\bar{\mathbf{x}}_i$ denote the (memory) target representation for a sample s_i. The probability that a feature $\mathbf{x}$ belongs to sample s_i is modeled by a generalized softmax function

$$P(s_i|\mathbf{x}) = \frac{1}{N\bar{Z}} \exp(\mathbf{x}^T \bar{\mathbf{x}}_i/\tau) \tag{2.1}$$

where $\bar{Z} = \frac{1}{N}\sum_{\bar{\mathbf{x}}}[\exp(\mathbf{x}^T\bar{\mathbf{x}}/\tau)]$ is the normalized partition function and τ is a temperature hyper-parameter that controls the softness of the distribution. In the case of AVID, $\mathbf{x}$ and $\bar{\mathbf{x}}$ may or may not be from the same modality.

The network f is trained to learn representations by solving multiple binary classification problems where it must choose its own target representation $\bar{\mathbf{x}}_i$ over representations $\bar{\mathbf{x}}_j$ in a negative set. The negative set consists of K 'other' instances drawn uniformly from $\mathcal{S}$, *i.e.*, $\mathcal{N}_i = \mathcal{U}(\mathcal{S})^K$. The probability of a feature $\mathbf{x}$ being from instance s_i as opposed to the instances

from the uniformly sampled negative set $\mathcal{N}_i$ is given as

$$P(D=1|\mathbf{x},\bar{\mathbf{x}}_i) = \frac{P(s_i|\mathbf{x})}{P(s_i|\mathbf{x}) + K/N}. \tag{2.2}$$

The NCE loss is defined as the negative log-likelihood

$$\mathcal{L}_{\text{NCE}}(\mathbf{x}_i;\bar{\mathbf{x}}_i,\mathcal{N}_i) = -\log P(D=1|\mathbf{x}_i,\bar{\mathbf{x}}_i) - \sum_{j\in\mathcal{N}_i} \log P(D=0|\mathbf{x}_i,\bar{\mathbf{x}}_j), \tag{2.3}$$

where $P(D=0|\cdot) = 1 - P(D=1|\cdot)$.

The three variants of AVID depicted in Figure 2.2 are trained to optimize variations of the NCE loss of Equation 2.3, by varying the target representations $\bar{\mathbf{x}}_i$.

$$\mathcal{L}_{\text{Self-AVID}} = \mathcal{L}_{\text{NCE}}(\mathbf{v}_i;\bar{\mathbf{v}}_i,\mathcal{N}_i) + \mathcal{L}_{\text{NCE}}(\mathbf{a}_i;\bar{\mathbf{a}}_i,\mathcal{N}_i) \tag{2.4}$$

$$\mathcal{L}_{\text{Cross-AVID}} = \mathcal{L}_{\text{NCE}}(\mathbf{v}_i;\bar{\mathbf{a}}_i,\mathcal{N}_i) + \mathcal{L}_{\text{NCE}}(\mathbf{a}_i;\bar{\mathbf{v}}_i,\mathcal{N}_i) \tag{2.5}$$

$$\mathcal{L}_{\text{Joint-AVID}} = \mathcal{L}_{\text{Self-AVID}}(\mathbf{v}_i,\mathbf{a}_i) + \mathcal{L}_{\text{Cross-AVID}}(\mathbf{v}_i,\mathbf{a}_i) \tag{2.6}$$

We analyze these variants next and show that the seemingly minor differences between them translate to significant differences in performance.

2.3.3 Analyzing AVID

We present experiments to analyze various properties of the AVID task and understand the key factors that enable the different variants of AVID to learn good representations.

Experimental Setup

We briefly describe the experimental setup for analysis and provide the full details in the supplemental.

Pre-training Dataset. All models are trained using the Audioset dataset [61] which contains 1.8M videos focusing on audio events. We randomly subsample 100K videos from this dataset to train our models. We use input video and audio clips of 1 and 2-second duration, respectively. The video model is trained on 16 frames of size 112×112 with standard data augmentation [187]. We preprocess the audio by randomly sampling the audio within 0.5 seconds of the video and compute a log spectrogram of size 100×129 (100 time steps with 129 frequency bands).

Video and audio models. The video model is a smaller version of the R(2+1)D models proposed in [191] with 9 layers. The audio network is a 9 layer 2D ConvNet with batch normalization. In both cases, output activations are max-pooled, projected into a 128-dimensional feature using a multi-layer perceptron (MLP), and normalized into the unit sphere. The MLP is composed of three fully connected layers with 512 hidden units.

Pre-training details. AVID variants are trained to optimize the loss in Equations 2.4-2.6 with 1024 random negatives. In early experiments, we increased the number of negatives up to 8192 without seeing noticeable differences in performance. Following [206], we set the temperature hyper-parameter τ to 0.07, the EMA update constant to 0.5, and the normalized partition function $\bar{Z}$ is approximated during the first iteration and kept constant thereafter ($\bar{Z} = 2.2045$). All models are trained with the Adam optimizer [101] for 400 epochs with a learning rate of 1e-4, weight decay of 1e-5, and batch size of 256.

Downstream tasks. We evaluate both the visual and audio features using transfer learning.

- Visual Features: We use the Kinetics dataset [196] for action recognition. We evaluate the pre-trained features by linear probing [68, 216] where we keep the pre-trained network fixed and train linear classifiers. We report top-1 accuracy on held-out data by averaging predictions over 25 clips per video.
- Audio Features: We evaluate the audio features on the ESC-50 [160] dataset by training linear classifiers on fixed features from the pre-trained audio network. Similar to the video

Table 2.1: Accuracy of linear probing on Kinetics.

Method	**block1**	**block2**	**block3**	**block4**	**Best**
Cross-AVID	**19.80**	**26.98**	**34.81**	**39.95**	**39.95**
Self-AVID	17.10	22.28	27.23	32.08	32.08
Joint-AVID	18.65	23.60	29.47	33.04	33.04

Table 2.2: Accuracy of linear probing on ESC.

	block1	**block2**	**block3**	**block4**	**Best**
Cross-AVID	**67.25**	**73.15**	**74.80**	**75.05**	**75.05**
Self-AVID	66.92	72.64	71.45	71.61	72.64
Joint-AVID	65.45	68.65	71.77	68.41	71.77

case, we report top-1 accuracy by averaging predictions over 25 clips per video.

Cross *vs*. within-modal instance discrimination

We study the three variants of AVID depicted in Figure 2.2 to understand the differences between cross-modal and within-modal instance discrimination and its impact on the learned representations. We evaluate the video and audio feature representations from these variants and report results in Table 2.1 and Table 2.2. We observe that Self-AVID is consistently outperformed by the Cross-AVID variant on both visual and audio tasks.

We believe the reason is that Self-AVID uses within-modality instance discrimination, which is an easier pretext task and can be partially solved by matching low-level statistics of the data [44, 8]. This hypothesis is supported by the fact that Joint-AVID, which combines the objectives of both Cross-AVID and Self-AVID, also gives worse performance than Cross-AVID. These results highlight that one *cannot* naively use within-modality instance discrimination when learning audio-visual representations. In contrast, Cross-AVID uses a "harder" cross-modal instance discrimination task where the video features are required to match the corresponding audio and vice-versa. As a result, it generalizes better to downstream tasks.

2.4 Beyond Instance Discrimination

We will show in §2.5 that Cross-AVID achieves state-of-the-art performance on action recognition downstream tasks. However, we identify three important limitations in the instance discrimination framework of Equation 2.3 and the cross-modal loss of Equation 2.5.

1. **Limited to instances:** Instance discrimination does not account for interactions *between* instances. Thus, two semantically related instances are never grouped together and considered 'positives'.

2. **False negative sampling:** The negative set $\mathcal{N}_i$, which consists of *all* other instances s_j, may include instances semantically related to s_i. To make matters worse, contrastive learning requires a large number K of negatives, increasing the likelihood that semantically related samples are used as negatives. This contradicts the goal of representation learning, which is to generate similar embeddings of semantically related inputs.

3. **No within-modality calibration:** The Cross-AVID loss of Equation 2.5 does not directly optimize for visual similarity $\mathbf{v}_i^T \mathbf{v}_j$. In fact, as shown experimentally in §2.3.3, doing so can significantly hurt performance. Nevertheless, the lack of within-modality calibration is problematic, as good visual representations should reflect visual feature similarities.

2.4.1 Relating instances through agreements

We extend AVID with Cross-Modal Agreement (CMA) to address these shortcomings. CMA builds upon insights from prior work [169] in multi-view learning. We hypothesize that, if two samples are similar in *both* visual and audio feature space, then they are more likely to be semantically related than samples that agree in only one feature space (or do not agree at all). We thus consider instances that agree in both feature spaces to be 'positive' samples for learning representations. Similarly, examples with a poor agreement in either (or both) spaces are used as

negatives. When compared to instance discrimination methods [206, 190, 46], CMA uses a larger positive set of semantically related instances and a more reliable negative set.

2.4.2 CMA Learning Objective

We define an agreement score for two instances s_i and s_j as

$$\rho_{ij} = \min(\mathbf{v}_i^T \mathbf{v}_j, \mathbf{a}_i^T \mathbf{a}_j). \tag{2.7}$$

This is large only when *both* the audio and video similarities are large. A set of positives and negatives is then defined per instance s_i. The positive set $\mathcal{P}_i$ contains the samples that are most similar to s_i in both spaces, while the negative set $\mathcal{N}_i$ is the complement of $\mathcal{P}_i$.

$$\mathcal{P}_i = \underset{j=1,\ldots,N}{\text{TopK}}\ (\rho_{ij}) \qquad \mathcal{N}_i = \{j | s_j \in (\mathcal{S} \setminus \mathcal{P}_i)\} \tag{2.8}$$

Furthermore, CMA enables self-supervision beyond single instances. This is achieved with a generalization of the AVID task, which accounts for the correspondences of Equation 2.8. At training time, K_n negative instances are drawn per sample s_i from the associated negative set $\mathcal{N}_i$ to form set $\mathcal{N}_i' = \mathcal{U}(\mathcal{N}_i)^{K_n}$. The networks f_v, f_a are learned to optimize a combination of *cross-modal instance discrimination* and *within-modal positive discrimination* (wMPD). The former is encouraged through the Cross-AVID loss of Equation 2.5. The latter exploits the fact that CMA defines *multiple* positive instances $\mathcal{P}_i$, thus enabling the optimization of within-modality positive discrimination

$$\mathcal{L}_{\text{wMPD}} = \frac{1}{K_p} \sum_{p \in \mathcal{P}_i} \mathcal{L}_{\text{NCE}}(\mathbf{v}_i; \bar{\mathbf{v}}_p, \mathcal{N}_i') + \mathcal{L}_{\text{NCE}}(\mathbf{a}_i; \bar{\mathbf{a}}_p, \mathcal{N}_i'). \tag{2.9}$$

Note that, unlike the Self-AVID objective of Equation 2.4, this term calibrates within-modal similarities between positive samples. This avoids within-modal comparisons to the instance

itself, which was experimentally shown to produce weak representations in §2.3.3. We then minimize the weighted sum of the two losses

$$\mathcal{L}_{\text{CMA}} = \mathcal{L}_{\text{Cross-AVID}}(\mathbf{v}_i, \mathbf{a}_i) + \lambda \mathcal{L}_{\text{wMPD}}(\mathbf{v}_i, \mathbf{a}_i), \tag{2.10}$$

where $\lambda > 0$ is an hyper-parameter that controls the weight of the two losses.

Implementation

After Cross-AVID pre-training, cross-modal disagreements are corrected by finetuning the audio and video networks to minimize the loss in Equation 2.10. Models are initialized with the Cross-AVID model at epoch 200, and trained for 200 additional epochs. We compare these models to a Cross-AVID model trained for 400 epochs, thus controlling for the total number of parameter updates. For each sample, we find 32 positive instances using the CMA criterion of Equation 2.8 applied to video and audio memory bank representations. For efficiency purposes, the positive set is updated every 50 epochs. In each iteration, 1024 negative memories (not overlapping with positives) were sampled. These positive and negative memories were then used to minimize the CMA loss of Equations 2.9-2.10. For evaluation purposes, we use the same protocol as in §2.3.3.

2.4.3 Analyzing CMA

The CMA objective consists of two terms that optimize cross-modal (Equation 2.5) and within-modal (Equation 2.9) similarity. We observed in §2.3.3 that within-modal comparisons for instance discrimination result in poor visual representations due to the relatively easy task of self-discrimination. Intuitively, since CMA identifies groups of instances ($\mathcal{P}_i$) that are likely related, calibrating within-modal similarity within these groups (instead of within the instance itself) should result in a better visual representation. To study this, we use CMA to obtain a

positive set $\mathcal{P}_i$ and analyse the CMA objective of Equation 2.10 by evaluating with different values of the hyper-parameter λ. The results shown in Figure 2.3 validates the advantages of CMA over Cross-AVID.

CMA calibration

To understand the effect of the CMA procedure on within-modal similarities, we analyzed the embedding space defined by memory bank representations obtained with AVID and CMA trained on the Kinetics dataset. Since representations are restricted to the unit sphere (due to normalization), the average inner-product between two randomly chosen samples should be 0 (assuming a uniform distribution of samples over the sphere). However, when training with Cross-AVID, the average inner-product is 0.23. This means that Cross-AVID learns collapsed representations (*i.e*. features are on average closer to other random features than the space permits). This is likely due to the lack of within-modal negatives when training for cross-modal discrimination. By seeking within modal-discrimination of positive samples, CMA effectively addresses the feature collapsing problem observed for Cross-AVID, and yields an average dot-product between random memories of 0 as expected.

CMA *vs*. within-modal expansion

CMA expands the positive set $\mathcal{P}_i$ to include instances that *agree in both* video and audio spaces. We inspected whether modeling this agreement is necessary for relating instances by exploring alternatives that do not model agreements in both spaces (see Figure 2.4). We consider alternatives that expand the set $\mathcal{P}_i$ by looking at instances that are similar in 1) only the audio space; 2) only the video space; or 3) either video or audio space. Each method in Figure 2.4 is trained to optimize the objective of Equation 2.10 with the corresponding $\mathcal{P}_i$. We also compare against the Cross-AVID baseline that uses only the instance itself as the positive set. Transfer performance is reported in Table 2.3.

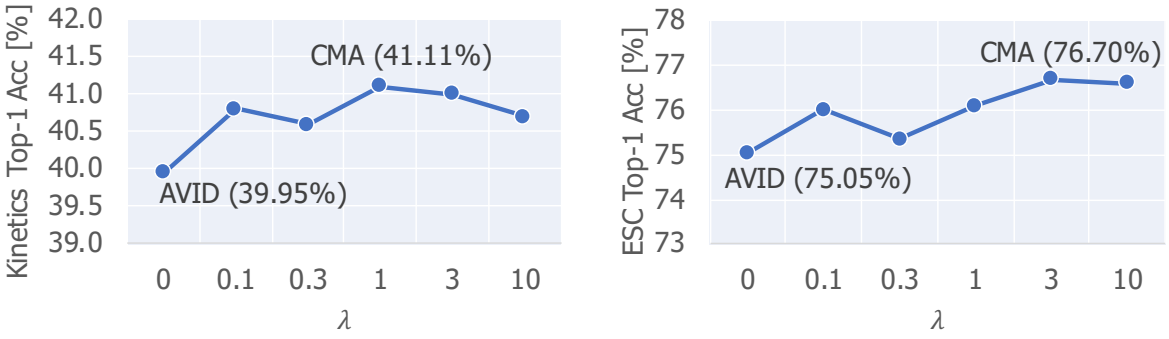


Figure 2.3: Impact of within-modal positive sample discrimination. Positive sample discrimination can improve the performance of Cross-AVID.

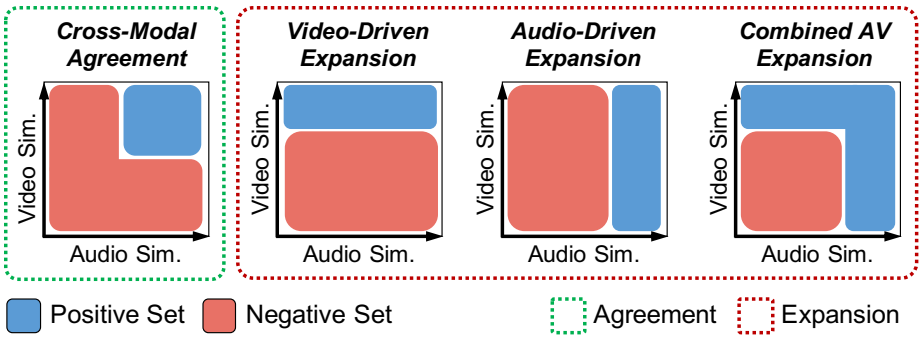


Figure 2.4: Comparison of CMA to expansion methods that relate instances without modeling agreement.

Compared to Cross-AVID, expanding the set of positives using only audio similarity (third row) hurts performance on Kinetics, and relying on video similarities alone (second row) only provides marginal improvements. We believe that expanding the set of positives only based on visual similarity does not improve the performance of visual features since the positives are *already close* in the feature space, and do not add extra information. CMA provides consistent gains over all methods on Kinetics, suggesting that modeling *agreement* can provide better positive sets for representation learning of visual features.

Table 2.3: Top-1 accuracy of linear probing on Kinetics. CMA enables better transfer for action recognition.

Method	block1	block2	block3	block4	Best
Cross-AVID (Base)	19.80	26.98	34.81	39.95	39.95
Base + Video-Exp.	19.93	27.39	35.64	40.17	40.17
Base + Audio-Exp.	20.14	27.28	35.68	39.62	39.62
Base + AV Exp	20.04	27.61	36.14	40.58	40.58
Base + CMA	**20.16**	**27.98**	**36.98**	**41.11**	**41.11**

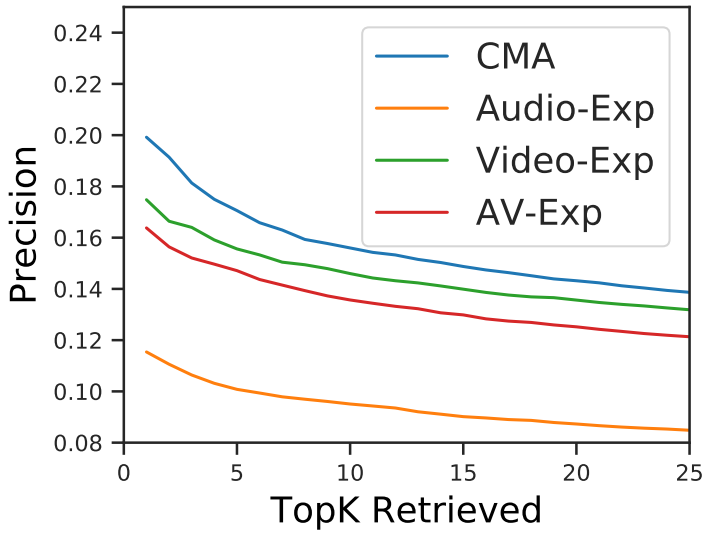


Figure 2.5: Precision@K. Expansion methods generate agreements of worse precision.

Qualitative Understanding

We show examples of positive and negative samples found by CMA in Figure 2.6 and observe that CMA can group together semantically related concepts. As it uses agreement between both spaces, visually similar concepts, like 'ambulance' and 'bus' (second row), can be distinguished based on audio similarity. This leads to more precise positive sets $\mathcal{P}_i$, as can be verified by inspecting the precision@K of $\mathcal{P}_i$ measured against ground truth labels (Figure 2.5). CMA consistently finds more precise positives compared to within-modal expansion methods showing the advantages of modeling agreement.

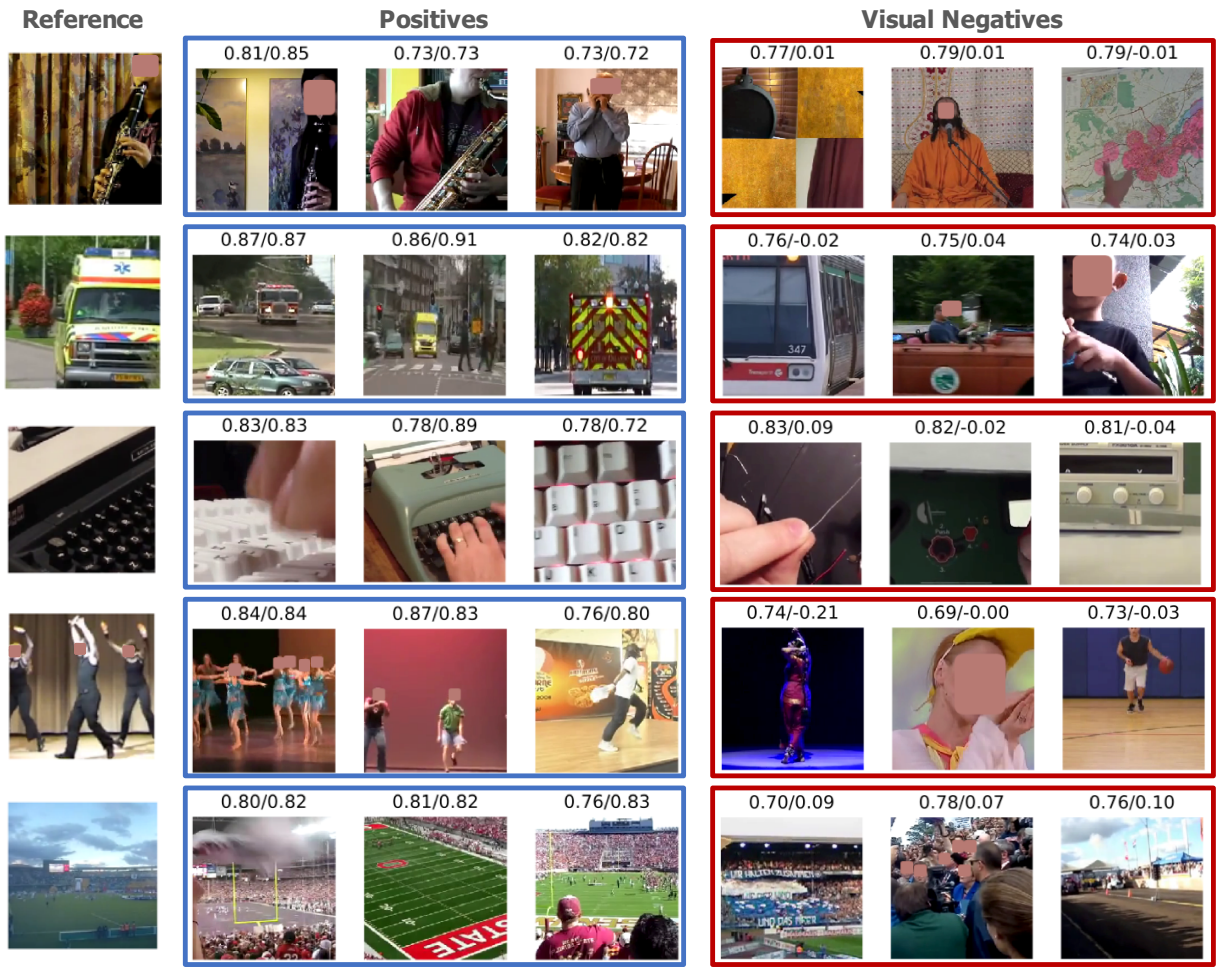


Figure 2.6: Examples extracted by the CMA procedure showing three images in their positive sets (Equation 2.8), and three negatives that were rejected from the positive set due to low audio similarity.

2.5 Cross-AVID and CMA at scale

Previous sections provide experimental validation for the proposed Cross-AVID and CMA procedures when training on a medium-sized dataset (100K videos from Audioset). We now study the proposed methods on large-scale datasets. We also compare Cross-AVID and CMA to prior work, including video-based self-supervised learning methods [134, 118, 210, 72], and methods that leverage the natural correspondence between audio and video [8, 148, 105].

Experimental setup.

We briefly describe the experimental setup, and refer the reader to supplementary material for full details. We use the 18-layer R(2+1)D network of [191] as the video encoder and a 9-layer (2D) CNN with batch normalization as the audio encoder. Models are trained on Kinetics-400 [196] and the full Audioset [61] datasets, containing 240K and 1.8M video instances, respectively. Video clips composed of 8 frames of size 224×224 are extracted at a frame rate of 16fps with standard data augmentation procedures [187]. Two seconds of audio is randomly sampled within 0.5 seconds of the video at a 24kHz sampling rate, and spectrograms of size 200×257 (200 time steps with 257 frequency bands) are used as the input to the audio network. For Cross-AVID, the cross-modal discrimination loss of Equation 2.5 is optimized with $K = 1024$ negative instances. We then find 128 positive instances for each sample using cross-modal agreements (Equation 2.8), and optimize the CMA criterion of Equation 2.10 with $K_p = 32$ positives, $K_n = 1024$ negatives and $\lambda = 1.0$. Video representations are evaluated on action recognition (§2.5.1), and audio representations on sound classification (§2.5.2).

2.5.1 Action recognition

We first evaluate the visual representations learned by Cross-AVID and AVID+CMA by training a linear classifier for the task of action recognition on the Kinetics dataset. The top-1 accuracy is reported for clip and video-level predictions. Clip-level predictions are obtained from a single 8-frame clip, while video-level predictions are computed by averaging clip-level predictions from 10 clips uniformly sampled from the whole video. The results shown in Table 2.4 clearly demonstrate the advantage of calibrating AVID representations using the CMA procedure, yielding significant gains across both metrics and pretraining datasets. These results demonstrate the value of the CMA procedure in large-scale datasets, thus showing that its effect goes beyond a simple regularization procedure to prevent overfitting.

Table 2.4: Top-1 accuracy of linear probing on Kinetics.

Pretraining DB	**Kinetics**		**Audioset**	
Method \ Metric	Clip@1	Video@1	Clip@1	Video@1
Cross-AVID	33.3	43.1	35.2	46.6
AVID+CMA	**35.1**	**44.5**	**37.4**	**48.9**

To compare to prior work, we follow [72, 105, 190] and evaluate visual representations on the UCF-101 [182] and HMDB-51 [109] datasets, by full network fine-tuning. Due to the large variability of experimental setups used in the literature, it is unrealistic to provide a direct comparison to all methods, as these often use different network encoders trained on different datasets with input clips of different lengths. To increase the range of meaningful comparisons, we fine-tuned our models using clips with both 8 and 32 frames. At inference time, video-level predictions are provided by averaging clip-level predictions for 10 uniformly sampled clips [105]. We report top-1 accuracy averaged over the three train/test splits provided with the original datasets.

Table 2.5 compares the transfer performance of Cross-AVID and CMA with previous self-supervised approaches. To enable well-grounded comparisons, we also list for each method the pre-training dataset and clip dimensions used while finetuning on UCF and HMDB. Despite its simplicity, Cross-AVID achieves state-of-the-art performance for equivalent data settings in most cases. In particular, when pre-trained on Audioset, Cross-AVID outperformed other audio-visual SSL methods such as L3 and AVTS by at least 1.0% on UCF and 2.5% on HMDB. Similar to Cross-AVID, L3 and AVTS propose to learn audio-visual representations by predicting whether audio/video pairs are in-sync. However, these methods optimize for the audiovisual correspondence task, which fails to reason about the data distribution at large. Cross-AVID also outperformed the concurrently proposed XDC [4] under equivalent data settings. When pretrained on Audioset and finetuned on UCF with 32 frames, XDC [4] does report higher accuracy, but the model was pretrained and finetuned using 32 frames, while we pretrain using only 8 frames.

Table 2.5: Top-1 accuracy on UCF and HMDB by full network finetuning with various pretraining datasets and clips of different sizes.

Method	Pretraining DB	Finetune Input Size	UCF	HMDB
Shuffle&Learn [134]	UCF	1×227^2	50.2	18.1
OPN [118]	UCF	1×227^2	56.3	23.8
ST Order [23]	UCF	1×227^2	58.6	25.0
CMC [190]	UCF	1×227^2	59.1	26.7
3D-RotNet [93]	Kinetics400	16×112^2	62.9	33.7
ClipOrder [210]	Kinetics400	16×112^2	72.4	30.9
DPC [72]	Kinetics400	25×128^2	75.7	35.7
CBT [186]	Kinetics400	16×112^2	79.5	44.6
L3* [8]	Kinetics400	16×224^2	74.4	47.8
AVTS [105]	Kinetics400	25×224^2	85.8	56.9
XDC [4]	Kinetics400	8×224^2	74.2	39.0
	Kinetics400	32×224^2	86.8†	52.6†
Cross-AVID (ours)	Kinetics400	8×224^2	82.3	49.1
	Kinetics400	32×224^2	<u>86.9</u>	<u>59.9</u>
AVID+CMA (ours)	Kinetics400	8×224^2	83.7	49.5
	Kinetics400	32×224^2	**87.5**	**60.8**
L3* [8]	Audioset	16×224^2	82.3	51.6
Multisensory [148]	Audioset	64×224^2	82.1	–
AVTS [105]	Audioset	25×224^2	89.0	61.6
XDC [4]	Audioset	8×224^2	84.9	48.8
	Audioset	32×224^2	93.0†	63.7†
Cross-AVID (ours)	Audioset	8×224^2	88.3	57.5
	Audioset	32×224^2	<u>91.0</u>	<u>64.1</u>
AVID+CMA (ours)	Audioset	8×224^2	88.6	57.6
	Audioset	32×224^2	**91.5**	**64.7**

It should be noted that, when pretraining and finetuning with clips of 8 frames, Cross-AVID outperforms XDC by 3.4% (84.9% vs 88.3%). CMA further improves the performance of Cross-AVID on all settings considered (*i.e.*, using both Kinetics and Audioset pretraining datasets, and evaluating on UCF and HMDB). We observed, however, that the improvements of CMA over Cross-AVID are smaller under the fine-tuning protocol than the linear evaluation of Table 2.4. Prior work [68, 216] observes that full fine-tuning significantly modifies the visual features and

tests the network initialization aspect of pre-training rather than the semantic quality of the representation. Thus, we believe that the feature calibration benefits of CMA are diminished under the full finetuning protocol.

2.5.2 Sound recognition

Audio representations are evaluated on the ESC-50 [160] and DCASE [183] datasets by linear probing [68] for the task of sound recognition. Following [105], both ESC and DCASE results are obtained by training a linear one-vs-all SVM classifier on the audio representations generated by the pre-trained models at the final layer before pooling. For training, we extract 10 clips per sample on the ESC dataset and 60 clips per sample on DCASE [105]. At test time, sample level predictions are obtained by averaging 10 clip level predictions, and the top-1 accuracy is reported in Table 2.6. For the ESC dataset, performance is the average over the 5 original train/test splits. Similarly to video, audio representations learned by Cross-AVID and CMA outperform prior work, outperforming ConvRBM on the ESC dataset by 2.7% and AVTS on DCASE by 3%.

2.6 Discussion

We proposed a self-supervised method to learn visual and audio representations by contrasting visual representations against multiple audios, and vice versa. Our method, Audio-Visual Instance Discrimination (AVID) builds upon recent advances in contrastive learning [206, 190] to learn state-of-the-art representations that outperform prior work on action recognition and sound classification. We propose and analyze multiple variants of the AVID task to show that optimizing for cross-modal similarity and not within-modal similarity matters for learning from video and audio.

We also identified key limitations of the instance discrimination framework and proposed

Table 2.6: Top-1 accuracy of linear classification on ESC-50 and DCASE datasets.

Method	Pretraining DB	ESC	DCASE
RandomForest [160]	None	44.3	–
ConvNet [159]	None	64.5	–
ConvRBM [170]	None	86.5	–
SoundNet [13]	Flickr-SoundNet	74.2	88
L3 [8]	Flickr-SoundNet	79.3	93
AVTS [105]	Kinetics	76.7	91
XDC [4]	Kinetics	78.5	–
Cross-AVID (Ours)	Kinetics	77.6	**93**
AVID+CMA (Ours)	Kinetics	**79.1**	**93**
AVTS [105]	Audioset	80.6	93
XDC [4]	Audioset	85.8	–
Cross-AVID (Ours)	Audioset	**89.2**	**96**
AVID+CMA (Ours)	Audioset	89.1	**96**

CMA to use agreement in the video and audio feature spaces to group together related videos. CMA helps us relate *multiple* instances by identifying more related videos. CMA also helps us reject 'false positives', *i.e.*, videos that are similar visually but differ in the audio space. We show that using these groups of related videos allows us to optimize for within-modal similarity, in addition to cross-modal similarity, and improve visual and audio representations. The generalization of CMA suggests that cross-modal agreements provide non-trivial correspondences between samples and are a useful way to learn improved representations in a multi-modal setting.

2.7 Appendix

2.7.1 Experimental setup

Architecture details

The architecture details of the video and audio networks used in the analysis experiments are shown in Table 2.12 and Table 2.13, and those used for comparison to prior work is shown

in Table 2.14 and Table 2.15.

Pre-training hyper-parameters

Optimization and data augmentation hyper-parameters for AVID and CMA pre-training are provided in Table 2.9.

Action recognition hyper-parameters

Optimization and data augmentation hyper-parameters for action recognition tasks are provided in Table 2.10.

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 2.9 for pre-training and Table 2.10 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.

2.7.2 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 2.7. Cross-AVID transfer performance seem to have already saturated after 200 epochs of pre-training.

2.7.3 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 2.8.

2.8 Acknowledgements

Chapter 2 is, in full, based on the material as it appears in the publication of "Audio-Visual Instance Discrimination with Cross-Modal Agreement", Pedro Morgado, Ishan Misra, Nuno Vasconcelos, to appear in the Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. The dissertation author was the primary investigator and author of this material.

Table 2.7: 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 2.8: 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**

Table 2.9: Pre-training optimization hyper-parameters. CMA models are initialized by the AVID model obtained at epoch 200.

Method	DB	bs	lr	wd	ep	es	msc	hf	bj	sj	cj	hj
AVID (§2.3)	Audioset	32	5e-4	1e-5	400	1e5	✓	0.5	0.4	0.4	0.4	0.2
AVID (§2.5)	Audioset	32	5e-4	1e-5	200	1.8e6	✓	0.5	0.4	0.4	0.4	0.2
AVID (§2.5)	Kinetics	32	2e-4	1e-5	300	2.4e5	✓	0.5	0.4	0.4	0.4	0.2
CMA (§2.4)	Audioset	32	5e-4	1e-5	200	1e5	✓	0.5	0.4	0.4	0.4	0.2
CMA (§2.5)	Audioset	32	5e-4	1e-5	200	1.8e6	✓	0.5	0.4	0.4	0.4	0.2
CMA (§2.5)	Kinetics	32	2e-4	1e-5	300	2.4e5	✓	0.5	0.4	0.4	0.4	0.2

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.

Table 2.10: Transfer learning optimization hyper-parameters.

DB	input size	bs	lr	wd	ep	es	gm	mls
Kinetics (§2.3, §2.4)	16×112^2	32	1e-4	0.	20	1e4	0.3	8,12,15,18
UCF (§2.5)	8×224^2	32	1e-4	0.	160	1e4	0.3	60,100,140
UCF (§2.5)	32×224^2	16	1e-4	0.	80	1e4	0.3	30,50,70
HMDB (§2.5)	8×224^2	32	1e-4	0.	250	3.4e3	0.3	75,150,200
HMDB (§2.5)	32×224^2	16	1e-4	0.	100	3.4e3	0.3	30,60,80

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;

Table 2.11: Data augmentation hyper-parameters.

DB	msc	hf	bj	sj	cj	hj
Kinetics (§4, §5)	✓	0.5	0.	0.	0.	0.
UCF (§2.5)	✓	0.5	0.4	0.4	0.4	0.2
HMDB (§2.5)	✓	0.5	1.	1.	1.	0.2

msc - multi-scale cropping; hf - horizontal flip probability;
bj/sj/cj/hj - brightness/saturation/contrast/hue jittering intensity.

Table 2.12: Architecture details of R(2+1)D video network for analysis experiments. The video network is based of R(2+1)D convolutions with ReLU activations and batch normalization at each layer.

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	-	-	-	-

X_s spatial activation size; X_t temporal activation size; C number of channels
K_s spatial kernel size; K_t temporal kernel size; S_s spatial stride; S_t temporal stride;

Table 2.13: Architecture details of Conv2D audio network for analysis experiments. The audio network is based on 2D convolutions with ReLU activations and batch normalization at each layer.

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	-	-	-	-

X_t temporal activation size; X_f frequency activation size; C number of channels
K_t temporal kernel size; K_f frequency kernel size; S_t temporal stride; S_f frequency stride.

Table 2.14: Architecture details of R(2+1)D video network for comparison to prior work. The video network is based of R(2+1)D convolutions with ReLU activations and batch normalization at each layer.

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	-	-	-	-

X_s spatial activation size, X_t temporal activation size, C number of channels
K_s spatial kernel size, K_t temporal kernel size, S_s spatial stride, S_t temporal stride.

Table 2.15: Architecture details of Conv2D audio network for comparison to prior work. The audio network is based on 2D convolutions with ReLU activations and batch normalization at each layer.

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	-	-	-	-

X_t temporal activation size, X_f frequency activation size, C number of channels
K_t temporal kernel size, K_f frequency kernel size, S_t temporal stride, S_f frequency stride

Chapter 3

Robust Audio-Visual Instance Discrimination

3.1 Introduction

Self-supervised representation learning aims to learn feature representations that can transfer to downstream tasks without costly human annotations. Many recent self-supervised methods [29, 76, 133, 32, 202, 190] use a variant of the instance discrimination framework [206, 47], which matches features from multiple views/augmentations of the *same* instance, while distinguishing these features from those of other instances. This often relies on a contrastive loss [70], where different augmentations are considered 'positives' and other samples 'negatives.'

Cross-modal instance discrimination (xID) extends instance discrimination to the realm of multiple modalities, where data modalities, such as video, audio, or text, act as the different 'views' of an instance. Since there is a strong correlation between audio and visual events (*e.g.*, the sound of an instrument or a baseball match), audio-visual instance discrimination has gained popularity [8, 148, 105, 138, 161, 4, 155, 3]. Representations learned by these methods show promising performance on tasks like action recognition and environmental sound classification. xID methods rely on two key assumptions - (1) the audio and video of a sample are informative of each other, *i.e.*, positives; (2) the audio and video of all other samples are not related, *i.e.*, negatives. In practice, both these assumptions are too strong and do not hold for a significant amount of real-world data. This results in *faulty positive* samples that are not related to each other and *faulty negative* samples that are semantically related.

Figure 3.1 shows examples of these faulty correspondences. Videos where the audio is uninformative of the visual content can lead to faulty positives, *e.g.*, videos containing audio from sources outside of the camera field-of-view or containing post-edited sounds like a soundtrack. Similarly, random negative sampling can produce faulty negatives, *i.e.*, negative samples that are semantically related to the positive. These faulty correspondences undermine the primary goal of representation learning, *i.e.*, to ensure that similar instances have similar feature representations. As we show empirically in Figure 3.7 and Table 3.1, they can hurt representation learning and

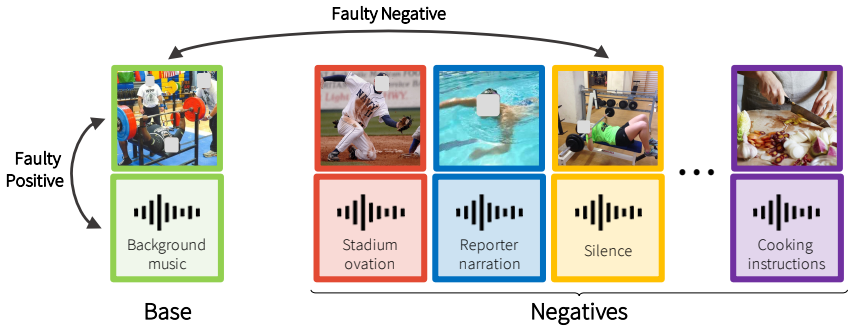


Figure 3.1: Example of a positive audio/video pair and negative instances used for contrastive learning. Faulty positive and negative samples are a common occurrence in audio-visual contrastive learning and hurt representation learning.

degrade downstream performance. Thus, we believe cross-modal learning should be seen as a problem of learning with *noisy targets*. This raises the question of how to identify faulty positive and negative samples in the absence of human annotations.

We propose to use cross-modal information during self-supervised training to detect both faulty positive and negative instances. This is done by estimating the quality of the audio-visual correspondence of each instance and optimizing a weighted contrastive learning loss that down-weighs the contribution of faulty positive examples. To address faulty negatives, we estimate the similarity *across* instances to compute a soft target distribution over instances. The model is then tasked to match this distribution. As a result, instances with enough evidence of similarity are no longer used as negatives and may even be used as positives.

The contributions of this work are as follows (Figure 3.2). We identify two sources of training noise in cross-modal learning: instances with weak cross-modal correspondence, which create *faulty positives*, and the sampling of semantically similar instances as negatives, which create *faulty negatives*. We show that removing faulty positives and negatives using an oracle can significantly improve the performance of a state-of-the-art xID method [138]. We then propose a mechanism to replace the oracle and a robust cross-modal instance discrimination loss that limits the impact of faulty correspondences. The effectiveness of the proposed method is demonstrated on several downstream tasks.

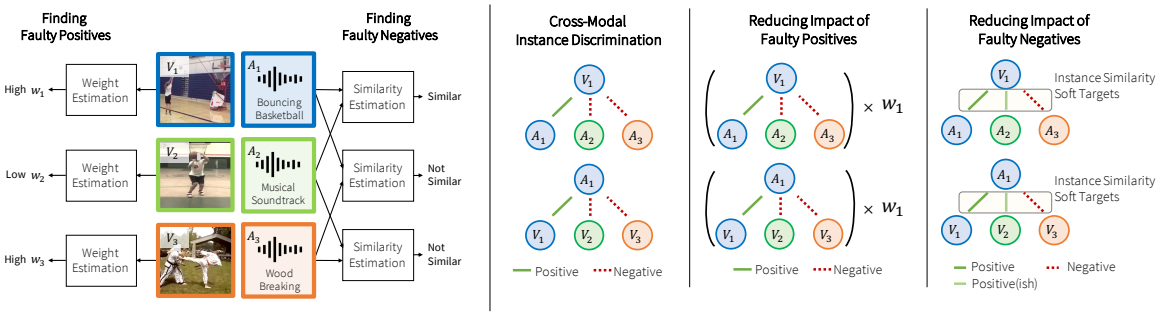


Figure 3.2: Comparison between standard cross-modal instance discrimination (xID) and the proposed procedure. The proposed method addresses the two main sources of noisy training signals: faulty positives and faulty negatives.

3.2 Related work

Self-supervised representation learning

Self-supervised representation learning aims to learn representations by solving pretext tasks defined from the data alone, *i.e.* without human annotations. In computer vision, pretext tasks involve reasoning about spatial context [142, 44, 99, 154, 64, 73, 163], temporal context [134, 118, 204, 99, 135, 72, 54, 199, 18, 73, 74, 163], other visual properties such as hue, brightness and flow [41, 113, 215, 114, 216, 190, 161], or clusters of features [27, 12, 29, 202]. One promising technique is the instance discrimination task proposed in [206, 47] and further explored in [76, 133, 32, 202, 208]. However, contrastive learning from a single modality requires heavy data augmentations to generate distinct views. Instead, we focus on cross-modal instance discrimination, which avoids this issue by generating views from different modalities.

Representation learning from audio-visual correspondences

Since, in video, the audio is naturally paired and synced with the visual component, audio-visual correspondences have been used to draw direct supervision for several tasks, such as visually guided-source separation and localization [58, 60, 221, 220, 57, 175], visually guided audio spatialization [137, 59], audio-visual embodied navigation [31], lip-speech synchronization [37]

and audio-visual speech recognition [2, 36].

In the context of contrastive learning, audio-visual correspondences are used to generate alternative views of an instance. While this has been known for a long time [39], self-supervised audio-visual representation learning gained popularity in recent years. For example, [8, 7] propose to learn representations by solving a *binary* classification problem that identifies audio and video clips belonging to the same instance. [105, 148] predict if audio and video clips are temporally synchronized, and [136] predicts if audio and video clips extracted from a 360 video are spatially aligned. [138, 155] improve upon the audio-visual correspondence problem [8] by posing it as a cross-modal instance discrimination task, where instances are contrasted to a large number of negatives. As a result, [138, 155] achieve impressive performance on downstream tasks such as action recognition.

In this work, we address two issues inherent to cross-modal instance discrimination, namely the detrimental impact of faulty positives and negatives. Recently, [4, 11] proposed to learn representations by iteratively clustering the audio and visual representations and seeking to predict cluster assignments from the opposite modality. While clustering can also discourage faulty negatives from acting as repelling forces, our method accomplishes this by optimizing a simple instance discrimination loss with soft targets, thus avoiding the significant computational overhead of clustering.

Supervised learning from noisy labels

Our work is closely related to supervised learning from noisy labels [167, 217, 156, 71, 122]. Since label collection is expensive and time-consuming, scaling human annotation to large datasets often requires the use of non-experts or non-curated labels such as user tags, which are prone to noise. Since deep neural networks can easily overfit to noisy labels [212], this results in poor generalization. Several techniques have been developed to increase the robustness of learning algorithms to label noise, including losses that reduce the impact of outliers [63, 217, 203], loss

correction approaches that model the sources of label noise [156, 81, 30, 167, 9, 128, 180], meta-learning procedures that learn how to correct the sources of label noise [122, 168, 121, 178, 219] and regularization procedures tailored to lower the impact of noise [213, 158]. We refer the reader to [181, 55] for a detailed survey of prior work on learning with label noise. In this work, we show that cross-modal instance discrimination should be seen as a problem of learning with noisy targets. However, instead of the class mislabeling, we identify two main sources of noise for cross-modal instance discrimination (faulty positives and faulty negatives) and propose an algorithm to mitigate them.

3.3 Analysis: Instance Discrimination

We analyze the cross-modal instance discrimination method [138, 190, 155] and show that faulty positives and negatives have a disproportionately large contribution to the training updates. Additionally, in Table 3.1, we document the detrimental empirical effects of faulty samples.

Cross-Modal Instance Discrimination

Consider a dataset $\mathcal{D} = \{(v_i, a_i)_{i=1}^N\}$ containing N samples (or instances) of video v_i and audio a_i. Cross-modal instance discrimination uses a contrastive loss [70] to learn video and audio encoders, $f_v(\cdot)$ and $f_a(\cdot)$, so as to align the two modalities belonging to the same instance [190, 138, 155] by minimizing

$$L_{\text{xID}}(\mathbf{v}_i, \mathbf{a}_i) = -\log P(\bar{\mathbf{a}}_i|\mathbf{v}_i;\tau) - \log P(\bar{\mathbf{v}}_i|\mathbf{a}_i;\tau) \tag{3.1}$$

$$\text{where} \quad P(\bar{\mathbf{t}}_i|\mathbf{s}_i;\tau) = \frac{\exp(\mathbf{s}_i^T \bar{\mathbf{t}}_i/\tau)}{\sum_k \exp(\mathbf{s}_i^T \bar{\mathbf{t}}_k/\tau)}, \tag{3.2}$$

where $\mathbf{v}_i = f_v(v_i)$ and $\mathbf{a}_i = f_a(a_i)$ are visual and audio features normalized to the unit sphere, $\bar{\mathbf{v}}_i$ and $\bar{\mathbf{a}}_i$ are target representations, and τ is a temperature hyper-parameter. Prior works differ by the type of target representations employed. For example, $\bar{\mathbf{v}}_i$ and $\bar{\mathbf{a}}_i$ can be entries of a memory bank as in [138, 206], the network representations themselves $\bar{\mathbf{v}}_i = f_v(v_i)$ and $\bar{\mathbf{a}}_i = f_a(a_i)$ as in SimCLR [32], the outputs of momentum encoders as in MoCo [76], or the centroids of an online clustering procedure as in SwAV or CLD [29, 202]. In this work, we build on the Audio-Visual Instance Discrimination (AVID) method of [138], focusing on target representations sampled from a memory bank. However, the principles introduced below can also be applied to SimCLR, MoCo or SwAV style targets.

Faulty positives and negatives in practice

The contrastive loss of Equation 3.1 is minimized when audio and visual representations from the same instance are aligned (dot-product similarities $\mathbf{v}_i^T \bar{\mathbf{a}}_i$ and $\mathbf{a}_i^T \bar{\mathbf{v}}_i$ as close to 1 as possible), and representations from different instances are far apart. In practice, however, the two modalities are not informative of each other for a significant number of instances (see Figure 3.1). We refer to these unclear correspondences as *faulty positives*.[1] On the other hand, a significant number of contrastive learning negatives are semantically similar to the base instance. We term these semantically similar negatives as *faulty negatives* since they should ideally be used as positives.

Figure 3.3 shows the histogram of similarities $\bar{\mathbf{v}}_i^T \bar{\mathbf{a}}_i$ after training an audio-visual model with the loss of Equation 3.1. As can be seen, instances with higher scores tend to have stronger correspondences (*i.e.* the audio and video signals are informative of each other). Instances where the two modalities are uninformative of each other tend to have lower scores and are generally faulty positives. On the other hand, Figure 3.4 shows the histograms of similarities between a video i and negatives j. As can be seen, faulty negatives tend to occur for negatives j with high

[1] We prefer 'faulty positives' over 'false positives' to distinguish from supervised learning where one has access to labels.

similarity $\bar{\mathbf{v}}_i^T \bar{\mathbf{a}}_j$.

How do faulty positives and negatives affect learning?

Faulty positives and negatives have a *disproportionately large* contribution to the training updates. To see this, examine the gradients that are computed when optimizing Equation 3.1. The partial derivatives are given as

$$-\frac{\partial L_{\text{xID}}}{\partial \mathbf{v}_i} = \underbrace{\frac{\bar{\mathbf{a}}_i}{\tau}(1 - P(\bar{\mathbf{a}}_i|\mathbf{v}_i))}_{\text{Attraction force}} - \underbrace{\sum_{n \neq i} \frac{\bar{\mathbf{a}}_n}{\tau} P(\bar{\mathbf{a}}_n|\mathbf{v}_i)}_{\text{Repulsion force}} \tag{3.3}$$

$$-\frac{\partial L_{\text{xID}}}{\partial \mathbf{a}_i} = \underbrace{\frac{\bar{\mathbf{v}}_i}{\tau}(1 - P(\bar{\mathbf{v}}_i|\mathbf{a}_i))}_{\text{Attraction force}} - \underbrace{\sum_{n \neq i} \frac{\bar{\mathbf{v}}_n}{\tau} P(\bar{\mathbf{v}}_n|\mathbf{a}_i)}_{\text{Repulsion force}}. \tag{3.4}$$

Intuitively, the target representations $\bar{\mathbf{v}}_i$ and $\bar{\mathbf{a}}_i$ of the instance itself act as 'attraction points' for the encoder of the opposing modality, while the target representations of other (negative) instances, $\bar{\mathbf{v}}_n$ and $\bar{\mathbf{a}}_n$, act as 'repelling points'. For example, in Equation 3.3, the negative gradient pushes $\mathbf{v}_i$ toward $\bar{\mathbf{a}}_i$ and away from $\bar{\mathbf{a}}_n, n \neq i$. The attraction forces are weighed by the complement of the prediction confidence, *i.e.*, $1 - P(\bar{\mathbf{v}}_i|\mathbf{a}_i)$ or $1 - P(\bar{\mathbf{a}}_i|\mathbf{v}_i)$. When positive samples are faulty, these gradients lead to noisy training signals. As show in Figure 3.3, faulty positives tend to have lower similarities and thus less confident predictions. As a result, the cross-modal loss of Equation 3.1 assigns stronger gradients to faulty positive samples. On the other hand, the repelling forces of negative instances are also weighted by the likelihood of matching the base sample, *i.e.* $P(\bar{\mathbf{v}}_n|\mathbf{a}_i)$ and $P(\bar{\mathbf{a}}_n|\mathbf{v}_i)$. However, as shown in Figure 3.4, faulty negatives tend to have high similarity scores, leading to high posteriors $P(\bar{\mathbf{v}}_n|\mathbf{a}_i)$ and $P(\bar{\mathbf{a}}_n|\mathbf{v}_i)$. Thus, the targets $\bar{\mathbf{v}}_n$ and $\bar{\mathbf{a}}_n$ of faulty negatives act as *strong* repelling forces for $\mathbf{a}$ and $\mathbf{v}$ (see Equation 3.3-3.4), even though they should ideally be close in feature space.

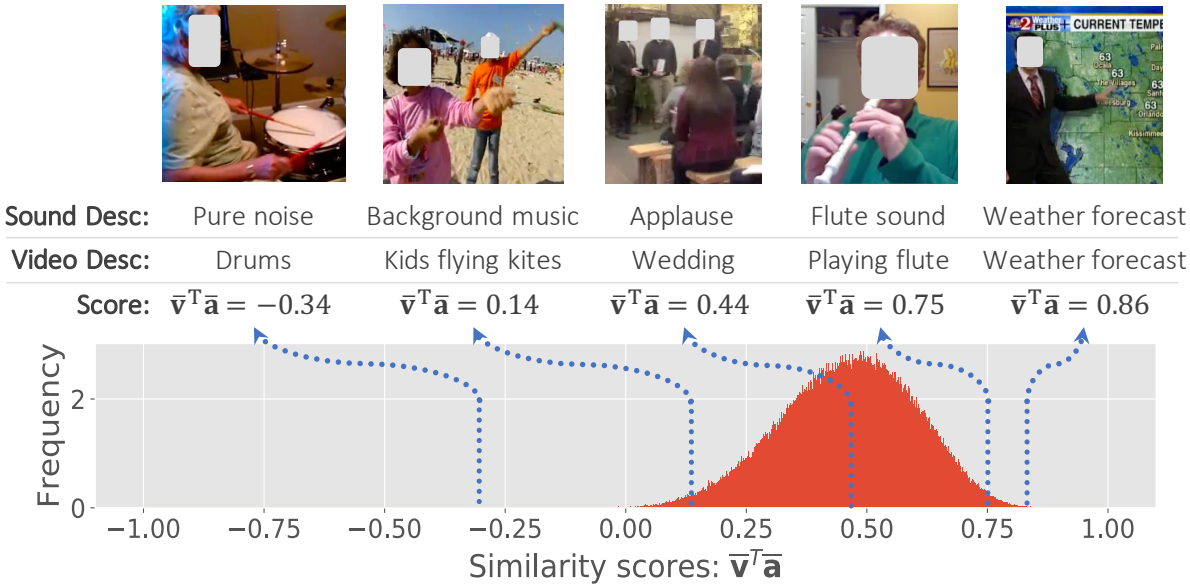


Figure 3.3: Faulty positives in a pretrained cross-modal model. Histogram of similarity scores $\bar{\mathbf{v}}_i^T \bar{\mathbf{a}}_i$ between video and audio representations, and examples obtained at various points of the distribution.

3.4 Robust audio-visual representation learning

We have seen that contrastive learning places too much emphasis on the impossible goals of bringing together the audio-visual components of faulty positives and repelling the feature representations from faulty negatives. We next propose solutions to these two problems.

3.4.1 Weighted xID: Tackling Faulty Positives

To reduce the impact of faulty positives, we propose to optimize a weighted loss. Let $w_i \in [0,1]$ be a set of sample weights that identify faulty positives. Robustness is achieved by re-weighting the xID loss of Equation 3.1

$$\mathcal{L}_{\mathrm{RxID}} = \frac{\sum_i w_i \mathcal{L}_{\mathrm{xID}}(\mathbf{v}_i, \mathbf{a}_i)}{\sum_i w_i}. \tag{3.5}$$

To estimate sample weights w_i, we leverage observations from Figure 3.3. Since low similarities $\bar{\mathbf{v}}_i^T \bar{\mathbf{a}}_i$ are indicative of faulty positives, we define the weights w_i to be proportional to the cumulative distribution of these scores. We assume the scores to be normally distributed and define w_i as

$$w_i = t_{w_{\min}}\left(C_{\mathcal{N}}\left(\bar{\mathbf{a}}_i^T \bar{\mathbf{v}}_i; \mu + \delta\sigma, \kappa\sigma^2\right)\right), \tag{3.6}$$

where μ and σ^2 are the sample mean and variance of the scores, $C_{\mathcal{N}}$ is the cumulative distribution of a transformed normal distribution $\mathcal{N}(\mu + \delta\sigma, \kappa\sigma^2)$, and $t_{w_{\min}}(x) = x \cdot (1 - w_{\min}) + w_{\min}$ is a soft truncation function used to assign a non-zero weight $w_{\min}$ to low score instances. δ, κ and $w_{\min}$ are shape hyper-parameters that provide flexibility to the weight function, adjusting the location and rate of decay of the weights. Figure 3.5 shows how the weighting function varies with the shape hyper-parameters δ, κ and $w_{\min}$.

3.4.2 Soft Targets: Tackling Faulty Negatives

As observed in §3.3, faulty negatives are overemphasized during training. The underlying reason is that the xID loss of Equation 3.1 has too strict a definition of negatives: every negative instance $j \neq i$ is considered 'equally negative.' To limit the impact of faulty negatives, we introduce a 'softer' definition by introducing soft targets $T(j|i)$, based on the similarity between instance i and negative j. We then minimize a soft-xID loss

$$\begin{aligned} \mathcal{L}_{\text{Soft-xID}}(\mathbf{v}_i, \mathbf{a}_i) = &- \textstyle\sum_j T_v(j|i) \log P(\bar{\mathbf{a}}_j | \mathbf{v}_i; \tau) \\ &- \textstyle\sum_j T_a(j|i) \log P(\bar{\mathbf{v}}_j | \mathbf{a}_i; \tau) \end{aligned} \tag{3.7}$$

$$T_v(j|i) = (1 - \lambda)\mathbf{1}_{i=j} + \lambda S_v(j|i) \tag{3.8}$$

$$T_a(j|i) = (1 - \lambda)\mathbf{1}_{i=j} + \lambda S_a(j|i) \tag{3.9}$$

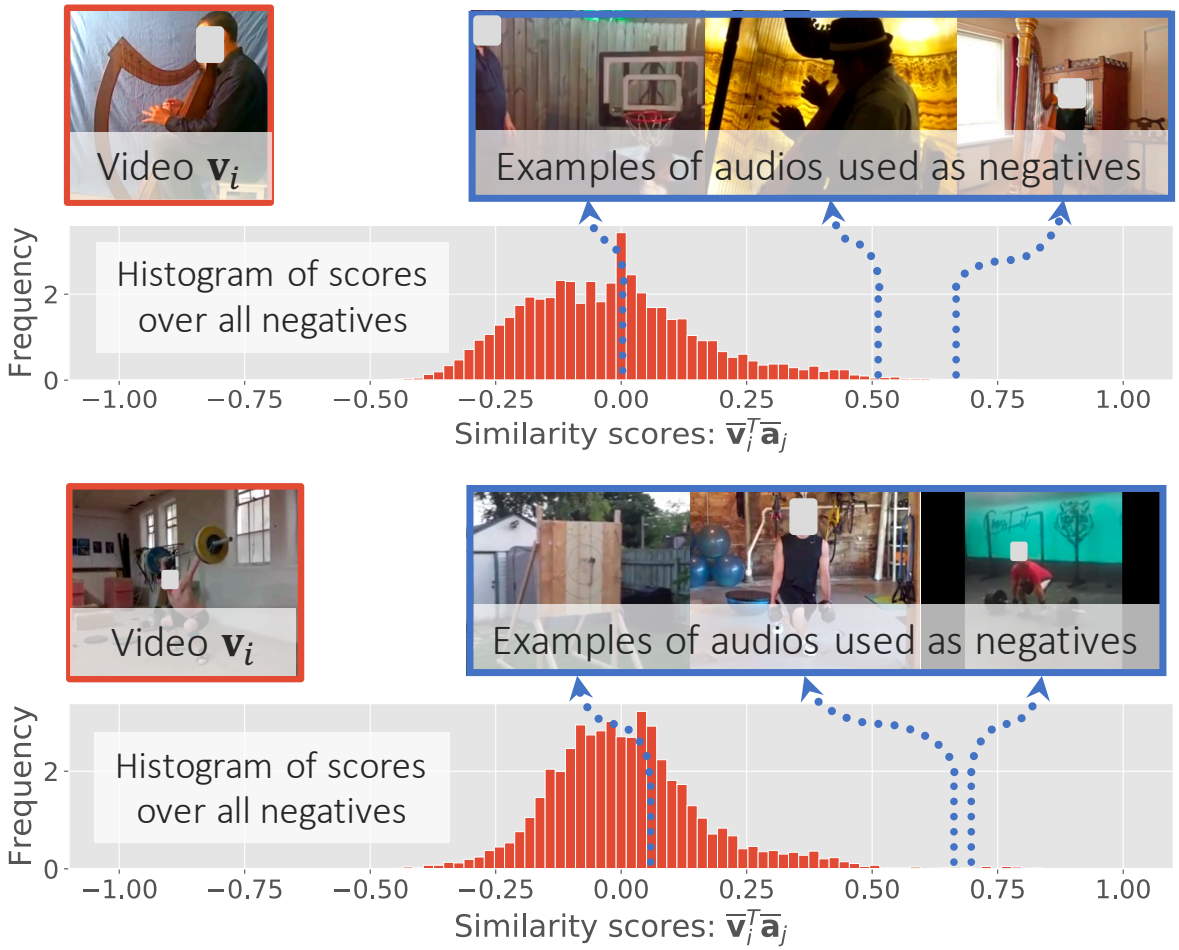


Figure 3.4: Faulty negatives in a pretrained cross-modal model. Two instances $\mathbf{v}_i$ and the corresponding negatives used by a xID model sorted by their similarity scores. The actual videos are provided in supplementary material. xID often uses faulty negatives for contrastive learning.

where $\mathbf{1}_{i=j}$ is the one-hot targets of vanilla xID, S_v and $S_a \in [0,1]$ are softening scores (described next) used to adjust the one-hot targets, and $\lambda \in [0,1]$ is a mixing coefficient that weighs the two terms. Equations 3.1 and 3.7 are identical when $\lambda = 0$. Since $T(j|i)$ is no longer strictly zero for similar instances, minimizing Equation 3.7 reduces the force to repel faulty negatives and thus their impact.

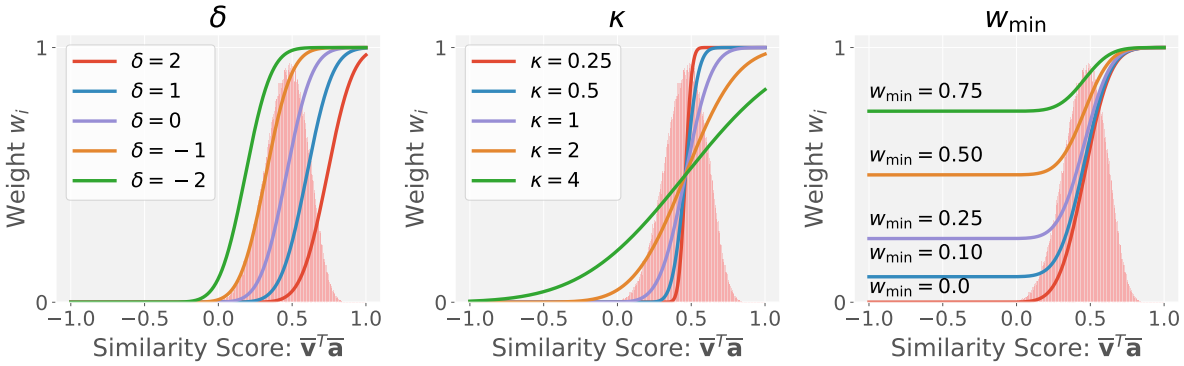


Figure 3.5: Weights as function of similarity scores $\bar{\mathbf{v}}_i^T \bar{\mathbf{a}}_i$ for different values of shape parameters δ, κ and $w_{\min}$. Parameters μ, σ are automatically determined from the histogram of similarity scores $\bar{\mathbf{v}}_i^T \bar{\mathbf{a}}_i$ (shown in red).

Estimating softening scores S

Since our approach focuses on self-supervised learning, we must estimate the softening scores S automatically, *i.e.*, without class labels. We describe multiple strategies for estimating these values and illustrate them in Figure 3.6.

- **Bootstrapping** [167] is a well established procedure to create soft targets. It uses the model's own predictions (posteriors) as the softening scores, *i.e.*,

$$S_v(j|i) = P(\bar{\mathbf{a}}_j|\bar{\mathbf{v}}_i;\tau_s) \text{ and } S_a(j|i) = P(\bar{\mathbf{v}}_j|\bar{\mathbf{a}}_i;\tau_s), \tag{3.10}$$

where τ_s controls the peakiness of the distribution. However, bootstrapping computes the target distribution without aggregating information from any other source other than each model's own posterior.

- **Swapped prediction** improves upon bootstrapping by using the posteriors of the opposite modality, *i.e.*, the softening scores S_v for the video modality are computed using the posterior of the audio encoder and vice-versa,

$$S_v(j|i) = P(\bar{\mathbf{v}}_j|\bar{\mathbf{a}}_i;\tau_s) \text{ and } S_a(j|i) = P(\bar{\mathbf{a}}_j|\bar{\mathbf{v}}_i;\tau_s). \tag{3.11}$$

As a result, in addition to the instance itself, the model is asked to predict which other instances are deemed similar in the opposite modality.

• **Neighbor prediction** relies on within-modal relationships to estimate the similarity between instances, thus avoiding potential mismatched audio and visual modalities when computing the soft targets. Specifically, we define

$$S_v(j|i) = \rho(\bar{\mathbf{v}}_i^T \bar{\mathbf{v}}_j/\tau_s) \text{ and } S_a(j|i) = \rho(\bar{\mathbf{a}}_i^T \bar{\mathbf{a}}_j/\tau_s), \tag{3.12}$$

where ρ is the softmax operator.

• **Cycle consistent prediction** improves upon 'swapped prediction' by focusing on negatives that are good correspondences themselves, *i.e.*, negatives with high similarity scores $\bar{\mathbf{v}}_j^T \bar{\mathbf{a}}_j$. In this case, we define

$$S_v(j|i) = \rho(\bar{\mathbf{v}}_i^T \bar{\mathbf{a}}_i/\tau_t + \bar{\mathbf{a}}_i^T \bar{\mathbf{v}}_j/\tau_s + \bar{\mathbf{v}}_j^T \bar{\mathbf{a}}_j/\tau_t) \tag{3.13}$$

$$S_a(j|i) = \rho(\bar{\mathbf{a}}_i^T \bar{\mathbf{v}}_i/\tau_t + \bar{\mathbf{v}}_i^T \bar{\mathbf{a}}_j/\tau_s + \bar{\mathbf{a}}_j^T \bar{\mathbf{v}}_j/\tau_t) \tag{3.14}$$

where τ_s and τ_t control the relative importance of swapped prediction target and avoiding negatives with weak correspondences. As shown in Figure 3.6, the terms $\bar{\mathbf{v}}_i^T \bar{\mathbf{a}}_i$ and $\bar{\mathbf{v}}_j^T \bar{\mathbf{a}}_j$ complete a cycle over instances i and j.

How do soft targets mitigate faulty negatives?

The soft xID loss of Equation 3.7 prevents overemphasizing faulty negatives by relying on soft targets $T(j|i)$ that encode similarities between instances. To better understand the mechanism,

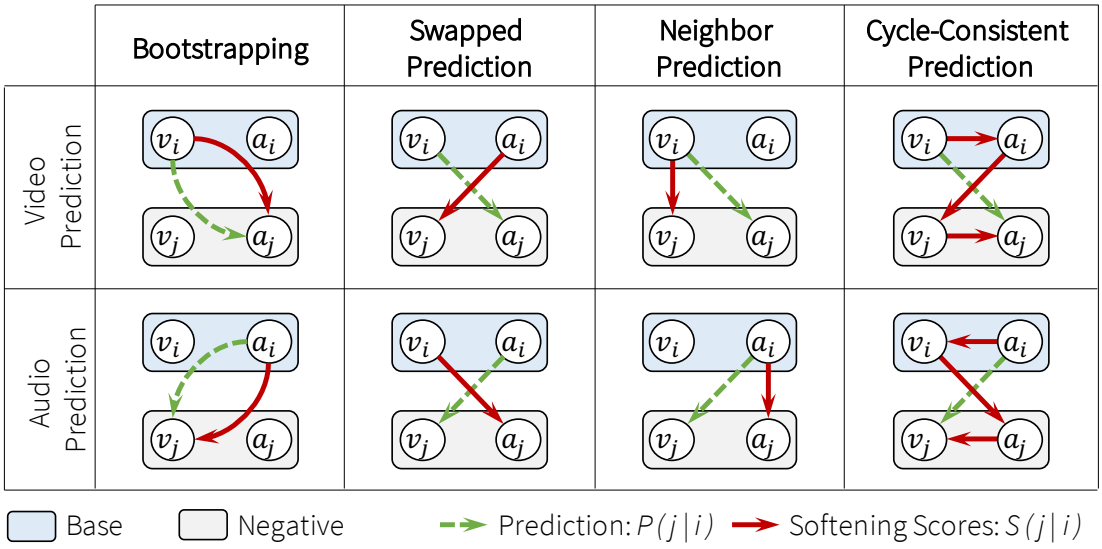


Figure 3.6: Strategies to estimate softening scores $S(i|j)$.

we examine the partial derivatives of the soft-xID loss:

$$-\frac{\partial L_{\text{Soft-xID}}}{\partial \mathbf{v}_i} = \sum_j \frac{\bar{\mathbf{a}}_j}{\tau}(T_v(j|i) - P(\bar{\mathbf{a}}_j|\mathbf{v}_i)) \tag{3.15}$$

$$-\frac{\partial L_{\text{Soft-xID}}}{\partial \mathbf{a}_i} = \sum_j \frac{\bar{\mathbf{v}}_j}{\tau}(T_a(j|i) - P(\bar{\mathbf{v}}_j|\mathbf{a}_i)). \tag{3.16}$$

Since faulty negatives j tend to be similar to the base instance i, the soft targets $T(j|i)$ are higher. Thus, the target representations $\bar{\mathbf{v}}_j$ and $\bar{\mathbf{a}}_j$ of faulty negatives act as weaker negatives, or even as positives when $T(j|i)$ is larger than the model posteriors.

3.4.3 Training

We introduced two procedures to deal with noisy training signals inherent to cross-modal instance discrimination. §3.4.1 presents a weighting mechanism that limits the effect of faulty positives, while §3.4.2 proposes a soft instance discrimination loss that predicts relations between instances, thus preventing the training algorithm from overemphasizing faulty negatives. Since both procedures rely on the alignment between audio and visual target representations to find

weak correspondences, we start by training the model for cross-modal instance discrimination alone (Equation 3.1). After the initial warmup stage, the two procedures can be combined by minimizing

$$\mathcal{L} = \frac{1}{\sum_k w_k} \sum_i w_i \mathcal{L}_{\text{Soft-xID}}(\mathbf{v}_i, \mathbf{a}_i) \tag{3.17}$$

where w_i are the sample weights of Equation 3.6 and $\mathcal{L}_{\text{Soft-xID}}$ is the xID loss with soft targets of Equation 3.7.

3.5 Experiments

We perform experiments to better understand cross-modal learning and validate the proposed improvements. We pretrain models on a subset of the Kinetics-400 [196] dataset containing 50K videos and evaluate the pretrained models by transfer learning.

3.5.1 Experimental Setup

Video and audio preprocessing

During training, we extract video clips of length $T = 8$ frames and resolution 80×80 at 16 fps. Video clips are augmented using temporal jittering, multi-scale cropping, horizontal flipping, color jittering, gray-scaling, and Gaussian blur [32]. All data augmentations are applied consistently over all frames. For the audio, we extract mono clips of length 2s at a sample rate of 11025Hz, and compute log spectrograms on 50ms windows with a hop size of 25ms. The spectrogram is then converted to a mel scale with 80 bands, yielding an audio input of size 80×80. Audio data is augmented by randomly changing the volume by at most 20%.

Video and audio models

The video encoder is a 9-layer version of the R(2+1)D model of [191]. Following [8, 138], we replaced global average pooling with max pooling. The audio encoder is a 9-layer 2D ConvNet with batch normalization and global max pooling. Both encoders yield 512-dimensional features, which are mapped into a 128-dimensional sphere using a non-linear projection head (as in [32]) followed by L2 normalization.

Pretraining

In the warm-up stage, the video and audio models are trained to optimize the loss of Equation 3.1 using the Adam optimizer [101] with default hyper-parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) for 400 epochs with a learning rate of $1e-4$ and a batch size of 224 split over 2 12Gb GPUs. In order to reduce the memory footprint of our models, we employ mixed-precision training [132] using PyTorch AMP [152]. Following [138, 206], the audio and video target representations, $\bar{\mathbf{a}}$ and $\bar{\mathbf{v}}$, are generated using memory banks updated by exponential moving average with an update constant of 0.5. The contrastive loss of Equation 3.1 is defined by opposing the target representation of the opposite modality to 1024 negatives randomly drawn from the memory bank. The temperature hyper-parameter is set to $\tau = 0.07$.

After the initial warm-up stage, models are trained for an additional 200 epochs to optimize the loss of Equation 3.17 using the Adam optimizer and a cosine learning rate schedule starting at $1e-4$ and ending at $1e-5$. The hyper-parameters for the weighting function (Equation 3.6) and the soft xID loss (Equation 3.7) are discussed below. To provide a fair comparison to the AVID baseline [138], we control for the number of epochs by training the baseline model for an additional 200 epochs as well.

Downstream tasks

We evaluate audio and video features using transfer learning. Video features are evaluated on the UCF [182] and HMDB [109] datasets. Models are fine-tuned using 8-frame clips for 200 epochs using the Adam optimizer with a batch size of 192 on a single GPU and a cosine learning rate schedule starting at $1e-4$ and ending at $1e-5$. To prevent overfitting, we use dropout after the global max-pooling layer, weight decay of $1e-3$, and reduced the learning rate for backbone weights by a factor of 10. At test time, top-1 accuracy is measured on video level predictions computed by averaging the predictions of 10 clips uniformly sampled over the entire video.

Following [11, 210], we also evaluate the quality of video representations by conducting retrieval experiments without fine-tuning. Feature maps of size $4\times4\times512$ are extracted from 10 clips per video and averaged. We then use videos in the test set to query the training set. As in [11, 210], a correct retrieval occurs when the class of one of the top-k retrieved videos matches the query, and performance is measured by the average top-k retrieval performance ($R@K$).

3.5.2 Weighted cross-modal learning

We analyze the impact of faulty positives on the representations learned by cross-modal instance discrimination.

Faulty positives are detrimental to representation learning

We artificially control the number of faulty positives to assess their impact on representation learning. The pretraining dataset *already contains* an unknown (but significant) number of faulty positives. We increase this number by injecting more faulty positives. A faulty positive is injected by replacing the audio of an instance with a randomly selected audio that is not part of the training set. After pretraining, the learned visual representation is evaluated on the UCF and HMDB datasets using both classification and retrieval protocols. Figure 3.7 shows that

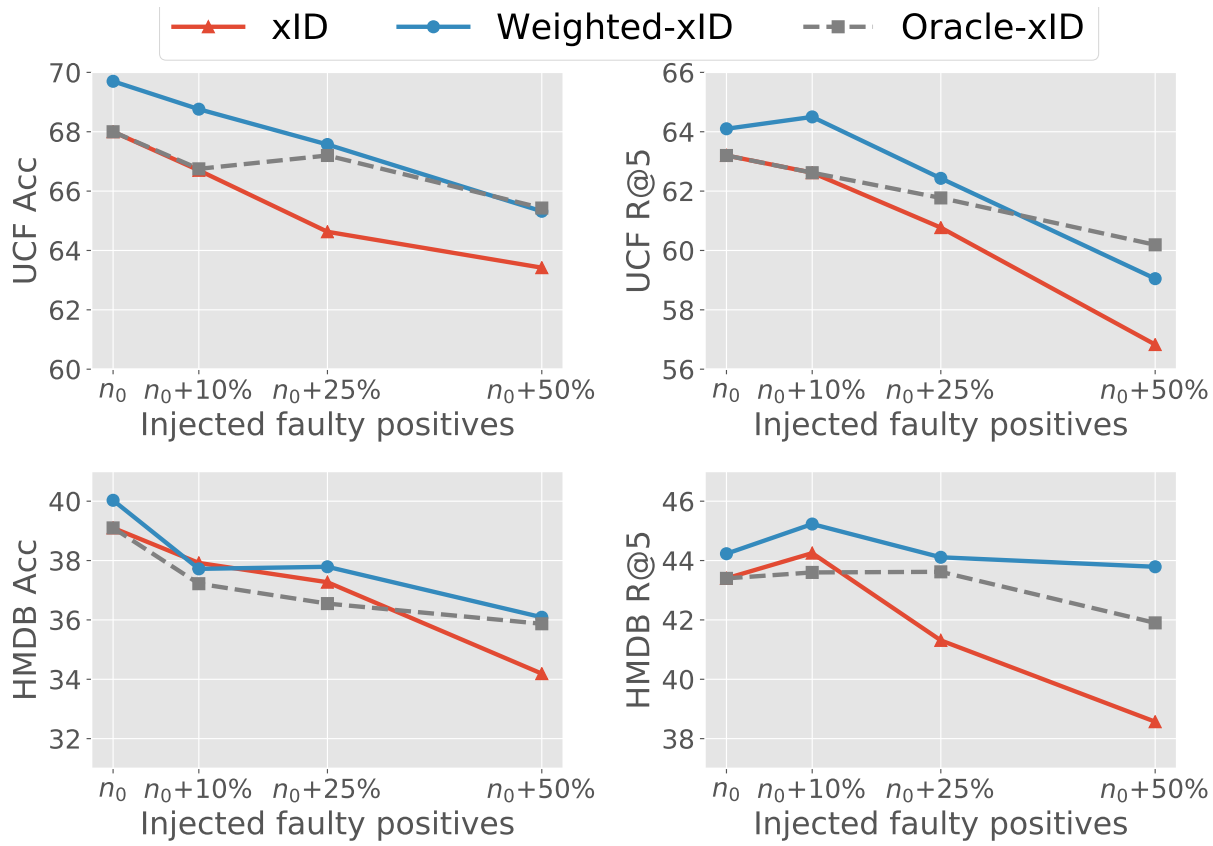


Figure 3.7: Transfer learning performance with and without faulty positives. The weighted loss (Weighted-xID) is less sensitive to faulty positives.

as the fraction of faulty positives increases, the transfer performance of cross-modal instance discrimination (xID) decreases significantly.

Weighted xID reduces the impact of faulty positives

We evaluate the effectiveness of the weighted xID loss (Equation 3.5) as a function of the number of faulty positives. We compare the representations learned by Weighted-xID to its unweighted counterpart (xID), as well as an oracle weight function (Oracle-xID) which assigns $w_i = 0$ to artificially altered instances and $w_i = 1$ otherwise. The weight function of Equation 3.5 is defined with $\kappa = 0.5$ and $w_{\min} = 0.25$. For simplicity, we assume that the noise level is known and set δ in Weighted-xID so that the midpoint of the weighting function coincides with the

Table 3.1: Different strategies for computing soft targets in the pretraining loss of Equation 3.7.

Target Distribution	UCF		HMDB	
	Acc	R@5	Acc	R@5
Oracle*	73.6	76.0	45.4	53.6
xID [138]	68.0	63.2	39.0	43.4
Bootstrapping	69.2	64.4	40.5	44.7
Neighbor Pred.	**70.5**	65.4	41.2	45.0
Swapped Pred.	70.0	64.9	41.3	45.4
CCP	70.3	**65.9**	**41.5**	**45.5**

**Uses class labels to generate target distribution.*

known fraction of altered samples. In practice, the noise level would need to be estimated either by cross-validation or by manual inspection. Weighted-xID is not very sensitive to these parameters (see appendix).

Figure 3.7 shows the performance of the three approaches. Oracle-xID consistently outperforms xID when the fraction of injected faulty positives is high. This shows that the detrimental impact of noisy correspondences can be mitigated with a weighting strategy. Weighted-xID also outperforms the unweighted version (xID) in nearly all cases, with larger margins for larger fractions of noisy correspondences. In fact, Weighted-xID even outperforms the oracle weight function, especially at lower noise levels. This is because the original Kinetics dataset *already contains* a significant amount of weak correspondences, which the oracle weight function treats as clean $w_i = 1$, while the weighting function of Equation 3.6 can suppress them.

3.5.3 Instance discrimination with soft targets

To limit the impact of faulty negatives, we proposed to match a soft target distribution that encodes instance similarity. We analyze different design decisions for creating the soft targets and their effect on transfer performance.

Comparison of strategies for computing targets

As summarized in Figure 3.6, the soft target distributions can be computed by aggregating evidence from all modalities. Four different strategies were proposed, bootstrapping, swapped or cycle consistent assignments. Models were trained to minimize the loss of Equation 3.7 with $\lambda = 0.5$. We empirically found that peakier target distributions work better, and set the temperature parameter τ_s to 0.02. For cycle consistent assignments, the terms $\bar{\mathbf{v}}_j^T \bar{\mathbf{a}}_j$ are used so as to focus on negatives that are good correspondences themselves. A temperature hyper-parameter of $\tau_t = 0.07$ was sufficient to impose such constraint. Beyond the baseline xID, we also compare to an *oracle* target distribution that has access to class labels to determine the similarity between instances. Specifically, the oracle considers two instances i and j to be similar if they share the same class label, and computes $T_v(j|i)$ and $T_a(j|i)$ by assigning a uniform distribution over similar instances, and 0 to non-similar ones.

Table 3.1 shows the performance of different target distributions. We observe a large gap between vanilla xID and xID with an oracle soft target, which demonstrates the detrimental effect of faulty negatives. In the self-supervised case, however, labels are not available for determining the target distribution. Nevertheless, the estimated target distributions (bottom four rows) still significantly improve over the xID loss. Regarding the various types of target distributions, bootstrapping is the least effective. This is expected since, in this case, the target distribution is a peakier version of the model posterior, *i.e.* it is obtained without aggregating information from any other sources. Cycle consistent prediction is the most effective most often. This is because cycle consistent prediction not only leverages the opposite modality to create the target distribution, but it also avoids targets that are not good correspondences themselves, *i.e.*, avoids samples with low cross-modal similarities.

Table 3.2: Combining weighted xID loss with soft targets.

Method	Robust Weighting	CCP Soft Targets	UCF		HMDB	
			Acc	R@5	Acc	R@5
xID [138]	✗	✗	68.0	63.2	39.0	43.4
Weighted-xID	✓	✗	69.7	64.1	40.1	44.3
Soft-xID	✗	✓	70.3	65.9	41.5	45.5
Robust-xID	✓	✓	**71.6**	**67.4**	**41.9**	**46.2**

3.5.4 Robust instance discrimination with soft targets

Sample weighting and soft targets are designed to address two different sources of noisy training signals inherent to cross-modal contrastive learning: faulty positives and faulty negatives. Table 3.2 shows that the two proposed improvements (Weighted-xID and Soft-xID) not only improve upon the representations of vanilla xID, they are also complementary to each other. By combining the two approaches using the loss of Equation 3.17, Robust-xID improved upon Weighted and Soft-xID.

3.6 Comparison to prior work

We compare Robust-xID to prior work in self-supervised learning. We train our models on the Kinetics dataset, using an 18-layer R(2+1)D model [191] for the video, and a 9-layer 2D ConvNet with batch normalization for the audio. Video clips of length 8-frames and 112×112 resolution are extracted at 16fps, and the same data augmentations from §3.5 are used. We extract audio clips of length 2s at 24KHz and compute log mel spectrograms with 128 time steps and 128 frequency bands. All models are trained with the Adam optimizer with a batch size of 512 distributed across 8 12Gb GPUs. We warm-up the models for 200 epochs by training on the xID loss alone with a learning rate of $5e-4$. The models are then trained with sample weights and cycle consistent soft targets for an additional 200 epochs using a cosine learning rate schedule from $5e{-}4$ to $5e{-}5$.

After pre-training, models are evaluated on UCF and HMDB. We fine-tune the models using either 8 or 32 frame clips for action recognition and report the top-1 accuracy of video level predictions (with 10 clips per video) in Table 3.3. The proposed procedure outperformed all prior work where pretraining is limited to a single node (8 GPUs), and even outperformed methods like SeLaVi, which require 8× more compute for training. We also conducted a close comparison to the CMA procedure of [138] (xID+CMA). While CMA can also partially address the problem of faulty negatives, Robust-xID showed better performance. Robust-xID is also easier to implement as it identifies both faulty positives and negatives in a simpler online fashion. We note that xID+CMA is a faithful implementation of AVID+CMA [138], as it follows the original code with improved data augmentations. However, the results reported for xID+CMA are lower than those originally reported in [138] because 1) distributed training was conducted on 8 GPUs instead of 64 (large batch sizes are known to have a substantial impact on contrastive learning performance [32, 33, 29]), and 2) [138] is trained and evaluated with videos of higher resolution (224 instead of 112). By training the proposed model with a larger batch size, we expect the performance to improve further.

We also compare the learned representations to prior work without fine-tuning. Following [11, 155], we conducted retrieval experiments, and report the retrieval performance $R@K$ for $K = 1$, $K = 5$ and $K = 20$ neighbors in Table 3.4. The retrieval protocol was described in §3.5. Following [94, 155], we also assessed the few-shot learning performance of Robust-xID models on UCF and HMDB. For the few-shot evaluation, we average the pretrained max-pooling features of 10 clips per video. The features from n videos per class are then used to learn a one-vs-all linear SVM classifier with $C = 1$. We report the top-1 accuracy averaged over 50 trials in Table 3.5. On both the retrieval and few-shot learning tasks, Robust-xID improves significantly over all prior work, reaffirming the importance of mitigating the training noise introduced by faulty positives and faulty negatives.

Table 3.3: Comparison to prior work (finetuning). Performance on the downstream UCF and HMDB datasets by full network fine-tuning after pre-training on Kinetics.

Method	Model	Compute # GPUs	Finetuning Resolution	UCF	HMDB
DPC [72]	S3D	4	25×128^2	75.7	35.7
CBT [186]	S3D	8	16×112^2	79.5	44.6
Multisensory [148]	3D-ResNet18	3	32×224^2	82.1	–
AVTS [105]	MC3-18	4	25×224^2	84.1	52.5
SeLaVi [11]	R(2+1)D-18	64	32×112^2	83.1*	47.1*
XDC [4]	R(2+1)D-18	64	8×224^2	74.2*	39.0*
	R(2+1)D-18	64	32×224^2	86.8*	52.6*
AVID-CMA [138]	R(2+1)D-18	64	8×224^2	83.7*	49.5*
	R(2+1)D-18	64	32×224^2	87.5*	60.8*
GDT [155]	R(2+1)D-18	64	32×224^2	89.3*	60.0*
xID+CMA [138]	R(2+1)D-18	8	8×112^2	80.6	48.6
	R(2+1)D-18	8	32×112^2	84.9	54.7
Robust-xID	R(2+1)D-18	8	8×112^2	81.9	49.5
	R(2+1)D-18	8	32×112^2	**85.6**	**55.0**

* *Models pre-trained with more than one compute node (8 GPUs).*

Table 3.4: Retrieval performance on UCF and HMDB datasets after pre-training on Kinetics for different numbers of retried neighbors.

Method	UCF			HMDB		
	R@1	R@5	R@20	R@1	R@5	R@20
SpeedNet [18]	13.0	28.1	49.5	-	-	-
VCP [126]	18.6	33.6	53.5	7.6	24.4	53.6
VSP [35]	24.6	41.9	76.9	10.3	26.6	54.6
CoCLR [74]	55.9	70.8	82.5	26.1	45.8	69.7
SeLaVi [11]	52.0	68.6	84.5	24.8	47.6	75.5
GDT [155]	57.4	73.4	88.1	25.4	51.4	75.0
xID+CMA [138]	60.1	76.6	90.1	29.7	54.4	77.1
Robust-xID	**60.9**	**79.4**	**90.8**	**30.8**	**55.8**	**79.7**

3.7 Discussion and future work

We identified and tackled two significant sources of noisy training signals in audio-visual instance discrimination, namely instances with weak audio-visual correspondence (or

Table 3.5: Few-shot learning on UCF and HMDB after pre-training on Kinetics. Classification is conducted using a one-vs-all SVM trained on the pretrained features of n images per class.

Method	UCF			HMDB		
	1-shot	5-shot	20-shot	1-shot	5-shot	20-shot
3D-RotNet [94]	15.0	31.5	47.1	-	-	-
GDT [155]	26.3	42.4	49.4	13.4	15.6	20.8
xID+CMA [138]	30.8	53.1	66.9	13.5	25.0	33.6
Robust-xID	**32.8**	**54.6**	**67.8**	**14.1**	**25.9**	**34.9**

faulty positives) and semantically similar negatives (or faulty negatives). We showed the impact of faulty correspondences on representation learning by removing them using an oracle with access to ground-truth annotations. We then proposed a method that mitigates the impact of faulty correspondences without relying on ground-truth annotations. Extensive analysis and experimental evaluations show that the proposed procedure enhances representation learning and improves transfer performance significantly.

Our findings show that cross-modal learning should be seen as a problem of learning with noisy targets. While we propose two specific methods to address faulty positives and faulty negatives (*i.e*. weighting and soft targets), there is a rich literature regarding supervised learning with noisy labels. Developing methods that tackle noisy correspondences are a promising avenue for future research. Furthermore, we focused on audio-visual learning, but other pairs of modalities such as RGB and flow or text from instructional videos also present similar problems. We believe that our method will also benefit cross-modal learning from other modalities.

3.8 Appendix

3.8.1 Parametric studies

We provide a parametric study of key Robust-xID hyper-parameters.

Weight function shape parameter δ

One critical parameter of Weighted-xID is the shape parameter δ, which specifies the mid-point location of the weight function. For example, when $\delta = -2$, the midpoint is located at $\mu - 2\sigma$ where μ and σ are the sample mean and standard deviation of the scores $\bar{\mathbf{v}}_i^T \bar{\mathbf{a}}_i$. This means that for $\delta = -2$, the majority of samples will have a weight of 1, and only a small fraction will have a weight close to $w_{\min}$. As δ increases, the proportion of samples that are down-weighted also increases. To study the impact of δ, we trained several models using Weighted-xID with different values of δ and for different amounts of injected faulty positives n_0. Other hyper-parameters were kept at their default values $w_{\min} = 0.25$ and $\kappa = 0.5$. The transfer performance is shown in Figure 3.8. As can be seen, the proposed robust xID procedure is not very sensitive to this hyper-parameter. This suggests that Robust-xID can help representation learning as long as clear faulty positives are suppressed.

Soft-xID: Mixing coefficient

The mixing coefficient λ specifies the degree to which the one-hot targets of instance discrimination are softened in Soft-xID. The one-hot instance discrimination targets are used when $\lambda = 0$. As λ increases, the softening scores $S(j|i)$ are increasingly used to adjust the one-hot targets. To study the impact of the mixing coefficient λ, we trained several models using Soft-xID with various values of λ. Cycle consistent targets were used as the softening strategy. Figure 3.9 shows the transfer performance of the learned models on UCF and HMDB under the fine-tuning and retrieval protocols. The trend is consistent across the two datasets and two evaluation protocols. Softening the instance discrimination targets enhances representation learning, with the optimal performance achieved with a mixing coefficient between 0.25 and 0.5. However, as the mixing coefficient increases substantially $\lambda > 0.65$, the targets are derived from the model prediction alone and disregard instance labels. In this case of large λ, the pre-training fails completely, *i.e.*, the learned representations have very low transfer performance.

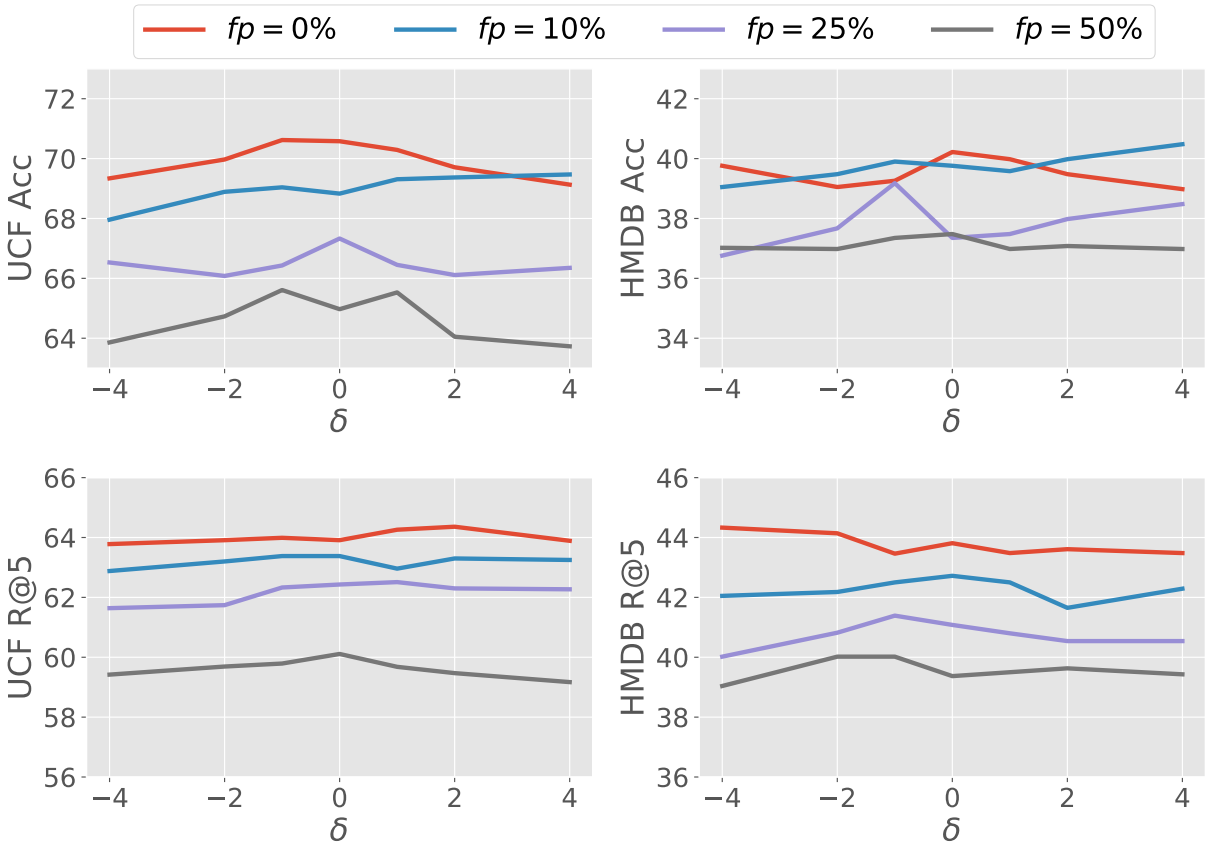


Figure 3.8: Effect of shape parameter δ in Weighted-xID. Transfer learning performance is evaluated on two datasets (UCF and HMDB) under two protocols (full finetuning and retrieval).

3.8.2 Additional analysis

The proposed approach learns high-quality feature representations that can be used to discriminate several action classes. This was shown in the main paper by reporting transfer learning results. We now provide additional qualitative evidence and analysis.

Retrieval

For each video, we extracted $4 \times 4 \times 512$ feature maps from the video encoder learned using Robust-xID on the full Kinetics dataset. Figure 3.11 depicts the top 4 closest videos for several query samples. As can be seen, Robust-xID produces highly semantic features, enabling correct retrievals for a large number of videos spanning a large number of classes. Furthermore,

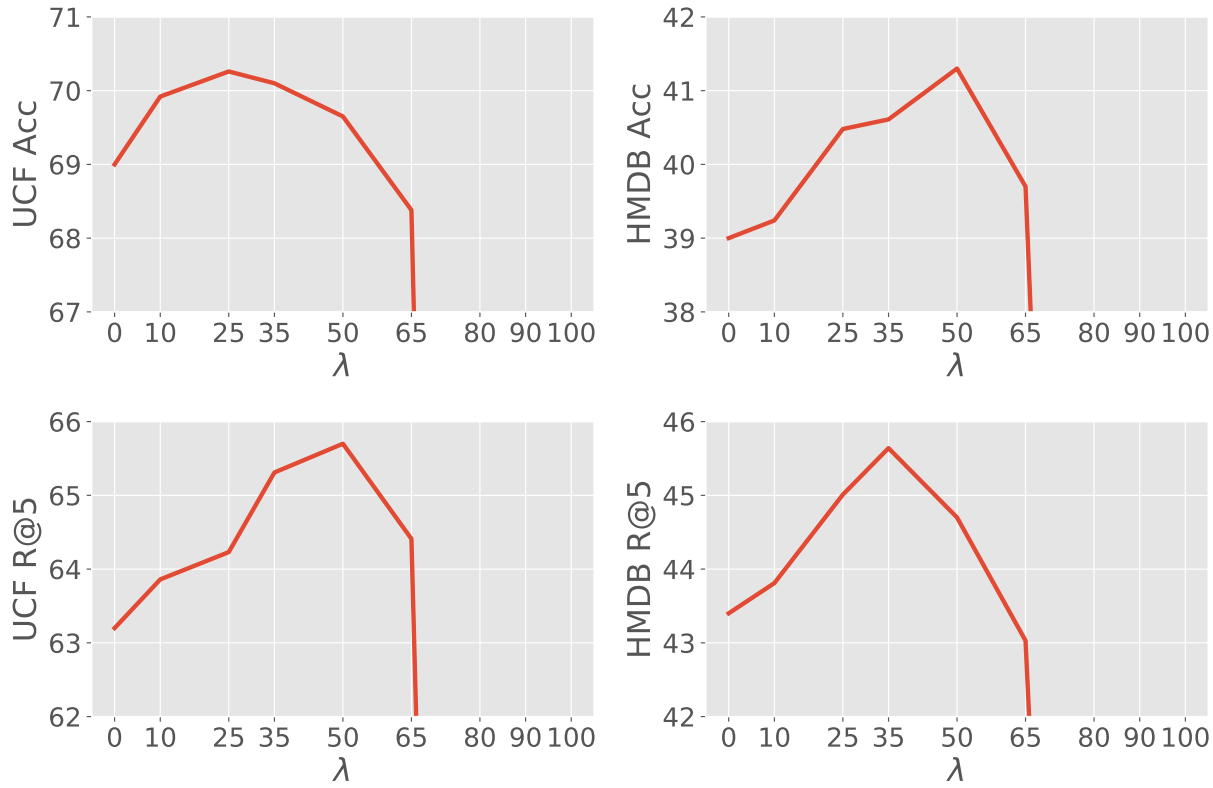


Figure 3.9: Effect of mixing coefficient λ in Soft-xID. Transfer learning performance is evaluated on two datasets (UCF and HMDB) under two protocols (full finetuning and retrieval).

even when a video of a different class is retrieved, the errors are intuitive (for example, the confusion between 'American football' and 'Hurling' in the third row). Failure cases also seem to be correlated with classes that are hard to distinguish from the audio alone (eg, different types of kicking sports or swimming strokes).

Class-based analysis

To better understand which classes are better modeled by the Robust-xID framework, we measured the top-1 retrieval performance ($R@1$) averaged across all images of each class. Similar to the analysis above, each video is represented by a $4 \times 4 \times 512$ feature map extracted from a video encoder learned using Robust-xID on the full Kinetics dataset. Figure 3.10 depicts a list of Kinetics classes sorted by their average $R@1$ score. As can be seen, action classes which are

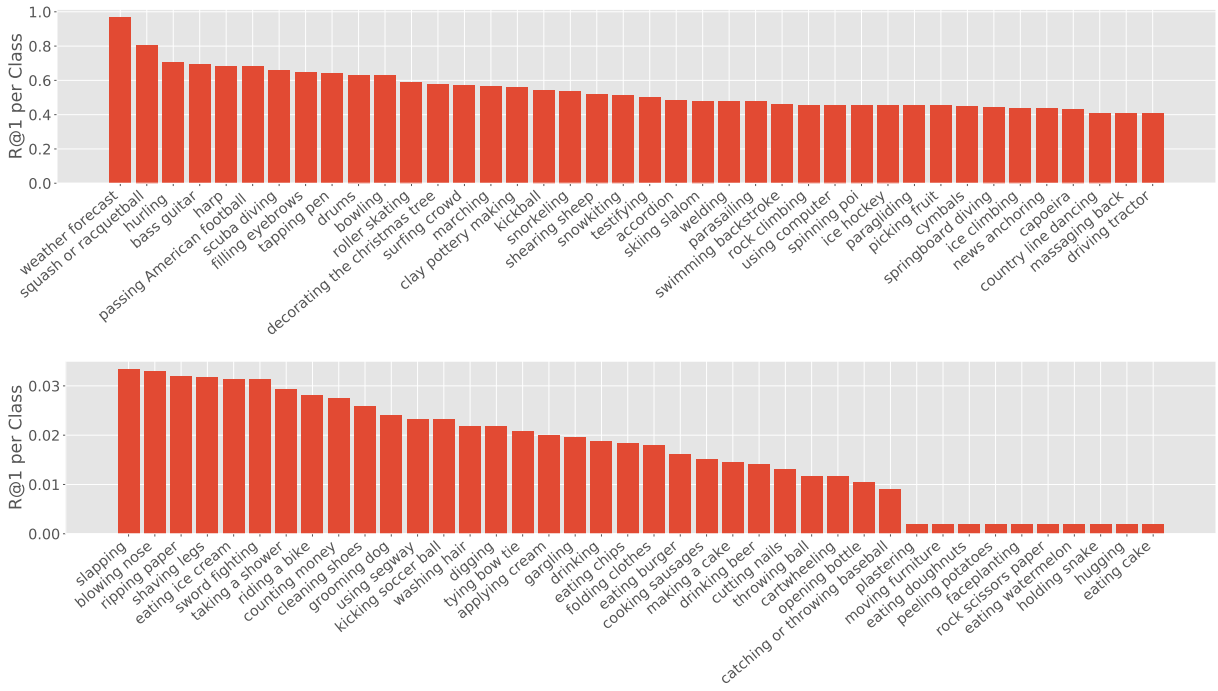


Figure 3.10: Best (top) and worse (bottom) Kinetics classes. For each class, we depict the top-1 retrieval performance ($R@1$) averaged across all images of each class.

often accompanied by long and distinctive sounds (*e.g.*, squash, harp, drums, accordion, or scuba diving) tend to be more easily distinguished from others. In contrast, classes with less distinctive audio (*e.g.*, making a cake, eating cake, or hugging) or classes where distinctive sounds are short-lived (*e.g.*, blowing nose, gargling or kicking ball) are harder to model using a cross-modal audio-visual framework. As a result, the features learned for such classes are less discriminative.

Faulty positive detection performance

To obtain a rough estimate of performance of the faulty positive detection procedure, we randomly sampled 100 videos from the 10000 most likely faulty positives, as identified by Robust-xID trained on the full Kinetics dataset. We then manually labeled them according to how related their audio and visual signals are. From those, 67 were clear faulty pairs; 24 contained narrative voice-overs (*i.e.*, required natural language understanding to link the two modalities); and 9 samples were clearly misidentified.

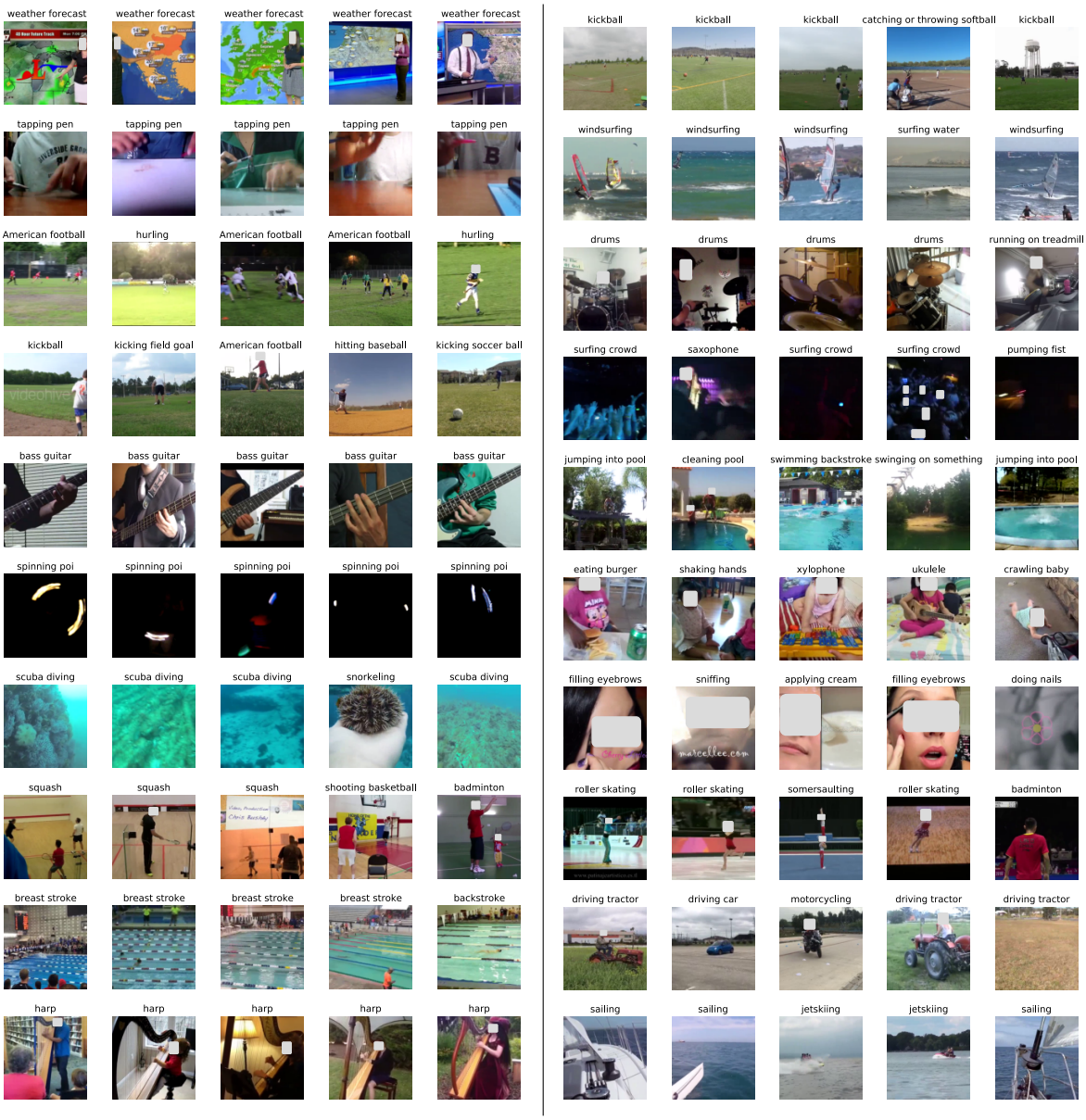


Figure 3.11: Examples of nearest neighbor retrievals. In each row, the first image depicts the query video, and the following four images depict the top 4 retrievals.

3.9 Acknowledgements

Chapter 3 is, in full, based on the material as it appears in the publication of "Robust Audio-Visual Instance Discrimination", Pedro Morgado, Nuno Vasconcelos, Ishan Misra, to appear in the Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition

(CVPR), 2021. The dissertation author was the primary investigator and author of this material.

Chapter 4

Audio-Visual Spatial Alignment

4.1 Introduction

Human perception is inherently multi-sensory. Since real-world events can manifest through multiple modalities, the ability to integrate information from various sensory inputs can significantly benefit perception. In particular, neural processes for audio and visual perception are known to influence each other significantly. These interactions are responsible for several well known audio-visual illusions such as the "McGurk effect" [131], the "sound induced flash effect" [177] or the "fusion effect" [6], and can even be observed in brain activation studies, where areas of the brain dedicated to visual processing have been shown to be activated by sounds that are predictive of visual events, even in the absence of visual input [50, 194].

In computer vision, the natural co-occurrence of audio and video has been extensively studied. Prior work has shown that this co-occurrence can be leveraged to learn representations in a self-supervised manner, i.e., without human annotations. A common approach is to learn to match audio and video clips of the same video instance [8, 7, 138]. Intuitively, if visual events are associated with a salient sound signature, then the audio can be treated as a label to describe the visual content [39]. Prior work has also demonstrated the value of temporal synchronization between audio and video clips for learning representations for downstream tasks such as action recognition [105, 148].

Since these methods do not need to localize sound sources, they struggle to discriminate visual concepts that often co-occur. For example, the sound of a car can be quite distinctive, and thus it is a good target description for the "car" visual concept. However, current approaches use this audio as a descriptor for the whole video clip, as opposed to the region containing the car. Since cars and roads often co-occur, there is an inherent ambiguity about which of the two produce the sound. This makes it is hard to learn good representations for visual concepts like "cars", distinguishable from co-occurring objects like "roads" by pure audio-visual correspondence or temporal synchronization. This problem was clearly demonstrated in [175] that shows the poor

audio localization achieved with AVC pretext training.

To address this issue, we learn representations by training deep neural networks with 1) 360°video data that contain audio-visual signals with strong spatial cues and 2) a pretext task to conduct audio-visual spatial alignment (AVSA, Figure 4.1). Unlike regular videos with mono audio recordings, 360°video data and spatial audio formats like ambisonics fully capture the spatial layout of audio and visual content within a scene. To learn from this spatial information, we collected a large 360°video dataset, five times larger than currently available datasets. We also designed a pretext task where audio and video clips are sampled from different viewpoints within a 360°video, and spatially misaligned audio/video clips are treated as negatives examples for contrastive learning. To enhance the learned representations, two modifications to the standard contrastive learning setup are proposed. First, the ability to perform spatial alignment is boosted using a curriculum learning strategy that initially focus on learning audio-visual correspondences at the video level. Second, we propose to reason over the full spatial content of the 360°video by combining representations from multiple viewpoints using a transformer network. We show the benefits of the AVSA pretext task on a variety of audio and visual downstream tasks, including audio-visual correspondence and spatial alignment, action recognition and video semantic segmentation.

4.2 Related work

360°media. The increasing availability of 360°data has sparked interest in developing vision systems for 360°imagery. For example, the SUN-360 dataset of static 360°images was collected to learn to recognize viewpoints within a scene [207]. Self-supervised monocular depth and camera motion estimation have also been studied by pairing 360°imagery with depth data [198, 157]. Another common topic of interest is to enhance 360°video consumption by guiding the viewer towards salient viewing angles within a video [218, 34], automating the

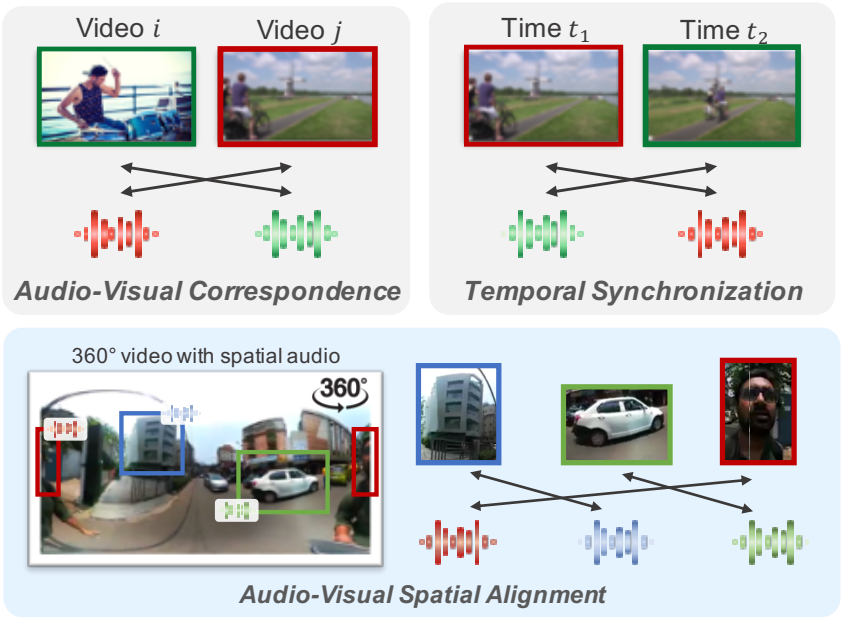


Figure 4.1: Comparison of audio-visual spatial alignment to prior work. Prior work on audio-visual representation learning leverages correspondences at the video level. Instead, we learn representations by performing audio-visual spatial alignment (AVSA) of 360°video and spatial audio.

field-of-view control for 360°video playback [86, 185], or by upgrading mono recordings into spatial sounds [137].

Self-supervised learning. Self-supervised learning methods learn representations without requiring explicit human annotation. Instead of predicting human labels, self-supervision learns representations that are predictive of the input data itself (or parts of it) while imposing additional constraints such as sparsity [117, 145, 144] or invariance [70, 166, 91, 133, 27]. An emergent technique, known as contrastive learning, relies on contrastive losses [70] to learn view invariant representations, where the different views of the data can be generated by data augmentation [206, 133, 76, 32, 84], chunking the input over time or space [147, 72] or using co-occurring modalities [8, 92, 138, 216, 190]. In this work, we also rely on contrastive losses, but utilize contrastive learning to perform audio-visual spatial alignment.

Similarly to the proposed AVSA task, spatial context has previously been used in visual representation learning. For example, [142, 44, 99] try to predict the relative locations of image

or video patches, and [72] uses contrastive learning to learn representations that are predictive of their spatio-temporal location. However, as shown in [138], using visual content as both the input and target for representation learning can yield sub-optimal representations, as low-level statistics can be explored to perform the task without learning semantic features. Our approach addresses this issue by leveraging the spatial context provided by a co-occurring modality (audio) that also contains strong spatial cues.

Audio-visual learning. The natural co-occurrence of vision and sound has been successfully used in various contexts such as visually guided source separation and localization [58, 60, 221, 220, 57], and audio spatialization [137, 59]. Audio-visual correspondences [8, 7] have also been used for learning representations for objects and scenes in static images [8, 7, 150], action recognition in video [148, 105, 138, 4], to perform temporal synchronization [37, 75, 148, 105] and audio classification [13]. As discussed in Figure 4.1, prior work is often implemented either by predicting audio-visual correspondences at the video level [8, 7, 138] or performing temporal synchronization using out-of-sync clips as hard negatives [148, 105]. However, [175] shows that basic audio-visual correspondences are ill-equipped to identify and localize sound sources in the video. We argue that this is because audio-visual correspondences are imposed by matching audio to the entire video clip. Thus, there is little incentive to learn discriminative features for objects that often co-occur. To address this issue, we explore the rich spatial cues present in both the 360°video and spatial audio. By learning to spatially align visual and audio contents, the network is encouraged to reason about the scene composition (i.e. the locations of the various sources of sound), thus yielding better representations for downstream tasks.

4.3 Audio-visual spatial alignment

We learn audio-visual representations by leveraging spatial cues in 360°media. 360°video and spatial audio encode visual and audio signals arriving from all directions (θ, ϕ) around

the recording location, where θ denotes the longitude (or horizontal) angle, ϕ the latitude (or elevation) angle. We adopt the equi-rectangular projection as the 360°video format and first-order ambisonics [62] for the spatial audio. Both formats can be easily rotated and/or decoded into viewpoint specific clips.

4.3.1 Pretext task

Regressive AVSA. A straight-forward implementation of audio-visual spatial alignment is to generate random rotations R of either the video or audio so as to create an artificial misalignment between them. A model can then be trained to predict the applied transformation by solving

$$\min_{f_v, f_a, g} \mathbb{E}_{v,a,R} \left\{ d \left[g(f_v(v), f_a(R(a))), R \right] \right\}, \tag{4.1}$$

where f_v and f_a are the video and audio encoders, g a rotation regression head, and d the distance between the predicted and ground-truth rotations R. However, this implementation has several disadvantages. Due to the continuous nature of the target variable R, the loss of (4.1) is difficult to optimize. Also, the task is defined on the full 360°video v, which limits the use of data augmentation techniques such as aggressive cropping that are critical for self-supervised learning.

Contrastive AVSA. Inspired by recent advances in contrastive learning [70, 147, 206, 190, 138], we propose to solve the audio-visual spatial alignment task in a contrastive fashion. As shown in Figure 4.1, given a 360°audio-video sample (v_i, a_i), K video and audio clips $\{(v_i^k, a_i^k)\}_{k=1}^K$ are extracted from K randomly sampled viewing directions $\{(\theta_k, \phi_k)\}_{k=1}^K$. Video clips v_i^k are obtained by extracting normal field-of-view (NFOV) crops using a Gnomonic projection [205] centered around (θ_k, ϕ_k), and audio clips a_i^k by realigning the global frame of reference of the ambisonics signal such that the frontal direction points towards (θ_k, ϕ_k) [108]. Audio-visual spatial alignment is then encouraged by tasking a network to predict the correct correspondence between the K video $\{v_i^k\}_{k=1}^K$ and the K audio $\{a_i^k\}_{k=1}^K$ signals.

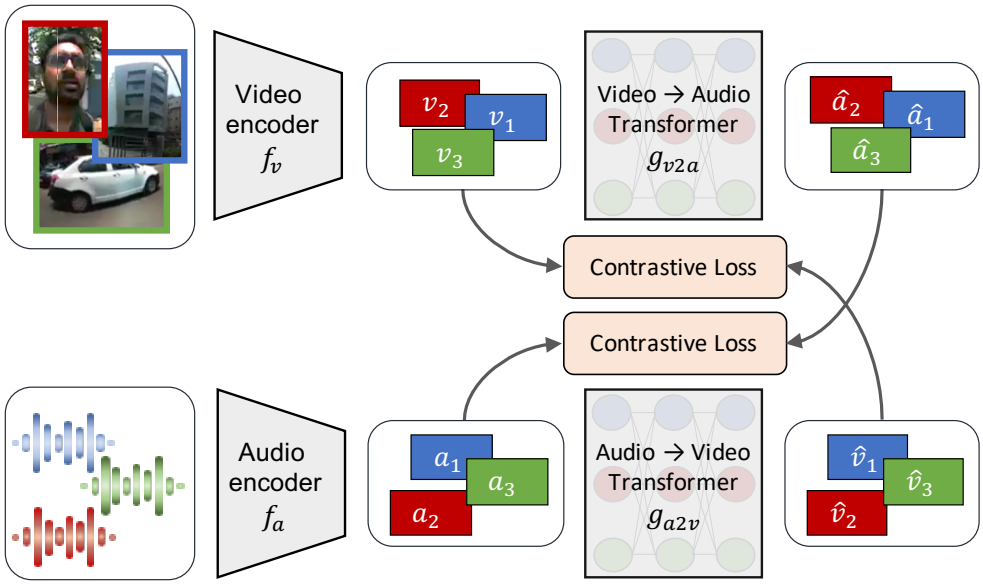


Figure 4.2: Architecture overview for contrastive audio-visual spatial alignment.

4.3.2 Architecture

Figure 4.2 summarizes the architecture used to solve the spatial alignment task. First, video and audio encoders, f_v and f_a, extract feature representations from each clip independently,

$$\mathbf{v}_i^k = f_v(v_i^k) \quad \text{and} \quad \mathbf{a}_i^k = f_a(a_i^k). \tag{4.2}$$

These representations are then converted between the two modalities using audio-to-video g_{a2v} and video-to-audio g_{v2a} feature translation networks

$$\bar{\mathbf{v}}_i^1, \ldots, \bar{\mathbf{v}}_i^K = g_{a2v}(\mathbf{a}_i^1, \ldots, \mathbf{a}_i^K) \quad \text{and} \quad \bar{\mathbf{a}}_i^1, \ldots, \bar{\mathbf{a}}_i^K = g_{v2a}(\mathbf{v}_i^1, \ldots, \mathbf{v}_i^K). \tag{4.3}$$

One important distinction between audio and video is the spatial localization of the signals. Unlike video, any sound source can be heard regardless of the listening angle. In other words, while an audio clip a_i^k sampled at position (θ_k, ϕ_k) contains audio from all sound sources present in a scene, only those physically located around (θ_k, ϕ_k) can be seen on the video clip v_i^k. This implies that, to enable accurate feature translation, networks g_{v2a} and g_{a2v} should combine

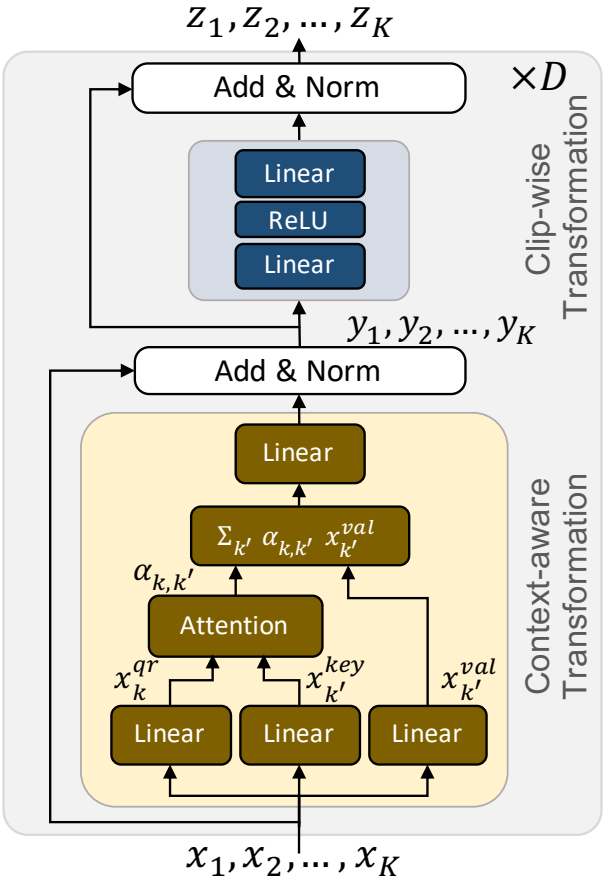


Figure 4.3: Transformer architecture for context-aware video-to-audio and audio-to-video feature translation.

features from all sampled locations. This is accomplished by using a translation network similar to the transformer of [193]. As shown in Fig. 4.3, given a set of K features $\{\mathbf{x}_k\}_{k=1}^K$, a transformer of depth D alternates D times between two modules. The first module combines the K features $\mathbf{x}_k$ using attention

$$\alpha_{k,1},\ldots,\alpha_{k,K} = \mathit{Softmax}\left(\frac{\left\langle W_{key}^T\mathbf{x}_k, W_{qr}^T\mathbf{x}_1\right\rangle}{\sqrt{d}},\ldots,\frac{\left\langle W_{key}^T\mathbf{x}_k, W_{qr}^T\mathbf{x}_K\right\rangle}{\sqrt{d}}\right) \tag{4.4}$$

$$\mathbf{y}_k = \mathit{Norm}\left(\mathbf{x}_k + W_0^T\textstyle\sum_{k'}\alpha_{k,k'}W_{val}^T\mathbf{x}_{k'}\right). \tag{4.5}$$

The second module computes a simple clip-wise feed-forward transformation

$$\mathbf{z}_k = \mathit{Norm}\left(\mathbf{y}_k + W_2^T\max(W_1^T\mathbf{y}_k, 0)\right). \tag{4.6}$$

In (4.4)-(4.6), $W_{key}, W_{qr}, W_{val}, W_0, W_1$ and W_2 are learnable weights and *Norm* is layer normalization [14]. We omit the biases of linear transformations and layer indices for simplicity of notation. Compared to the original transformer [193], the proposed translation network differs in two aspects. First, motivated by early empirical results which showed no improvements on downstream tasks when utilizing multi-head attention, we simplified the transformer architecture to rely on a single attention head. Second, we removed positional encodings which are used to indicate the position of each token $\mathbf{x}_k$. While these encodings could be used to encode the viewing direction (θ_k, ϕ_k) of each clip, doing so would allow the model to solve the spatial alignment task without learning semantic representations.

4.3.3 Learning strategy

AVSA is a difficult task to optimize since it requires discriminating between various crops from the same video. To enhance learning, we employed a curriculum learning strategy [19]. In the first phase, the network is trained to identify audio-visual correspondences (AVC) [8, 138] at

the video level. This is accomplished by extracting a single crop (v_i, a_i) for each video i from a randomly drawn viewing angle. The visual and audio encoders, f_v and f_a, are then trained to minimize

$$L_{AVC} = \sum_i L_{InfoNCE}\left(\mathbf{v}_i, \mathbf{a}_i, \{\mathbf{a}_j\}_{j=1}^{N}\right) + L_{InfoNCE}\left(\mathbf{a}_i, \mathbf{v}_i, \{\mathbf{v}_j\}_{j=1}^{N}\right) \tag{4.7}$$

where $\mathbf{v}_i = f_v(v_i)$ and $\mathbf{a}_i = f_a(a_i)$ are the video and audio representations. $L_{InfoNCE}$ is the InfoNCE loss [147] defined as

$$L_{InfoNCE}(\mathbf{x}, \mathbf{x}_t, \mathcal{P}_\mathbf{x}) = -\log \frac{\exp(h(\mathbf{x}_t, \mathbf{x})/\tau)}{\sum_{\mathbf{x}_p \in \mathcal{P}_\mathbf{x}} \exp(h(\mathbf{x}_p, \mathbf{x})/\tau)} \tag{4.8}$$

where $h(\mathbf{x}, \mathbf{x}_t)$ is a prediction head that computes the cosine similarity between $\mathbf{x}$ and $\mathbf{x}_t$ after linear projection into a low-dimensional space, and τ is a temperature hyper-parameter. In the case of AVC, the target representation $\mathbf{x}_t$ for the InfoNCE loss is the feature from the crop of same video but opposing modality, and the proposal distribution $\mathcal{P}_\mathbf{x}$ is composed by the target feature representations of all videos in the batch.

In the second phase, the network is trained on the more challenging task of matching audio and video at the crop level, i.e. matching representations in the presence of multiple crops per video. This is accomplished by augmenting the proposal set $\mathcal{P}_\mathbf{x}$ to include representations from multiple randomly sampled viewing angles $\{(v_i^k, a_i^k)\}_{k=1}^{K}$ from the same video. In this phase, we also introduce the feature translation networks g_{v2a} and g_{a2v} and require the translated features ($\bar{\mathbf{v}}_i^k$ and $\bar{\mathbf{a}}_i^k$) to match the encoder outputs ($\mathbf{v}_i^k$ and $\mathbf{a}_i^k$) obtained for the corresponding viewing angle k. Encoders f_v and f_a and feature translation networks g_{v2a} and g_{a2v} are jointly trained to minimize

$$L_{AVSA} = \sum_i \sum_k L_{InfoNCE}\left(\bar{\mathbf{v}}_i^k, \mathbf{v}_i^k, \left\{\mathbf{v}_j^l\right\}_{j,l=1}^{N,K}\right) + L_{InfoNCE}\left(\bar{\mathbf{a}}_i^k, \mathbf{a}_i^k, \left\{\mathbf{a}_j^l\right\}_{j,l=1}^{N,K}\right). \tag{4.9}$$

4.4 YouTube-360 dataset

We collected a dataset of 360°video with spatial audio from YouTube, containing clips from a diverse set of topics such as musical performances, vlogs, sports, and others. This diversity is critical to learn good representations. Similarly to prior work [137], search results were cleaned by removing videos that 1) did not contain valid ambisonics, 2) only contain still images, or 3) contain a significant amount of post-production sounds such as voice-overs and background music. The resulting dataset, denoted YouTube-360 (YT-360), contains a total of 5 506 videos, which was split into 4 506 videos for training and 1 000 for testing. Since we use audio as target for representation learning, periods of silence were ignored. This was accomplished by extracting short non-overlapping clips whose volume level is above a certain threshold. In total, 88 733 clips of roughly 10s each were collected (246 hours of video content). As shown in Table 4.1, the YT-360 dataset contains five times more videos than the largest 360°video dataset previously collected.

To assess the ability of AVSA pre-training to localize objects in a scene, we conduct evaluations on semantic segmentation as a downstream task. Due to the large size of our dataset, collecting ground-truth annotations is impractical. Instead, we used the state-of-the-art ResNet101 Panoptic FPN model [103] trained on the MS-COCO dataset [124] to segment the 32 most frequent objects and background classes on YT-360. A description of the segmentation procedure, including the selected classes, is provided in appendix. These segmentation maps are used to evaluate AVSA representations by knowledge distillation, as discussed in Section 4.5.3. Examples from the YT-360 dataset are shown in Figure 4.4 together with the predicted segmentation maps and a heat-map representing the directions of higher audio volume.

Table 4.1: Comparison of 360°video datasets.

	Spatial Audio	Unique Videos	Hours
Duanmu *et al*. [48]		12	0.3
Li *et al*. [120]		73	3.8
Pano2VID [185]		86	7.3
SptAudioGen [137]	✓	1146	113
YT-360	✓	5506	246

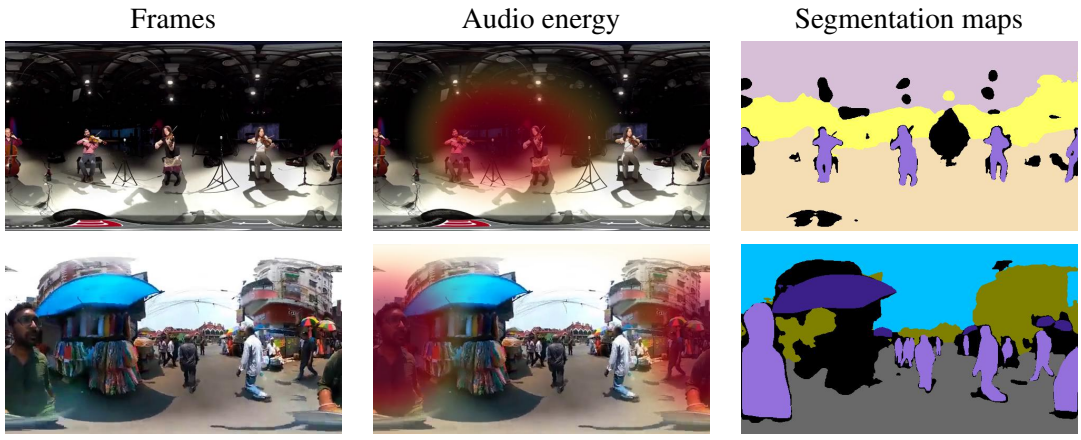


Figure 4.4: Examples from Youtube-360 dataset.

4.5 Experiments

We evaluate the representations learned by AVSA pre-training on several downstream tasks. We explain the experimental setting below, and refer the reader to appendix for additional details.

4.5.1 Experimental setting

Video pre-processing

We sampled $K = 4$ crops per video at different viewing angles. Since up and down viewing directions are often less informative, we restrict the center of each crop to latitudes $\phi \in \{-60°, 60°\}$. We also ensure that viewing angles are sampled at least 36°apart. Normal field-of-view (NFOV) crops are extracted using a Gnomonic projection with random angular coverage between 25°and 90°wide for data augmentation. If naive equi-rectangular crops were taken, the distortion patterns of these crops at latitudes outside the horizon line could potentially reveal the vertical position of the crop, allowing the network to "cheat" the AVSA task. Following NFOV projection, video clips are resized into 112×112 resolution. Random horizontal flipping, color jittering and Z normalization are applied. Each video clip is $0.5s$ long and is extracted at 16fps.

Audio pre-processing

First-order ambisonics (FOA) are used for spatial audio. Audio clips for the different viewing angles are generated by simply rotating the ambisonics [108]. One second of audio is extracted at 24kHz, and four channels (FOA) of normalized log mel-spectrograms are used as the input to the audio encoder. Spectrograms are computed using an STFT with a window of size 21ms, and hop size of 10ms. The extracted frequency components are aggregated in a mel-scale with 128 levels.

Architecture and optimization

The video encoder f_v is the 18-layer R2+1D model [191], and the audio encoder f_a is a 9-layer 2D convolutional neural network operating on the time-frequency domain. The translation networks, g_{v2a} and g_{a2v}, are instantiated with depth $D = 2$. Training is conducted using the Adam optimizer [101] with a batch size of 28 distributed over 2 GPUs, learning rate of $1e-4$, weight decay of $1e-5$ and default momentum parameters $(\beta_1, \beta_2) = (0.9, 0.999)$. Both curriculum learning phases are trained for 50 epochs. To control for the number of iterations, models trained only on the first or second phases are trained for 100 epochs.

Baseline pre-training methods

We compare AVSA to Audio-Visual Correspondence (AVC) [8, 7, 138] and Audio-Visual Temporal Synchronization (AVTS) [105, 148]. Since prior works perform pretext training on flat video datasets (i.e. without spatial audio), a direct comparison is impossible. Instead, we train AVC and AVTS models on the YouTube-360 dataset. For fair comparisons, we use the architecture and optimization settings described above. AVC is trained to optimize the loss of (4.7), which only uses negatives from different videos. Note that (4.7) is similar to the loss used in [8, 7] but considers multiple negatives simultaneously. This has actually been shown to improve generalization in [138]. To implement AVTS, we augment the proposal set $\mathcal{P}_{\mathbf{x}}$ of the InfoNCE loss of (4.8) with clips sampled from different moments in time. Following [105, 148], we ensure that negative pairs of audio and video clips are sufficiently separated in time. We also use a curriculum learning strategy composed by an AVC pre-training phase as in [105]. In the base AVC and AVTS implementations, we directly match the audio and visual features computed by the encoders f_v and f_a directly, as done in the original papers [8, 138, 105, 148]. However, to control for the number of seen crops, we also conduct AVC and AVTS pre-training using multiple crops of the same video and the feature translation networks g_{a2v} and g_{v2a}. Since AVC requires predictions at the video level (not for each individual clip), clip representations are combined by

Table 4.2: Accuracy of binary AVC and AVSA predictions using one or four viewpoints on the YT-360 test set.

Evaluation Task		AVC-Bin		AVSA-Bin	
# Viewpoints		1	4	1	4
AVC	no transf.	79.82	82.68	59.48	59.25
	transf.	–	83.87	–	61.20
AVTS	no transf.	80.08	82.77	59.78	60.37
	transf.	–	83.77	–	60.73
AVSA	no transf.	86.19	91.67	64.97	68.87
	transf.	–	89.83	–	69.97

max-pooling.

4.5.2 Audio-visual spatial alignment

We start by considering the performance on the AVC and AVSA tasks themselves. AVC performance is measured by randomly generating 50% of audio-video pairs from the same sample (positives), and 50% of pairs from different samples (negatives). Similarly, we designed a binary AVSA evaluation task in which positive audio-video pairs are spatially aligned, while negative pairs were artificially misaligned by randomly rotating the ambisonic audio of a positive pair. Rotations are constrained around the yaw axis (horizontal) to ensure the audio from positive and negative pairs have the same distribution, and thus making the AVSA task more challenging. Since models trained by AVC are not tuned for AVSA evaluation and vice-versa, the pretext models cannot be directly evaluated on the above binary tasks. Instead, we trained a new binary classification head on top of video and audio features, while keeping pretext representations frozen. Also, since NFOV video crops only cover a small portion of the 360°frame, we also consider predictions obtained by averaging over four viewpoints.

Table 4.2 shows that the proposed AVSA pretext training mechanism significantly outperforms AVC and AVTS on both evaluation tasks. Remarkably, even though AVC pretext training

optimizes for the AVC task directly, representations learned with AVSA outperformed those learned with AVC by more than 6% on the AVC task itself (AVC-Bin). Furthermore, both AVC and AVTS models learned by audio-video correspondence or temporal synchronization do not transfer well to the spatial alignment task (AVSA-Bin). In result, AVSA outperforms AVC and AVTS by more than 5% on spatial alignment. By learning representations that are discriminative of different viewpoints, AVSA also learns a more diverse set of features. This is especially helpful when combining information from multiple viewpoints, as demonstrated by the differences in the gains obtained by 4 crop predictions. For example, AVC and AVTS only benefit by a 2-3% gain from 4 crop predictions on the AVC-Bin task, while AVSA performance improves by 5.5%. On the AVSA-Bin task, 4 crop predictions do not improve AVC or AVTS significantly, while AVSA performance still improves by 4%. We also observe improvements by using the transformer architecture in 5 out of 6 configurations (3 pretext tasks $\times$ 2 evaluations), showing its effectiveness at combining information from different viewpoints.

4.5.3 Semantic segmentation by knowledge distillation

AVSA representations are also evaluated on semantic segmentation. As shown in Figure 4.5, the video encoder f_v was used to extract features at multiple scales, which were combined using a feature pyramid network (FPN) [123] for semantic segmentation. To measure the value added by audio inputs, we concatenate the features from the audio encoder f_a at the start of the top-down pathway of the FPN head. Similarly, to measure the benefits of combining features from multiple viewpoints, we concatenate the context-aware representations computed by the feature translation modules g_{v2a} and g_{a2v}. Since the goal is to evaluate the pretext representations, networks trained on the pretext task were kept frozen. The FPN head was trained by knowledge distillation, i.e. using the predictions of a state-of-the-art model as targets. We also compare to a fully supervised video encoder pre-trained on Kinetics for the task of action recognition. Similar to the self-supervised models, the fully supervised model was kept frozen. To provide an upper

bound on the expected performance, we trained the whole system end-to-end (encoders, feature translation modules and the FPN head). A complete description of the FPN segmentation head and training procedure is given in appendix.

Table 4.3 shows the pixel accuracy and mean IoU scores obtained using video features alone, or their combination with audio and context features. Examples of segmentation maps obtained with the AVSA model with context features are also shown in Figure 4.6. The results support several observations. AVSA learns significantly better visual features for semantic segmentation than AVC. This is likely due to the fine-grained nature of the AVSA task which requires discrimination of multiple crops within the same video frame. As a result, AVSA improves the most upon AVC on background classes such as rocks (34.7% accuracy vs. 27.7%), window (46.0% vs. 41.2%), pavement (36.8% vs. 33.3%), sand (42.1% vs. 38.8%), sea (50.1% vs. 46.8%) and road (47.1% vs. 45.1%).

AVSA also learns slightly better visual features than AVTS. While the gains over AVTS using visual features alone are smaller, AVTS cannot leverage the larger spatial context of 360°video data. When context features from four viewpoints are combined, using the translation networks g_{v2a} and g_{a2v}, further improvements are obtained. With context features, AVSA yields a 3% mIoU improvement over AVC and 1% over AVTS.

Finally, we evaluated two ablations of AVSA. To verify the benefits of curriculum learning, we optimized the AVSA loss of (4.9) directly. Without curriculum, AVSA achieved 1.5% worse mIoU (see Table 4.3 AVSA no curr.). We next verified the benefits of modeling spatial context by disabling the transformer ability to combine information from all viewpoints. This was accomplished by replacing the attention module of Figure 4.3 with a similarly sized multi-layer perceptron, which forced the translation networks to process each viewpoint independently. While this only produced slightly worse visual representations, the ability to leverage spatial context was significantly affected. Without the transformer architecture, AVSA yielded 1.5% worse mIoU scores when using context features (see Table 4.3 AVSA mlp).

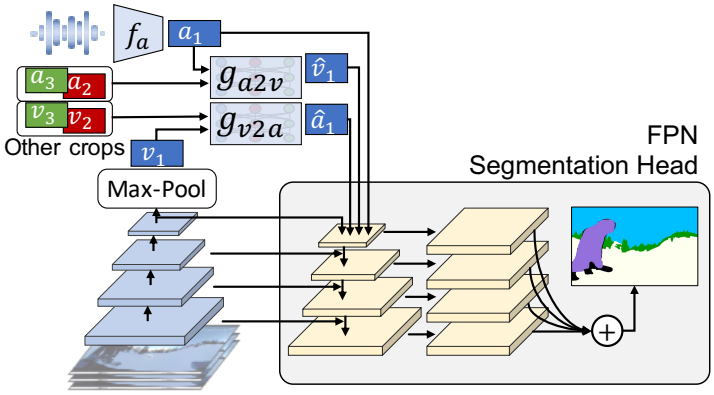


Figure 4.5: Architecture used for semantic segmentation. Pre-trained networks are kept frozen. A lightweight FPN segmentation head [123] is trained by knowledge distillation.

Table 4.3: Pixel accuracy and mean IoU of semantic segmentation predictions on YT-360 test set. We evaluate the performance of an FPN head that uses 1) visual features alone, 2) visual and audio features, and 3) visual, audio and context features obtained from four viewpoints.

	Video only		+Audio		+Audio+Context	
	Pix Acc	mIoU	Pix Acc	mIoU	Pix Acc	mIoU
AVC	71.16	32.85	71.07	32.69	–	–
AVTS	73.24	34.88	72.97	**34.88**	–	–
AVSA	**73.44**	**35.11**	**73.11**	34.63	**73.85**	**35.83**
AVSA (no curr.)	71.95	33.66	71.49	33.23	72.06	34.30
AVSA (mlp)	73.10	35.02	73.21	34.83	72.68	34.35
Kinetics (sup)	75.47	36.91	–	–	–	–
End-to-end (upper bound)	77.37	41.05	77.93	42.00	79.65	43.21

4.5.4 Action recognition

Action recognition is a common downstream task used to benchmark audio-visual self-supervised approaches. Following standard practices, we finetuned the pretext models either on the UCF [182] or the HMDB [109] datasets, and measure the top-1 accuracies obtained for a single clip or by averaging predictions over 25 clips per video. For comparison, we also provide the performance of our model trained on UCF and HMDB from a random initialization (Scratch), or finetuned from a fully supervised model trained on Kinetics [196] (Kinetics Sup.). Full details of the training procedure are given in appendix. The results shown in Table 4.4 show once more

Figure 4.6: Predictions from an AVSA pre-trained model with an FPN segmentation head on the YT-360 test set.

the benefits of AVSA pretext training. AVSA dense predictions outperform AVC by 4% on UCF and 3% on HMDB, and outperform AVTS by 3.5% on UCF and 2% on HMDB.

4.6 Discussion, future work and limitations

We presented a novel self-supervised learning mechanism that leverages the spatial cues in audio and visual signals naturally occurring in the real world. Specifically, we collected a 360°video dataset with spatial audio, and trained a model to spatially align video and audio clips extracted from different viewing angles. The proposed AVSA task was shown to yield better representations than prior work on audio-visual self-supervision for downstream tasks like audio-visual correspondence, video semantic segmentation, and action recognition. We also proposed to model 360°video data as a collection of NFOV clips collected from multiple viewpoints, using a transformer architecture to combine view specific information. Being able to summarize

Table 4.4: Action recognition performance on UCF and HMDB datasets. The top-1 accuracy of single clip and dense predictions are reported.

	UCF		HMDB	
	Clip@1	Video@1	Clip@1	Video@1
Scratch	54.85	59.95	27.40	31.10
Kinetics Sup.	78.50	83.43	46.45	51.90
AVC	64.63	69.68	31.33	34.58
AVTS	65.65	70.34	32.29	35.89
AVSA	**68.52**	**73.80**	**32.96**	**37.66**

information from the whole 360°video frame was proven advantageous for downstream tasks defined on 360°video data. For additional parametric and ablation studies, we refer the reader to supplementary material, where we ablate several components of the proposed approach, including the type of audio input provided to the network, the number and type of viewpoints in the AVSA objective, and the influence of curriculum learning and the transformer module.

Since AVSA requires discrimination of different viewpoints within a 360°scene, the learned models are encouraged to localize sound sources in the video and audio signals in order to match them. In addition to better performance on downstream tasks, this pre-training objective also translates into improved localization ability, based on a qualitative analysis. Fig. 4.7 shows several examples of GradCAM [174] visualizations for AVC and AVSA models (GradCAM is applied to each model's audio-visual matching score). As can be seen, AVSA models tend to localize sound sources better. Furthermore, while the proposed method relies on randomly extracted video and audio clips, more sophisticated sampling techniques are an interesting direction of future work. For example, sampling can be guided towards objects using objectness scores, towards moving objects using optical flow, or towards sound sources by oversampling viewpoints with high audio energy. Such sampling techniques would better mimic a human learner, by actively choosing which parts of the environment to dedicate more attention. They would also under-sample less informative viewpoints (e.g. crops dominated by background),

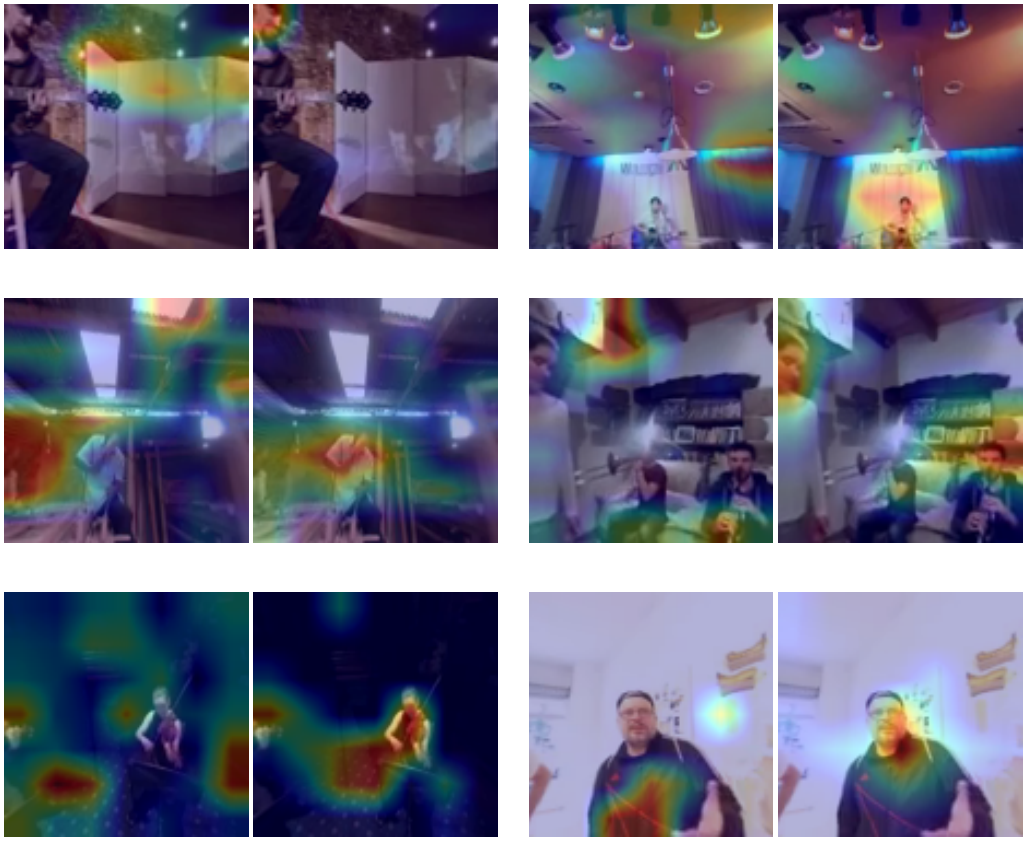


Figure 4.7: Sound localization maps (GradCAM of audio-visual matching scores) obtained from models trained by AVC (first image of each pair) and AVSA (second of each pair).

which are hard to match to the corresponding sound, and thus may harm the quality of learned representations.

Finally, we note that AVSA requires 360°data with spatial audio, which is still less prevalent than regular video. Previous methods, such as AVC and AVTS [8, 148, 105, 138], are often trained on datasets several orders of magnitude larger than YT-360, and can achieve better performance on downstream tasks such as action recognition. However, this work shows that, for the same amount of training data, AVSA improves the quality of the learned representations significantly. Due to the growing popularity of AR/VR, 360°content creation is likely to grow substantially. As the number of available 360°videos with spatial audio increases, the quality of representations learned by AVSA should improve as well.

4.7 Appendix

4.7.1 Implementation details

In this section, we describe in detail the implementation of the proposed AVSA pre-training as well as the semantic segmentation and action recognition downstream tasks.

Audio-visual spatial alignment

The architecture of the video and audio encoder networks, f_v and f_a, are shown in Table 4.5 and Table 4.6. The feature translation networks are described in Section 3.2 and depicted in Figure 3 of the main text. These are transformer networks of base dimension 512 and expansion ration 4. In other words, the output dimensionality of the linear transformations of parameters $W_{key}, W_{qr}, W_{val}, W_0$ and W_2 are 512, and that of W_1 is 2048. Models are pre-trained to optimize loss (7) for AVC task or (9) for AVTS and AVSA tasks. AVTS models are trained using negatives obtained from the same viewpoint but different moments in time. AVSA models are obtained using negatives obtained from the same moment in time but different viewpoints. All models were trained using the Adam optimized. Pre-training hyper-parameters are summarized in Table 4.7.

Semantic segmentation

For semantic segmentation, we used a lightweight FPN segmentation head. As originally proposed, lateral connections are implemented with a 1×1 convolution that maps all feature maps into a 128 dimensional space followed by a 3×3 convolution for increased smoothing. Since the FPN head is used to perform semantic segmentation of a single frame given a video clip with multiple frames, we perform global temporal pooling of the feature maps before feeding them to the lateral connections. Semantic segmentation predictions are then computed based on the features at all levels. First, features from low-resolution layers are upsampled through a sequence of 3×3 convolutions with dilation of 2 into 56×56 resolution and added together to perform

Table 4.5: Architecture details of R(2+1)D video network based of R(2+1)D convolutions, and the audio on 2D convolutions with ReLU activations and batch normalization at each layer.

Video Network

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

X_s spatial activation size, X_t temporal activation size, C number of channels
K_s spatial kernel size, K_t temporal kernel size, S_s spatial stride, S_t temporal stride.

pixel-wise classification. All parameters of the FPN head are trained to minimize the softmax cross-entropy loss average across all pixels. Since we are using the output of a state-of-the-art model as ground truth, we avoid using low-confidence ground-truth labels. Thus, all pixels for which the state-of-the-art model was less than 75% confident were kept unlabeled. These low confidence regions were also ignored while computing evaluation metrics. The model was trained using the Adam optimizer with batch size 20, learning rate $1e-4$ and weight decay $5e-4$ for 10 epochs. The learning rate was decayed at epochs 5 and 8. Video clips were extracted from random viewpoints within the 360°video, with random angular coverage between 45°and 90°for data augmentation. Color jittering and horizontal flipping was also applied.

Table 4.6: Architecture details of Conv2D audio network based on 2D convolutions with ReLU activations and batch normalization at each layer.

Audio Network

Layer	X_f	X_t	C	K_f	K_t	S_f	S_t
`audio`	129	100	N	-	-	-	-
`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

X_t temporal activation size, X_f frequency activation size, C number of channels, K_t temporal kernel size, K_f frequency kernel size, S_t temporal stride, S_f frequency stride.

Table 4.7: Pre-training optimization hyper-parameters. AVSA models are initialized by the AVC model obtained at epoch 100.

Method	bs	nv	lr	wd	cj	hf	hfov_{min}	hfov_{max}	in	sn	tn	τ
AVC	112	1	1e-4	1e-5	✓	0.5	25	90	✓			0.07
AVTS	28	4	1e-4	1e-5	✓	0.5	25	90	✓		✓	0.07
AVSA	28	4	1e-4	1e-5	✓	0.5	25	90	✓	✓		0.07

bs–batch size; nv–number of viewpoints; lr–learning rate; wd–weight decay; cj–color jittering; hf–horizontal flip probability; hfov_{min}/hfov_{max}–minimum/maximum horizontal field-of-view in degrees; in/sn/tn–instance/spatial/temporal negatives; τ–InfoNCE temperature.

Action recognition

The video encoder network was evaluated on the task of action recognition using UCF and HMDB datasets. We augmented the video encoder with a linear classification layer after the global max-pooling operation, and finetune the whole network. We used Adam optimization for 100 epochs, with batch size 28, learning rate 10^{-4} decayed at epochs 40, 60 and 80. Performance is reported on first train/test split originally defined for the UCF and HMDB datasets.

4.7.2 Ablations and parametric studies

We assess different components of the proposed pre-trained mechanism through several ablation and parameteric studies shown in Table 4.8. All models are evaluated on AVC-Bin, AVSA-Bin, semantic segmentation, and action recognition tasks as introduced in §5.2–5.4 of main text (4 crops per video are used for AVC-Bin and AVSA-Bin). We report accuracies for the AVC-Bin, AVSA-Bin tasks using 4 viewpoints, mean IoU for the semantic segmentation task and clip level accuracy for action recognition on UCF.

Influence of spatial audio

To demonstrate the value of spatial audio, we train the AVSA pretext task using different inputs to the audio network: single channel mono audio, two channel stereo audio, and four channel ambisonic audio. The three versions of the audio input can be easily computed from the full ambisonics signal. The mono version of audio is generated by taking the projection of the ambisonic signal into the spherical harmonics at each viewing angle. To generate stereo, we use a standard ambisonic binauralizer that models a human listener looking at each viewing angle. To generate ambisonics, we simply rotate the original signal to align with each viewing angle. Assuming a typical ambisonics format with 4 channels, this is done by applying a 3D rotation matrix to its first-order spherical harmonic components (X, Y and Z channels), while keeping the zeroth-order component (W channel) fixed.

Table 4.8a shows substantial improvement ($\sim 7\%$) in AVC and AVSA tasks by using full ambisonics for each crop over mono audio, suggesting that the latter may not be sufficient to encode spatial information of sound sources. Using stereo audio which retains partial spatial information also improves over mono input, but with a smaller margin. For semantic tasks (segmentation and action recognition on UCF), learning with ambisonics also proved to be more effective.

Influence of number of viewpoints

As more viewpoints are extracted from each sample, the difficulty of the AVSA task increases since more options are provided for matching. To investigate whether the increased difficulty correlates with the quality of the learned representation, we vary the number of viewpoints during AVSA pre-training.

Table 4.8b shows the AVC and AVSA performance increases monotonically as more viewpoints are used. However, these gains not always translates into better performance on semantic tasks. Semantic segmentation achieved the best performance by training to discriminate 2 or 4 viewpoints, while action recognition peaked at 4 viewpoints.

Influence of type of negative crops

The AVSA pretext tasks uses a combination of easy and hard (spatial) negatives: Easy negatives are clips from different video *instances*. Hard (spatial) negatives are sampled from different viewpoints, but the same moment in time. We also trained a network with hard spatio-temporal negatives, which can be sampled from any viewpoint and moment in time within the video. Table 4.8c shows the performance of models trained with different kinds of negatives crops. As can be seen, the combination of instance-based and spatial negatives (as used by the AVSA approach) yields better performance than using instance-based negatives alone (as used by AVC approaches). This shows the use of spatial negatives is complementary to AVC. However, the results are mixed when combining AVSA with temporal negatives (as used by AVTS approaches), producing slightly better semantic segmentations, but worse UCF performance.

Influence of curriculum learning

Prior work indicates that curriculum learning can benefit training by starting from easier sub-tasks and progressively increase the difficulty of the task being learned. To test this hypothesis in the AVSA context, we evaluate our network trained with and without the curriculum learning

Table 4.8: Ablation studies.

	AVC@4	AVSA@4	Segm	UCF
Mono	82.39	62.95	34.21	64.90
Stereo	84.47	**71.11**	34.54	64.68
Ambisonics	**89.83**	69.97	**35.83**	**68.52**

(a) Spatial audio format.

	AVC@4	AVSA@4	Segm	UCF
1 Viewpoint	84.60	61.77	35.37	64.71
2 Viewpoints	87.70	63.71	**36.63**	66.64
4 Viewpoints	89.83	69.97	35.83	**68.52**
8 Viewpoints	**91.65**	**74.64**	34.84	66.44

(b) Number of viewpoints.

	AVC@4	AVSA@4	Segm	UCF
Instance	83.87	61.20	34.05	64.09
+ Spatial	**89.83**	69.97	35.83	**68.52**
+ Spatial + Temporal	89.65	**72.81**	**36.11**	65.77

(c) Negative crop type.

	AVC@4	AVSA@4	Segm	UCF
Easy Only	83.87	61.20	34.05	64.09
Hard Only	**93.22**	**77.71**	20.97	59.15
No Curriculum	91.93	71.77	35.29	65.49
Curriculum	89.83	69.97	**35.83**	**68.52**

(d) Curriculum learning.

	AVC@4	AVSA@4	Segm	UCF
Direct Prediction	**91.67**	68.87	34.50	65.59
Transformer (Depth=1)	90.64	**72.95**	35.77	66.97
Transformer (Depth=2)	89.83	69.97	35.83	**68.52**
Transformer (Depth=4)	89.86	70.09	**35.97**	66.88

(e) Modeling spatial context.

strategy (first optimizing for easy negatives, i.e. AVC, then optimizing for easy and hard negatives combined). We also compare to baselines where the model is only optimized for easy or hard

negatives.

Table 4.8d shows that training on hard negatives directly leads to the best AVC and AVSA performance. However, the learned representations significantly overfit to the pretext task, and do not transfer well to semantic tasks, as seen by the low performance on semantic segmentation and action recognition. Using a combination of easy and hard negatives proved to be beneficial for these two downstream tasks, with the curriculum learning strategy achieving the best results.

Influence of modeling spatial context

We propose to use a transformer network to leverage the rich spatial context of spatial audio and 360° video while translating features across the two modalities. To assess the importance of modeling spatial context, we evaluate models trained with and without the transformer networks. We further vary the depth of transformer module in search of a good trade-off between model complexity and quality of learned representations.

Table 4.8e shows that modeling spatial context is not required to predict whether audio and video clips originate from the same sample (achieving lower AVC accuracy). However, the ability to perform spatial alignment is significantly impacted without the transformer network, showing that it is harder to perform spatial alignment without combining information from multiple viewpoints. The lack of spatial context also impacted both semantic segmentation and action recognition on UCF. For semantic tasks, a transformer of depth $D = 2$ provided a good trade-off between model complexity and model performance.

4.8 Acknowledgements

Chapter 4 is, in full, based on the material as it appears in the publication of "Learning Representations from Audio-Visual Spatial Alignment", Pedro Morgado, Yi Li, Nuno Vasconcelos, as appears in the Advances of Neural Information Processing Systems (NeurIPS), 2020.

Chapter 5

Spatial Audio Generation

5.1 Introduction

360° video provides viewers an immersive viewing experience where they are free to look in any direction, either by turning their heads with a Head-Mounted Display (HMD), or by mouse-control while watching the video in a browser. Capturing 360° video involves filming the scene with multiple cameras and stitching the result together. While early systems relied on expensive rigs with carefully mounted cameras, recent consumer-level devices combine multiple lenses in a small fixed-body frame that enables automatic stitching, allowing 360° video to be recorded with a single push of a button.

As humans rely on audio localization cues for full scene awareness, *spatial audio* is a crucial component of 360° video. Spatial audio enables viewers to experience sound in all directions, while adjusting the audio in real time to match the viewing position. This gives users a more immersive experience, as well as providing cues about which part of the scene might have interesting content to look at. However, unlike 360° video, producing spatial audio content still requires a moderate degree of expertise. Most consumer-level 360° cameras only record mono audio, and syncing an external spatial audio microphone can be expensive and technically challenging. As a consequence, while most video platforms (e.g., YouTube and Facebook) support spatial audio, it is often ignored by content creators, and at the time of submission, a random polling of 1000 YouTube 360° videos yielded *less than 5%* with spatial audio.

In order to close this gap between the audio and visual experiences, we introduce three main contributions: (1) we formalize the *360° spatialization* problem; (2) design the first 360° spatialization procedure; and (3) collect two datasets and propose an evaluation protocol to benchmark ours and future algorithms. 360° spatialization aims to *upconvert a single mono recording into spatial audio guided by full 360 view video*. More specifically, we seek to generate spatial audio in the form of a popular encoding format called first-order ambisonics (FOA), given the mono audio and corresponding 360° video. In addition to formulating the 360° spatialization

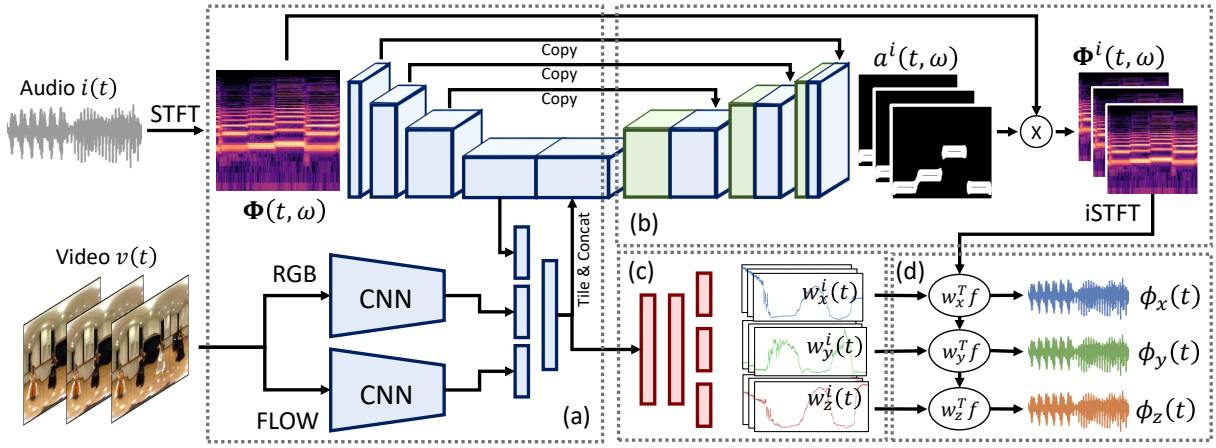


Figure 5.1: Architecture overview. Our approach is composed of four main blocks: analysis block; separation block; localization block; and ambisonics generation.

task, we design the first data-driven system to upgrade mono audio using self-supervision from 360° videos recorded with spatial audio. The proposed procedure is based on a novel neural network architecture that disentangles two fundamental challenges in audio spatialization: the separation of sound sources from a mixed audio input and respective localization. In order to train and validate our approach, we introduce two 360° video datasets with spatial audio, one recorded by ourselves in a constrained domain, and a large-scale dataset collected *in-the-wild* from YouTube. During training, the captured spatial audio serves as ground truth, with a mixed down mono version provided as input to our system. Experiments conducted in both datasets show that the proposed neural network can generate plausible spatial audio for 360° video. We further validate each component of the proposed architecture and show its superiority over a state-of-the-art, but domain-independent baseline architecture.

In the interest of reproducibility, code, data and trained models will be made available to the community at https://pedro-morgado.github.io/spatialaudiogen.

5.2 Related Work

To the best of our knowledge, we propose the first system for audio spatialization. In addition to spatial audio, the fields most related to our work are self-supervised learning, audio generation, source separation and audio-visual cross-modal learning, which we now briefly describe.

Spatial audio

Artificial environments, such as those rendered by game engines, can play sounds from any location in the video. This capability requires recording sound sources separately and mixing them according to the desired scene configuration (i.e., the positions of each source relative to the user). In a real world recording, however, sound sources cannot be recorded separately. In this case where sound sources are naturally mixed, spatial audio is often encoded using Ambisonics [62, 42, 107].

Ambisonics aims to approximate the sound pressure field at a single point in space using a spherical harmonic decomposition. More specifically, an audio signal $f(\boldsymbol{\theta},t)$ arriving from direction $\boldsymbol{\theta} = (\varphi,\vartheta)$ (where φ is the zenith angle and ϑ the azimuth angle) at time t is represented by a truncated spherical harmonic expansion of order N

$$f(\boldsymbol{\theta},t) = \sum_{n=0}^{N}\sum_{m=-n}^{n} Y_n^m(\varphi,\vartheta)\phi_n^m(t) \tag{5.1}$$

where $Y_n^m(\varphi,\vartheta)$ is the real spherical harmonic of order n and degree m, and $\phi_n^m(t)$ are the coefficients of the expansion. For ease of notation, Y_n^m and ϕ_n^m can be stacked into vectors $\boldsymbol{y}_N$ and $\boldsymbol{\phi}_N$, and (Eq. 5.1) written as $f(\boldsymbol{\theta},t) = \boldsymbol{y}_N^T(\boldsymbol{\theta})\boldsymbol{\phi}_N(t)$.

In a controlled environment, sound sources with *known* locations can be synthetically encoded into ambisonics using their spherical harmonic projection. More specifically, given a set

of k audio signals $s_1(t), \ldots, s_k(t)$ originating from directions $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k$,

$$\boldsymbol{\phi}_N(t) = \sum_{i=1}^{k} \boldsymbol{y}_N(\boldsymbol{\theta}_i) s_i(t). \tag{5.2}$$

For ambisonics playback, $\boldsymbol{\phi}_N$ is then *decoded* into a set of speakers or headphone signals in order to provide a plane-wave reconstruction of the sound field. In sum, the coefficients $\boldsymbol{\phi}_N$, also known as *ambisonic channels*, are sufficient to encode and reproduce spatial audio. Hence, our goal is to generate $\boldsymbol{\phi}_N$ from non-spatial audio and the corresponding video.

Self-supervised learning

Neural networks have been successfully trained through self- supervision for tasks such as image super-resolution [45, 100] and image colorization [88, 215]. In the audio domain, self-supervision has also enabled the detection of sound-video misalignment [148] and audio super-resolution [110]. Inspired by these approaches, we propose a self-supervised technique for audio spatialization. We show that the generation of ambisonic audio can be learned using a dataset of 360° video with spatial audio collected in-the-wild without additional human intervention.

Generative models

Recent advances in generative models such as Generative Adversarial Networks (GANs) [67] or Variational Auto-Encoders (VAE) [102] have enabled the generation of complex patterns, such as images [67] or text [95]. In the audio domain, Wavenet [146] has demonstrated the ability to produce high fidelity audio samples of both speech and music, by generating a waveform from scratch on a sample-by-sample basis. Furthermore, neural networks have also outperformed prior solutions to audio super-resolution [110] (e.g. converting from 4kHz to 16kHz audio) using a U-Net encoder-decoder architecture, and have enabled "automatic-Foley" type applications [179, 149]. In this work, instead of generating audio from scratch, our goal is to

augment the input audio channels so as to introduce spatial information. Thus, unlike Wavenet, efficient audio generation can be achieved without sacrificing audio fidelity, by transforming the input audio. We also demonstrate the advantages of our approach, inspired by the ambisonics encoding process in controlled environments, over a generic U-Net architecture.

Source separation

Source separation is a classic problem with an extensive literature. While early methods present the problem as independent component analysis, and focused on maximizing the statistical independence of the extracted signals [96, 38, 17, 5], recent approaches focus on data-driven solutions. For example, [87] proposes a recurrent neural-network for monaural separation of two speakers, [1, 56, 51] seek to isolate sound sources by leveraging synchronized visual information in addition to the audio input, and [197] studies a wide range of frequency-based separation methods. Similarly to recent trends, we rely on neural networks guided by cross-modal video analysis. However, instead of only separating human speakers [197] or musical instruments [221], we aim to separate multiple unidentified types of sound sources. Also, unlike previous algorithms, no explicit supervision is available to learn the separation block.

Source localization

Sound source localization is a mature area of signal processing and robotics research [10, 141, 140, 184]. However, unlike the proposed 360° spatialization problem, these works rely on microphone arrays using beamforming techniques [192] or binaural audio and HRTF cues similar to those used by humans [85]. Furthermore, the need for carefully calibrated microphones limits the applicability of these techniques to videos collected in-the-wild.

Cross visual-audio analysis

Cross-modal analysis has been extensively studied in the vision and graphics community, due to the inherently paired nature of video and audio. For example, [13] learns audio feature representations in an unsupervised setting by leveraging synchronized video. [90] segments and localizes dominant sources using clustering of video and sound features. Other methods correlate repeated motions with sounds to identify sources such as the strumming of a guitar using canonical correlation analysis [97, 98], joint embedding spaces [179, 149] or other temporal features [16].

5.3 Method

In this section, we define the 360° spatialization task to upconvert common audio recordings to support spatial audio playback. We then introduce a deep learning architecture to address this task, and two datasets to train the proposed architecture.

5.3.1 Audio spatialization

The goal of 360° spatialization is to generate ambisonic channels $\boldsymbol{\phi}_N(t)$ from non-spatial audio $i(t)$ and corresponding video $v(t)$. To handle the most common audio formats supported by commercial 360° cameras and video viewing platforms (e.g., YouTube and Facebook), we upgrade monaural recordings (mono) into first-order ambisonics (FOA). FOA consists of four channels that store the first-order coefficients, $\phi_0^0, \phi_1^{-1}, \phi_1^0$ and ϕ_1^1, of the spherical harmonic expansion in (Eq. 5.1). For ease of notation, we refer to these tracks as ϕ_w, ϕ_y, ϕ_z and ϕ_x, respectively.

Self-supervised audio spatialization

Converting mono to FOA ideally requires learning from videos with paired mono and ambisonics recordings, which are difficult to collect in-the-wild. In order to learn from self-

supervision, we assume that monaural audio is recorded with an omni-directional microphone. Under this assumption, mono is equivalent to zeroth-order ambisonics (up to an amplitude scale) and, as a consequence, the upconversion only requires the synthesis of the missing higher-order channels. More specifically, we learn to predict the first-order components $\phi_x(t), \phi_y(t), \phi_z(t)$ from the (surrogate) mono audio $i(t) = \phi_w(t)$ and video input $v(t)$. Note that the proposed framework is also applicable to other conversion scenarios, e.g. FOA to second-order ambisonics (SOA), simply by changing the number of input and output audio tracks (see Sec 5.5).

5.3.2 Architecture

Audio spatialization requires solving two fundamental problems: source separation and localization. In controlled environments, where the separated sound sources $s_i(t)$ and respective localization $\boldsymbol{\theta}_i$ are known in advance, ambisonics can be generated using (Eq. 5.2). However, since $s_i(t)$ and $\boldsymbol{\theta}_i$ are not known in practice, we design dedicated modules to isolate sources from the mixed audio input and localize them in the video. Also, because audio and video are complementary for identifying each source, both separation and localization modules are guided by a multi-modal audio-visual analysis module. A schematic description of our architecture is shown in Fig. 5.1. We now describe each component. Details of network architectures are provided in Appendix A.

Audio and visual analysis

Audio features are extracted in the time-frequency domain, which has produced successful audio representations for tasks such as audio classification [82] and speaker identification [139]. More specifically, we extract a sequence of short-term Fourier transforms (STFT) computed on 25ms segments of the input audio with 25% hop size and multiplied by Hann window functions. Then, we apply a (two-dimensional) CNN encoder to the audio spectrogram, which progressively reduces the spectrogram dimensionality and extracts high-level features.

Video features are extracted using a two-stream network, based on Resnet-18 [79], to encode both appearance (RGB frames) and motion (optical flow predicted by FlowNet2 [89]). Both streams are initialized with weights pre-trained on ImageNet [40] for classification, and fine-tuned on our task.

A joint audio-visual representation is then obtained by merging the three feature maps (audio, RGB and flow) produced at each time t. Since audio features are extracted at a higher frame rate than video features, we first synchronize the audio and video feature maps by nearest neighbor up-sampling of video features. Each feature map is then projected into a feature vector (1024 for audio and 512 for RGB and flow), and the outputs concatenated and fed to the separation and localization modules.

Audio separation

Although the number of sources may vary, this is often small in practice. Furthermore, psycoaccoustic studies have shown that humans can only distinguish a small number of simultaneous sources (three according to [172]). We thus assume an upper-bound of k simultaneous sources, and implement a separation network that extracts k audio tracks $f^i(t)$ from the input audio $i(t)$. The separation module takes the form of a U-Net decoder that progressively restores the STFT dimensionality through a series of transposed convolutions and skip connections from the audio analysis stage of equivalent resolution. Furthermore, to visually guide the separation module, we concatenate the multi-modal features to the lowest resolution layer of the audio encoder. In the last up-sampling layer, we produce k sigmoid activated maps $a^i(t,\omega)$, which are used to modulate the STFT of the mono input $\mathbf{\Phi}(t;\omega)$. The STFT of the i^{th} source $\mathbf{\Phi}^i(t;\omega)$ is thus obtained through the soft-attention mechanism $\mathbf{\Phi}^i(t;\omega) = a^i(t,\omega)\cdot\mathbf{\Phi}(t;\omega)$, and the separated audio track $f^i(t)$ reconstructed as the inverse STFT of $\mathbf{\Phi}^i(t;\omega)$ using an overlap-add method.

Localization

To localize the sounds $f^i(t)$ extracted by the separation network, we implement a module that generates, at each time t, the localization weights $\mathbf{w}^i(t) = (w_x^i(t), w_y^i(t), w_z^i(t))$ associated with each of the k sources, through a series of fully-connected layers applied to the multi-modal feature vectors of the analysis stage. In a parallel to the encoding mechanism of (Eq. 5.2) used in controlled environments, $\mathbf{w}^i(t)$ can be interpreted as the spherical harmonics $\mathbf{y}_N(\boldsymbol{\theta}_i(t))$ evaluated at the predicted position of the i^{th} source $\boldsymbol{\theta}_i(t)$.

Ambisonic generation

Given the localization weights $\mathbf{w}^i(t)$ and separated wave-forms $f^i(t)$, the first-order ambisonic channels $\boldsymbol{\phi}(t) = (\phi_x(t), \phi_y(t), \phi_z(t))$ are generated by

$$\boldsymbol{\phi}(t) = \sum_{i=1}^{k} \mathbf{w}^i(t) f^i(t). \tag{5.3}$$

In summary, we split the generation task into two components: generating the attenuation maps $a^i(t, \omega)$ for source separation, and the localization weights $\mathbf{w}^i(t)$. As audio is not generated from scratch, but through a transformation of the original input inspired by the encoding framework of (Eq. 5.2), we are able to achieve fast deployment speeds with high quality results.

5.3.3 Evaluation metrics

Let $\boldsymbol{\phi}(t)$ and $\hat{\boldsymbol{\phi}}(t)$ be the ground-truth and predicted ambisonics, and $\boldsymbol{\Phi}(t;\omega)$ and $\hat{\boldsymbol{\Phi}}(t;\omega)$ their respective STFTs. We now discuss several metrics used for evaluating the generated signals $\hat{\boldsymbol{\phi}}(t)$.

STFT distance

Our network is trained end-to-end to minimize errors between STFTs, i.e.,

$$MSE_{\text{stft}} = \sum_{p\in\{x,y,z\}} \sum_t \sum_\omega \|\Phi_p(t,\omega) - \hat{\Phi}_p(t,\omega)\|^2, \tag{5.4}$$

where $\|\cdot\|$ is the euclidean complex norm. MSE_{stft} has well-defined and smooth partial derivatives and, thus, it is a suitable loss function. Furthermore, unlike the euclidean distance between raw waveforms, the STFT loss is able to separate the signal into its frequency components, which enables the network to learn the easier parts of the spectrum without distraction from other errors.

Envelope distance (ENV)

Due to the high-frequency nature of audio and the human insensitivity to phase differences, frame-by-frame comparison of raw waveforms do not capture perceptual similarity of two audio signals. Instead, we measure the euclidean distance between *envelopes* of $\boldsymbol{\phi}(t)$ and $\hat{\boldsymbol{\phi}}(t)$.

Earth Mover's Distance (EMD)

Ambisonics model the sound field $f(\boldsymbol{\theta},t)$ over all directions $\boldsymbol{\theta}$. The energy of the sound field measured over a small window w_t around time t along direction $\boldsymbol{\theta}$ is

$$E(\boldsymbol{\theta},t) = \sqrt{\tfrac{1}{T}\sum_{\tau\in w_t} f(\boldsymbol{\theta},\tau)^2} = \sqrt{\tfrac{1}{T}\sum_{\tau\in w_t} \left(\mathbf{y}_N^T(\boldsymbol{\theta})\boldsymbol{\phi}_N(\tau)\right)^2}. \tag{5.5}$$

Thus, $E(\boldsymbol{\theta},t)$ represents the directional energy map of $\boldsymbol{\phi}(t)$. In order to measure the *localization* accuracy of the generated spatial audio, we propose to compute the EMD [119] between the energy maps $E(\boldsymbol{\theta},t)$ associated with $\boldsymbol{\phi}(t)$ and $\hat{\boldsymbol{\phi}}(t)$. In practice, we uniformly sample the maps $E(\boldsymbol{\theta},t)$ over the sphere, normalize the sampled map so that $\sum_i E(\boldsymbol{\theta}_i,t) = 1$, and measure the distance between samples over the sphere's surface using cosine (angular) distances for EMD

Figure 5.2: Example video frames from each dataset.

calculation.

5.3.4 Datasets

To train our model, we collected two datasets of 360° videos with FOA audio. The first dataset, denoted REC-STREET, was recorded by us using a Theta V 360° camera with an attached TA-1 spatial audio microphone. REC-STREET consists of 43 videos of outdoor street scenes, totaling 3.5 hours and 123k training samples (0.1s each). Due to the consistency of capture hardware and scene content, the audio of REC-STREET videos is relatively easier to spatialize.

The second dataset, denoted YT-ALL, was collected in-the-wild by scraping 360° videos from YouTube using queries related to spatial audio, e.g., *spatial audio, ambisonics*, and *ambix*. To clean the search results, we automatically removed videos that did not contain valid ambisonics, as described by YouTube's format, keeping only videos containing all 4 channels or with only the Z channel missing (a common spatial audio capture scenario). Finally, we performed a manual curation to remove videos that consisted of 1) still images, 2) computer generated content, or 3) containing post-processed and non-visually indicated sounds such as background music or voice-overs. During this pruning process, 799 videos were removed, resulting in 1146 valid videos

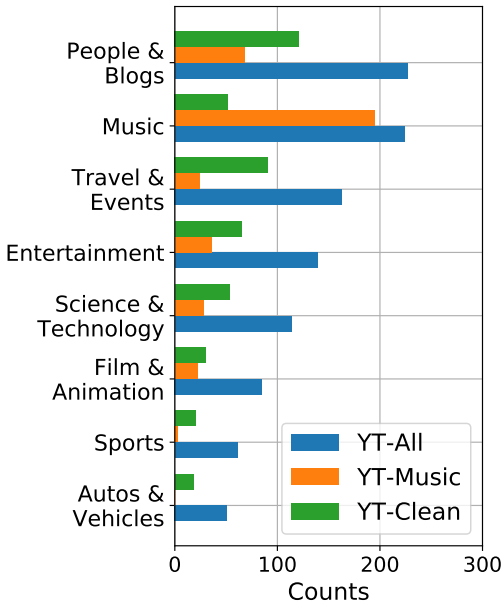


Figure 5.3: Dataset statistics

totaling 113.1 hours of content (3976k training samples). YT-ALL was further separated into live musical performances, YT-MUSIC (397 videos), and videos with a small number of superimposed sources which could be localized in the image, YT-CLEAN (496 videos). Upgrading YT-MUSIC videos into spatial audio is especially challenging due to the large number of mixed sources (voices and instruments). We also identified 489 videos that were recorded with a "horizontal" spatial audio microphone (i.e. only containing $\phi_w(t)$,$\phi_x(t)$ and $\phi_y(t)$ channels). In this case, we simply ignore the Z channel $\phi_z(t)$ when computing each metric including the STFT loss. Fig. 5.2 shows illustrative video frames and summarizes the most common categories for each dataset.

5.4 Evaluation

For our experiments, we randomly sample three partitions, each containing 75% of all videos for training and 25% for testing. Networks are trained to generate audio at 48kHz from

Table 5.1: Quantitative comparisons. We report three quality metrics: Envelope distance (ENV), Log-spectral distance (LSD), and earth-mover's distance (EMD).

	REC-STREET			YT-CLEAN			YT-MUSIC			YT-ALL		
	STFT	ENV	EMD	STFT	ENV	EMD	STFT	ENV	EMD	STFT	ENV	EMD
SPATIAL PRIOR	0.187	0.958	0.492	1.394	2.063	1.478	4.652	4.355	3.479	2.691	3.394	2.246
U-NET BASELINE	0.180	0.935	0.449	1.361	2.039	**1.403**	4.338	4.678	2.855	2.658	3.239	2.137
OURS-NOVIDEO	0.178	0.973	0.450	1.370	2.081	1.428	4.220	4.591	2.654	2.635	3.200	2.117
OURS-NORGB	**0.158**	0.779	**0.425**	**1.339**	1.847	**1.405**	3.664	3.569	2.432	2.546	2.907	2.063
OURS-NOFLOW	0.172	0.784	0.440	**1.349**	**1.778**	**1.402**	3.615	3.467	2.403	2.455	2.665	**2.023**
OURS-NOSEP	**0.152**	0.790	**0.422**	1.381	**1.773**	1.415	3.627	3.602	2.447	**2.435**	2.694	2.050
OURS-FULL	**0.158**	**0.767**	**0.419**	1.379	**1.776**	1.417	**3.524**	**3.366**	**2.350**	2.447	**2.649**	**2.019**

input mono audio processed at 48kHz and video at 10Hz. Each training sample consists of a chunk of 0.6s of mono audio and a single frame of RGB and flow, which are used to predict 0.1s of spatial audio at the center of the 0.6s input window. To make the model more robust and remove any bias to content in the center, we augment datasets during training by randomly rotating both video and spatial audio around the vertical (z) axis. Spatial audio can be rotated by multiplying the ambisonic channels with the appropriate rotation matrix as described in [107], and video frames (in equirectangular format) can be rotated using horizontal translations with wrapping. Networks are trained by back-propagation using the Adam optimizer [101] for 150k iterations (roughly two days) with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 1e-8$, batch size of 32, learning rate of $1e-4$ and weight decay of 0.0005. During evaluation, we predict a chunk of 0.1s for each second of the test video, and average the results across all chunks. Also, to avoid bias towards longer videos, all evaluation metrics are computed for each video separately, and averaged across videos.

5.4.1 Real time performance

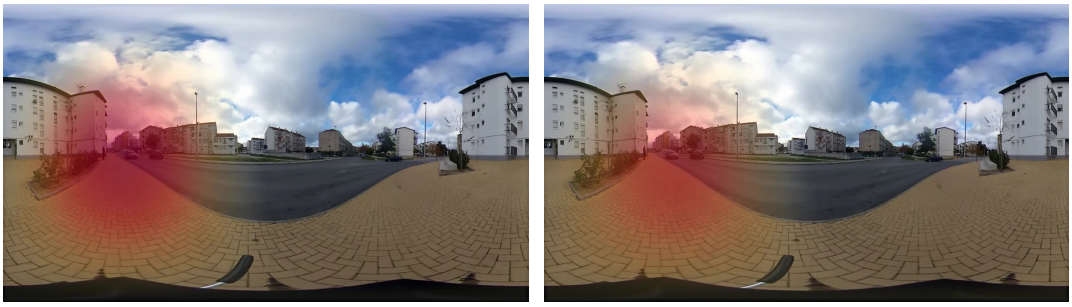
The proposed procedure can generate 1s of spatial audio at 48000Hz sampling rate in 103ms, using a single 12GB Titan Xp GPU (3840 cores running at 1.6GHz).

Figure 5.4: Qualitative Results. Comparison between predicted and recorded FOA. Spatial audio is visualized as a color overlay over the frame, with darker red indicating locations with higher audio energy.

Ground-truth U-NET

Ground-truth NOAUDIO

Ground-truth NOSEP



Ground-truth OURS

Figure 5.5: Comparison of predicted FOA produced by different procedures.

Figure 5.6: Predicted FOA on videos recorded with a real mono microphone (unknown FOA).

5.4.2 Baselines

Since spatial audio generation is a novel task, no established methods exist for comparison purposes. Instead, we ablate our architecture to determine the relevance of each component, and compare it to the prior spatial distribution of audio content and a popular, domain-independent baseline architecture. Quantitative results are shown in Table 5.1.

To determine the role of the visual input, we remove the RGB encoder (NORGB), the flow encoder (NOFLOW), or both (NOVIDEO). We also remove the separation block entirely (NOSEP), and multiply the localization weights with the input mono directly. The results indicate that the network is highly relying on visual features, with NOVIDEO being one of the worse performers overall. Interestingly, most methods performed well on REC-STREET and YT-CLEAN. However, the visual encoder and separation block are necessary for more complex videos as in YT-MUSIC and YT-ALL.

Since the main sound sources in 360° videos often appear in the center, we validate the need for a complex model by directly using the prior distribution of audio content (SPATIAL-

PRIOR). We compute the spatial prior $\bar{E}(\theta)$ by averaging the energy maps $E(\theta,t)$ of (Eq. 5.5) over all videos in the training set. Then, to induce the same distribution on test videos, we decompose $\bar{E}(\theta)$ into its spherical harmonics coefficients (c_w, c_x, c_y, c_z) and upconvert the input mono using $(\phi_w(t), \phi_x(t), \phi_y(t), \phi_z(t)) = (1, c_x/c_w, c_y/c_w, c_z/c_w)\, i(t)$. As shown in Table 5.1, relying solely on the prior distribution is not enough for accurate ambisonic conversion.

We finally compare to a popular encoder-decoder U-NET architecture, which has been sucessfully applied to audio tasks such as audio super-resolution [110]. This network consists of a number of convolutional downsampling layers that progressively reduce the dimension of the signal, distilling higher level features, followed by a number of upsampling layers to restore the signal's resolution. In each upsampling layer, a skip connection is added from the encoding layer of equivalent resolution. To generate spatial audio, we set the number of units in the output layer to the number of ambisonic channels, and concatenate video features to the U-Net bottleneck (i.e., the lowest resolution layer). See Appx. A for details. Our approach significantly outperforms the U-NET architecture, which demonstrates the importance of an architecture tailored to the task of spatial audio generation.

5.4.3 Qualitative results

Designing robust metrics for comparing spatial audio is an open problem, and we found that only so much can be determined by these metrics alone. For example, fully flat predictions can have a similar EMD to a mis-placed prediction, but perceptually be much worse. Therefore, we also rely on qualitative evaluation and a user study. Fig. 5.4 shows illustrative examples of the spatial audio output of our network, and Fig. 5.5 shows a comparison with other baselines. To depict spatial audio, we overlay the directional energy map $E(\boldsymbol{\theta},t)$ of the predicted ambisonics (Eq. 5.5) over the video frame at time t. As can be seen, our network generates spatial audio that has a similar spatial distribution of energy as the ground truth. Furthermore, due to the form of the audio generator, the sound fidelity of the original mono input is carried over to the synthesized

audio. These and other examples, together with the predicted spatial audio, are provided in Supp. material.

The results shown in Table 5.1 and Fig. 5.4 use videos recorded with ambisonic microphones and converted to mono audio. To validate whether our method extends to real mono microphones, we scraped additional videos from YouTube that were *not* recorded with ambisonics, and show that we can still generate convincing spatial audio (see Fig. 5.6 and Supp. material).

5.4.4 User study

The real criteria for success is whether viewers believe that the generated audio is correctly spatialized. To evaluate this, we conducted a "real vs fake" user study, where participants were shown a 360° video and asked to decide whether the perceived location of the audio matches the location of its sources in the video (real) or not (fake). Two studies were conducted in different viewing environments: a popular in-browser 360° video viewing platform (YouTube), and with a head-mounted display (HMD) in a controlled environment. We recruited 32 participants from Amazon Mechanical Turk for the in-browser study. For the HMD study, we recruited 9 participants (aged between 20 and 32, 1 female) through an engineering school email list of a large university. In both cases, participants were asked to have normal hearing, and to listen to the audio using headphones. In the HMD study, participants were asked to wear a KAMLE VR Headset. To familiarize participants with the spatial audio experience, each participant was first asked to watch two versions of a pre-selected video with and without correct spatial audio. After the practice round, participants watched 20 randomly selected videos whose audio was generated by one of four methods: GT, the original ground-truth recorded spatial audio; MONO, just the mono track (no spatialization); U-NET, the baseline method; and OURS, the result of our full method. After each video, participants were asked to decide whether its audio was real or fake. In total, 280 clips per method were watched for the in-browser study, and 45 per method in the HMD study.

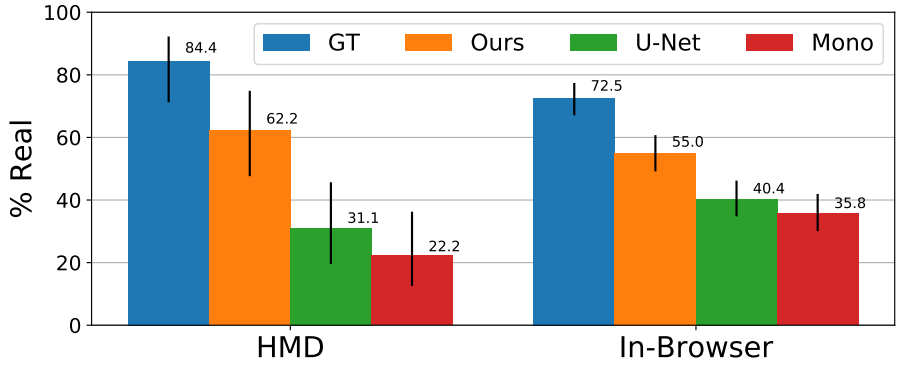


Figure 5.7: User studies results. Percentage of videos labeled as "Real" when viewed with audio generated by various methods (GT, OURS, U-NET and MONO) under two viewing experiences (using a HMD device, and in-browser viewing).

The results of both studies, shown in Fig 5.7, support several conclusions. First, our approach outperforms the U-NET baseline and MONO by statistically significant margins in both studies. Second, in comparison to in-browser video platforms, HMD devices offers a more realistic viewing experience, which enables non-spatial audio to be identified more easily. Thus, participants were convinced by the ambisonics predicted by our approach at higher rates while wearing an HMD device (62% HMD vs. 55% in-browser). Finally, spatial audio may not always be experienced easily, e.g., when the video does not contain clean sound sources. As a consequence, even videos with GT ambisonics were misclassified in both studies at a significant rate.

5.5 Discussion

Limitations

We observe several cases where sound sources are not correctly separated or localized. This occurs with challenging examples such as those with many overlapping sources, reverberant environments which are hard to separate, or where there is an ambiguous mapping from visual

Ground truth Ours

Figure 5.8: Limitations. Our algorithm predicts the wrong people who are laughing in a room full of people (top), and the wrong violin who is currently playing in the live performance (right).

Figure 5.9: Comparison of spatial resolution between first and second order ambisonics. Examples from our synthetic FOA to SOA conversion experiment.

	MONO → FOA	FOA → SOA
ENV	1.870	**0.333**
LSD	3.228	**0.513**
EMD	1.400	**0.232**

Figure 5.10: Comparison between Mono to FOA and FOA to SOA conversion tasks.

appearance to sound source (such as multiple, similar looking cars). Fig. 5.8 shows a few examples. While general purpose spatial audio generation is still an open problem, we provide a first approach. We hope that future advances in audio-visual analysis and audio generation will enable more robust solutions. Also, while total amount of content (in hours) is on par with other video datasets, the number of videos is still low, due to the limited number of 360° video with spatial audio available from online sources. As this number increases, our method should also improve significantly.

Future work

Although hardware trends change and we begin to see commercial cameras that include spatial audio microphone arrays capable of recording FOA, we believe that up-converting to spatial audio will remain relevant for a number of reasons. Besides the spatialization of legacy recordings with only mono or stereo audio, our method can be used to further increase spatial resolution, for example by up-converting first into second-order ambisonics (SOA). Unfortunately, ground-truth SOA recordings are difficult to collect in-the-wild, since SOA microphones are rare and expensive. Instead, to demonstrate future potential, we applied our approach to the FOA to SOA conversion task, using a small synthetic dataset where pre-recorded sounds are placed at chosen locations, which move over time in random trajectories. These are accompanied by an artificially constructed video consisting of a random background image with identifying icons synchronized with the sound location (see Fig. 5.10). The results shown in Fig. 5.10 indicate that converting FOA into SOA may be significantly easier than ZOA to FOA. This is because FOA signals already contain substantial spatial information, and partially separated sounds. Given these findings, a promising area for future work is to synthesize a realistic large scale SOA dataset for learning to convert FOA into high-order ambisonics and in order to support more realistic viewing experience.

5.6 Conclusion

We presented the first approach for up-converting conventional mono recordings into spatial audio given a 360° video, and introduced an end-to-end trainable network tailored to this task. We also demonstrate the benefits of each component of our network and show that the proposed generator performs substantially better than a domain independent baseline.

5.7 Appendix

5.7.1 Network Architectures

Both video and flow encoders use the ResNet-18 architecture up to the last convolutional layer. Then, a 1x1 convolutional layer reduces the dimensionality of the feature maps to 128, and a fully-connected layer is used to compute from the resulting map of size 7x14x128, a 512-dimensional global feature vector. Flow features are extracted from the X and Y displacements, as well as the magnitude of the corresponding velocity vector. The audio encoder is a 5 layer CNN applied to the input STFT and detailed in Fig. 5.11.

The concatenated audio and video features are then fed to the separation and localization blocks, shown in Figs. 5.12 and 5.13, respectively. The separation net outputs the $k = 32$ frequency activation maps to be used for modulation of the input STFT, and separated wave-forms $f^i(t)$ are computed by inverse STFT. In our implementation, the number of frequency components is 1024. The localization net outputs, for each of the $k = 32$ sources, the 3 localization weights $\mathbf{w}^i$ associated with the three ambisonics channels $\boldsymbol{\phi} = (\phi_x, \phi_y, \phi_z)$.

Given the localization weights $\mathbf{w}^i(t)$ and separated wave-forms $f^i(t)$, the FOA are generated by $\boldsymbol{\phi}(t) = \sum_{i=1}^{k} \mathbf{w}^i(t) f^i(t)$.

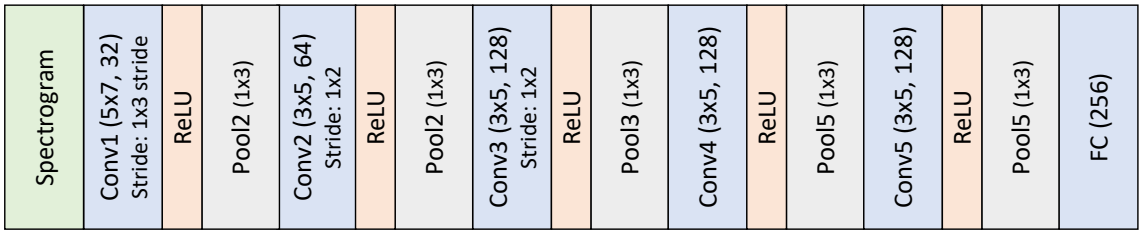


Figure 5.11: Detailed representation of audio encoder architecture. Forward pass is left to right.

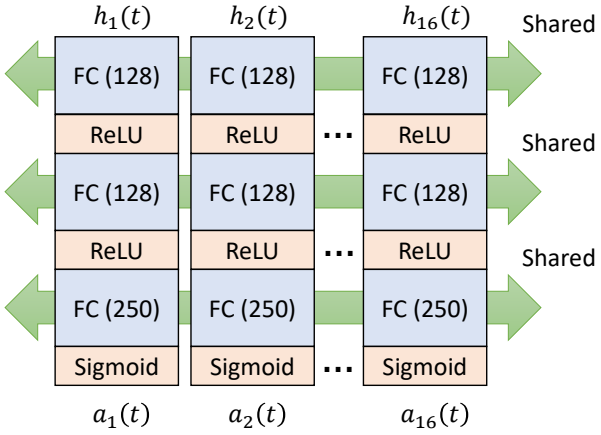


Figure 5.12: Detailed representation of source separation architecture. Forward pass is top to bottom.

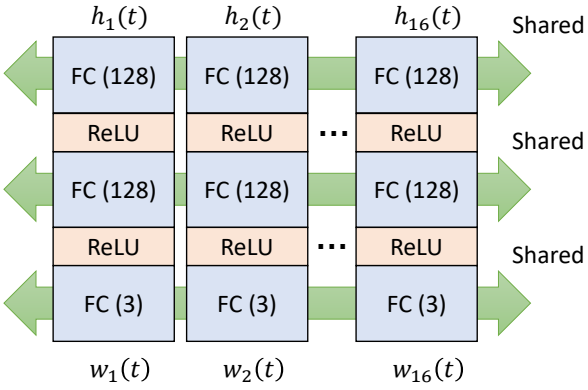


Figure 5.13: Detailed representation of localization architecture. Forward pass is top to bottom.

5.8 Acknowledgements

Chapter 5 is, in full, based on the material as it appears in the publication of “Self-Supervised Generation of Spatial Audio for 360 Video”, Pedro Morgado, Nuno Vasconcelos, Timothy Langlois and Oliver Wang, as appears in the Advances of Neural Information Processing Systems (NeurIPS), 2018. The dissertation author was the primary investigator and author of this material.

Chapter 6

Conclusion

In this thesis, we have studied the value of audio as targets for self-supervision. We have proposed a series of self-supervised learning frameworks that build on each other to learn better models, and accomplish more with less human annotations.

We started by demonstrating that paired modalities, like audio and vision, offer correlated but distinct views of an instance. As a result, they can be used to form strong positive pairs for contrastive learning. We proposed a strong self-supervised learning task that can learn powerful representations by leveraging these audio-visual associations. The learned representations were proven useful for human action recognition and environmental sound classification tasks.

We then identified a major challenge with audio supervision, *i.e.*, the existence of faulty audiovisual correspondences. We showed that, if left unaddressed, these faulty correspondences can be detrimental and proposed effective solutions for them. The proposed method addresses the two main sources of noisy training signals: faulty positives and faulty negatives. Faulty positives are discounted by down-weighting instances with poor audio-visual correspondence. Faulty negatives are addressed by optimizing the loss over a soft target distribution that encodes instance similarity. This work shows that self-supervised learning, in general, should be treated as a problem of learning with noisy targets.

We then showed that the lack of spatial grounding of audio signals can lead to misleading associations. This was addressed by solving a spatial alignment task where models are required to reason about the spatial context of audio and visual signals. We also propose a new cross-modal translation network that can translate between audio and visual features from multiple viewpoints of a 360 video. The proposed model is inspired by a class of models, called transformers, very popular for language translation and NLP in general, and combines representations from multiple viewpoints through a stack of self-attention layers. Extensive experiments show that effective localization is critical to maximize the value of audio supervision.

Finally, we introduced a new task, spatial audio generation, where the goal is to upgrade the mono audio of 360 videos. We also developed an architecture inspired by the spatial audio

generation process that proved effective for solving this task. Spatial audio generation is an example of a practical task for which human annotations would be hard or even impossible to collect.

In general, these works show that naturally co-occurring sensory signals like audio and video can be used as a target to learn powerful representations for visual inputs without relying on costly human annotations. When done properly, audiovisual learning can benefit many applications, including representation learning for action and audio recognition, visually-driven sound source localization, and spatial sound generation.

Bibliography

[1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. The conversation: Deep audio-visual speech enhancement. In *Conf. of the International Speech Communication Association (EuroCOLT)*, 2018.

[2] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Deep lip reading: A comparison of models and an online application. In *Conf. of the International Speech Communication Association (EuroCOLT)*, 2018.

[3] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-Supervised MultiModal Versatile Networks. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2020.

[4] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *Adv. Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Shun-ichi Amari, Andrzej Cichocki, Howard Hua Yang, et al. A new learning algorithm for blind signal separation. In *Adv. Neural Information Processing Systems (NeurIPS)*, pages 757–763. MIT Press, 1996.

[6] Tobias S Andersen, Kaisa Tiippana, and Mikko Sams. Factors influencing audiovisual fission and fusion illusions. *Cognitive Brain Research*, 21(3):301–308, 2004.

[7] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *Eur. Conf. Computer Vision (ECCV)*, 2018.

[8] Relja Arandjelović and Andrew Zisserman. Look, listen and learn. *Int. Conf. Computer Vision (ICCV)*, 2017.

[9] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin Mcguinness. Unsupervised label noise modeling and loss correction. In *Int. Conf. on Machine Learning (ICML)*, 2019.

[10] Sylvain Argentieri, Patrick Danès, and Philippe Souères. A survey on sound source localization in robotics: From binaural to array processing methods. *Computer Speech & Language*, 34(1):87–112, 2015.

[11] Yuki M. Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2020.

[12] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *Int. Conf. on Machine Learning (ICML)*, 2019.

[13] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2016.

[14] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[15] Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Int. Conf. on Machine Learning (ICML)*, 2004.

[16] Zohar Barzelay and Yoav Y Schechner. Harmony in motion. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007.

[17] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[18] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Int. Conf. on Machine Learning (ICML)*, pages 41–48, 2009.

[20] Steffen Bickel and Tobias Scheffer. Multi-view clustering. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2004.

[21] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Conf. on Computational Learning Theory*, 1998.

[22] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *Int. Conf. on Machine Learning (ICML)*, 2017.

[23] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *Eur. Conf. Computer Vision (ECCV)*, 2018.

[24] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 6154–6162, 2018.

[25] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 43(1):172–186, 2019.

[26] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Eur. Conf. Computer Vision (ECCV)*, pages 213–229. Springer, 2020.

[27] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Eur. Conf. Computer Vision (ECCV)*, 2018.

[28] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Int. Conf. Computer Vision (ICCV)*, 2019.

[29] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2020.

[30] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2017.

[31] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *Eur. Conf. Computer Vision (ECCV)*, 2020.

[32] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Int. Conf. on Machine Learning (ICML)*, 2020.

[33] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[34] Hsien-Tzu Cheng, Chun-Hung Chao, Jin-Dong Dong, Hao-Kai Wen, Tyng-Luh Liu, and Min Sun. Cube padding for weakly-supervised saliency prediction in 360 videos. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1420–1429, 2018.

[35] Hyeon Cho, Taehoon Kim, Hyung Jin Chang, and Wonjun Hwang. Self-supervised spatio-temporal representation learning using variable playback speed prediction. *arXiv preprint arXiv:2003.02692*, 2020.

[36] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Lip reading sentences in the wild. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[37] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Asian Conf. Computer Vision (ACCV)*, 2016.

[38] Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287 – 314, 1994. Higher Order Statistics.

[39] Virginia R de Sa. Learning classification with unlabeled data. In *Adv. Neural Information Processing Systems (NeurIPS)*, 1994.

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[41] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *Int. Conf. Computer Vision (ICCV)*, 2015.

[42] Glenn Dickins and Rodney Kennedy. Towards optimal soundfield representation. In *Audio Engineering Society Convention 106*, May 1999.

[43] Tom Diethe, David R Hardoon, and John Shawe-Taylor. Multiview fisher discriminant analysis. In *Workshop in Advances in Neural Information Processing Systems (NeurIPS)*, 2008.

[44] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Int. Conf. Computer Vision (ICCV)*, 2015.

[45] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Eur. Conf. Computer Vision (ECCV)*, pages 184–199. Springer, 2014.

[46] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 38(9):1734–1747, 2016.

[47] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2014.

[48] Fanyi Duanmu, Yixiang Mao, Shuai Liu, Sumanth Srinivasan, and Yao Wang. A subjective study of viewer navigation behaviors when watching 360-degree videos on computers. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018.

[49] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[50] Lauren L Emberson, John E Richards, and Richard N Aslin. Top-down modulation in the infant brain: Learning-induced expectations rapidly affect the sensory cortex at 6 months. *National Academy of Sciences*, 112(31):9585–9590, 2015.

[51] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. *ACM Trans. on Graphics (TOG)*, 37(4):1–11, 2018.

[52] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Int. Conf. Computer Vision (ICCV)*, 2017.

[53] Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning by rotation feature decoupling. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[54] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[55] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2013.

[56] Aviv Gabbay, Asaph Shamir, and Shmuel Peleg. Visual speech enhancement using noise-invariant training. *arXiv preprint arXiv:1711.08789*, 2017.

[57] Chuang Gan, Deng Huang, Hang Zhao, Joshua B Tenenbaum, and Antonio Torralba. Music gesture for visual sound separation. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[58] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *Eur. Conf. Computer Vision (ECCV)*, 2018.

[59] Ruohan Gao and Kristen Grauman. 2.5d visual sound. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[60] Ruohan Gao and Kristen Grauman. Co-separating sounds of visual objects. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[61] Jort F Gemmeke, Daniel PW 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 *Int. Conf. on Acoustics, Speech and Signal Processing*, 2017.

[62] Michael A Gerzon. Periphony: With-height sound reproduction. *Journal of the audio engineering society*, 21(1):2–10, 1973.

[63] Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI Conference on Artificial Intelligence*, 2017.

[64] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Int. Conf. Learning Representations (ICLR)*, 2018.

[65] Ross Girshick. Fast r-cnn. In *Int. Conf. Computer Vision (ICCV)*, pages 1440–1448, 2015.

[66] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2014.

[67] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Adv. Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

[68] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Int. Conf. Computer Vision (ICCV)*, 2019.

[69] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *ICAIS*, 2010.

[70] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2006.

[71] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2018.

[72] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Eur. Conf. Computer Vision Workshops (ECCV-W)*, 2019.

[73] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *Eur. Conf. Computer Vision (ECCV)*, 2020.

[74] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2020.

[75] David Harwath, Antonio Torralba, and James Glass. Unsupervised learning of spoken language with visual context. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2016.

[76] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[77] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Int. Conf. Computer Vision (ICCV)*, pages 2961–2969, 2017.

[78] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[79] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Eur. Conf. Computer Vision (ECCV)*, pages 630–645. Springer, 2016.

[80] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *Int. Conf. on Machine Learning (ICML)*. PMLR, 2020.

[81] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2018.

[82] 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 *Int. Conf. on Acoustics, Speech and Signal Processing*, pages 131–135. IEEE, 2017.

[83] 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 *Int. Conf. Learning Representations (ICLR)*, 2018.

[84] Chih-Hui Ho and Nuno Vasconcelos. Contrastive learning with adversarial examples. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2020.

[85] Jonas Hornstein, Manuel Lopes, José Santos-Victor, and Francisco Lacerda. Sound localization for humanoid robots-building audio-motor maps based on the HRTF. In *IEEE/RSJ International Conf. on Intelligent Robots and Systems*, 2006.

[86] Hou-Ning Hu, Yen-Chen Lin, Ming-Yu Liu, Hsien-Tzu Cheng, Yung-Ju Chang, and Min Sun. Deep 360 pilot: Learning a deep agent for piloting through 360 sports videos. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1396–1405. IEEE, 2017.

[87] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Deep learning for monaural speech separation. In *Int. Conf. on Acoustics, Speech and Signal Processing*, pages 1562–1566. IEEE, 2014.

[88] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Trans. on Graphics (TOG)*, 35(4):110, 2016.

[89] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017.

[90] Hamid Izadinia, Imran Saleemi, and Mubarak Shah. Multimodal analysis for identification and segmentation of moving-sounding objects. *IEEE Transactions on Multimedia*, 15(2):378–390, 2013.

[91] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information distillation for unsupervised image segmentation and clustering. *arXiv preprint arXiv:1807.06653*, 2018.

[92] Huaizu Jiang, Gustav Larsson, Michael Maire Greg Shakhnarovich, and Erik Learned-Miller. Self-supervised relative depth learning for urban scene understanding. In *Eur. Conf. Computer Vision (ECCV)*, 2018.

[93] Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018.

[94] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv preprint arXiv:1811.11387*, 2018.

[95] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

[96] Christian Jutten and Jeanny Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1 – 10, 1991.

[97] Einat Kidron, Yoav Y Schechner, and Michael Elad. Pixels that sound. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005.

[98] Einat Kidron, Yoav Y Schechner, and Michael Elad. Cross-modal localization via sparsity. *IEEE transactions on signal processing*, 55(4):1390–1404, 2007.

[99] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI Conference on Artificial Intelligence*, 2019.

[100] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654, 2016.

[101] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[102] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *stat*, 1050:1, 2014.

[103] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 9404–9413, 2019.

[104] Marius Kloft and Gilles Blanchard. The local rademacher complexity of lp-norm multiple kernel learning. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2011.

[105] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2018.

[106] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2012.

[107] Matthias Kronlachner. Spatial transformations for the alteration of ambisonic recordings. Master's thesis, Graz University of Technology, 2014.

[108] Matthias Kronlachner and Franz Zotter. Spatial transformations for the enhancement of ambisonic recordings. In *International Conference on Spatial Audio*, 2014.

[109] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *Int. Conf. Computer Vision (ICCV)*. IEEE, 2011.

[110] Volodymyr Kuleshov, S. Zayd Enam, and Stefano Ermon. Audio super resolution using neural networks. *CoRR*, abs/1708.00853, 2017.

[111] Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2011.

[112] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5(Jan):27–72, 2004.

[113] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *Eur. Conf. Computer Vision (ECCV)*, 2016.

[114] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[115] Quoc V Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S Corrado, Jeff Dean, and Andrew Y Ng. Building high-level features using large scale unsupervised learning. In *Int. Conf. on Acoustics, Speech and Signal Processing*, 2013.

[116] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1, 1989.

[117] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2007.

[118] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[119] Elizaveta Levina and Peter Bickel. The earth mover's distance is the mallows distance: Some insights from statistics. In *Int. Conf. Computer Vision (ICCV)*, volume 2, pages 251–256. IEEE, 2001.

[120] Benjamin J Li, Jeremy N Bailenson, Adam Pines, Walter J Greenleaf, and Leanne M Williams. A public database of immersive vr videos with corresponding ratings of arousal, valence, and correlations between head movements and self report measures. *Frontiers in psychology*, 8:2116, 2017.

[121] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[122] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *Int. Conf. Computer Vision (ICCV)*, 2017.

[123] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[124] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Eur. Conf. Computer Vision (ECCV)*. 2014.

[125] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[126] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. In *AAAI Conference on Artificial Intelligence*, 2020.

[127] Fan Ma, Deyu Meng, Qi Xie, Zina Li, and Xuanyi Dong. Self-paced co-training. In *Int. Conf. on Machine Learning (ICML)*, 2017.

[128] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *Int. Conf. on Machine Learning (ICML)*, 2018.

[129] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-svms for object detection and beyond. In *Int. Conf. Computer Vision (ICCV)*, 2011.

[130] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *ICANN*. Springer, 2011.

[131] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264(5588):746–748, 1976.

[132] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

[133] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[134] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Eur. Conf. Computer Vision (ECCV)*, 2016.

[135] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *Int. Conf. on Machine Learning (ICML)*, 2009.

[136] Pedro Morgado, Yi Li, and Nuno Vasconcelos. Learning representations from audio-visual spatial alignment. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2020.

[137] Pedro Morgado, Nuno Vasconcelos, Timothy Langlois, and Oliver Wang. Self-supervised generation of spatial audio for 360 video. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2018.

[138] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021.

[139] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: A large-scale speaker identification dataset. *Conf. of the International Speech Communication Association (EuroCOLT)*, pages 2616–2620, 2017.

[140] Kazuhiro Nakadai, Hiroshi G Okuno, and Hiroaki Kitano. Real-time sound source localization and separation for robot audition. In *International Conference on Spoken Language Processing*, 2002.

[141] Keisuke Nakamura, Kazuhiro Nakadai, Futoshi Asano, and Gökhan Ince. Intelligent sound source localization and its application to multimodal human tracking. In *IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[142] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Eur. Conf. Computer Vision (ECCV)*, 2016.

[143] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Int. Conf. Computer Vision (ICCV)*, 2017.

[144] Bruno A Olshausen. Sparse coding of time-varying natural images. In *Proc. of the Int. Conf. on Independent Component Analysis and Blind Source Separation*, 2000.

[145] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.

[146] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[147] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[148] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Eur. Conf. Computer Vision (ECCV)*, 2018.

[149] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[150] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *Eur. Conf. Computer Vision (ECCV)*, 2016.

[151] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Int. Conf. Computer Vision (ICCV)*, 2015.

[152] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Adv. Neural Information Processing Systems (NeurIPS)*. 2019.

[153] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[154] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[155] 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.

[156] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[157] Greire Payen de La Garanderie, Amir Atapour Abarghouei, and Toby P Breckon. Eliminating the blind spot: Adapting 3d object detection and monocular depth estimation to 360 panoramic imagery. In *Eur. Conf. Computer Vision (ECCV)*, pages 789–807, 2018.

[158] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

[159] Karol J Piczak. Environmental sound classification with convolutional neural networks. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015.

[160] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press, 2015.

[161] AJ Piergiovanni, Anelia Angelova, and Michael S. Ryoo. Evolving losses for unsupervised video representation learning. *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[162] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[163] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021.

[164] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Eur. Conf. Computer Vision (ECCV)*, 2018.

[165] Novi Quadrianto and Christoph Lampert. Learning multi-view neighborhood preserving projections. In *Int. Conf. on Machine Learning (ICML)*, 2011.

[166] Marc'aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2007.

[167] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[168] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *Int. Conf. on Machine Learning (ICML)*, 2018.

[169] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *WACV*, 2005.

[170] Hardik B Sailor, Dharmesh M Agrawal, and Hemant A Patil. Unsupervised filterbank learning using convolutional restricted boltzmann machine for environmental sound classification. In *Conf. of the International Speech Communication Association (EuroCOLT)*, 2017.

[171] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455, 2009.

[172] Olli Santala and Ville Pulkki. Directional perception of distributed sound sources. *The Journal of the Acoustical Society of America*, 129(3):1522–1530, 2011.

[173] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[174] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Int. Conf. Computer Vision (ICCV)*, pages 618–626, 2017.

[175] Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, and In So Kweon. Learning to localize sound source in visual scenes. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[176] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2018.

[177] Ladan Shams, Yukiyasu Kamitani, and Shinsuke Shimojo. What you see is what you hear. *Nature*, 408(6814):788–788, 2000.

[178] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2019.

[179] Matthias Soler, Jean-Charles Bazin, Oliver Wang, Andreas Krause, and Alexander Sorkine-Hornung. Suggesting sounds for images from video collections. In *Eur. Conf. Computer Vision (ECCV)*, pages 900–917. Springer, 2016.

[180] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *Int. Conf. on Machine Learning (ICML)*, 2019.

[181] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*, 2020.

[182] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. Technical Report CRCV-TR-12-01, University of Central Florida, 2012.

[183] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D Plumbley. Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 17(10):1733–1746, 2015.

[184] N Strobel, S Spors, and R Rabenstein. Joint audio-video object localization and tracking. *IEEE Signal Processing Magazine*, 18(1):22–31, 2001.

[185] Yu-Chuan Su, Dinesh Jayaraman, and Kristen Grauman. Pano2vid: Automatic cinematography for watching 360°videos. In *Asian Conference on Computer Vision (ACCV)*, 2016.

[186] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019.

[187] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015.

[188] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Int. Conf. Computer Vision (ICCV)*, pages 5229–5238, 2019.

[189] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Int. Conf. on Machine Learning (ICML)*, pages 6105–6114. PMLR, 2019.

[190] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Eur. Conf. Computer Vision (ECCV)*, 2020.

[191] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[192] Jean-Marc Valin, François Michaud, and Jean Rouat. Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering. *Robotics and Autonomous Systems*, 55(3):216–228, 2007.

[193] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[194] Petra Vetter, Fraser W Smith, and Lars Muckli. Decoding sound and imagery content in early visual cortex. *Current Biology*, 24(11):1256–1262, 2014.

[195] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Int. Conf. on Machine Learning (ICML)*. ACM, 2008.

[196] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *arXiv:1705.06950*, 2017.

[197] DeLiang Wang and Jitong Chen. Supervised speech separation based on deep learning: an overview. *arXiv preprint arXiv:1708.07524*, 2017.

[198] Fu-En Wang, Hou-Ning Hu, Hsien-Tzu Cheng, Juan-Ting Lin, Shang-Ta Yang, Meng-Li Shih, Hung-Kuo Chu, and Min Sun. Self-supervised learning of depth and camera motion from 360°videos. In *Asian Conf. Computer Vision (ACCV)*, pages 53–68. Springer, 2018.

[199] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *Eur. Conf. Computer Vision (ECCV)*, 2020.

[200] Wei Wang and Zhi-Hua Zhou. Analyzing co-training style algorithms. In *Proceeding of the European Conference on Machine Learning (ECML)*. Springer, 2007.

[201] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Int. Conf. Computer Vision (ICCV)*, 2015.

[202] Xudong Wang, Ziwei Liu, and Stella X Yu. Unsupervised feature learning by cross-level discrimination between instances and groups. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2020.

[203] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Int. Conf. Computer Vision (ICCV)*, 2019.

[204] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[205] Eric W Weisstein. Gnomonic projection. *From MathWorld–A Wolfram Web Resource. http://mathworld. wolfram. com/GnomonicProjection. html*, 2020.

[206] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[207] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2695–2702. IEEE, 2012.

[208] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Delving into inter-image invariance for unsupervised visual representations. *arXiv preprint arXiv:2008.11702*, 2020.

[209] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1500, 2017.

[210] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[211] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[212] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *Int. Conf. on Machine Learning (ICML)*, 2017.

[213] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *Int. Conf. Learning Representations (ICLR)*, 2018.

[214] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[215] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Eur. Conf. Computer Vision (ECCV)*, 2016.

[216] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[217] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2018.

[218] Ziheng Zhang, Yanyu Xu, Jingyi Yu, and Shenghua Gao. Saliency detection in 360 videos. In *Eur. Conf. Computer Vision (ECCV)*, pages 488–503, 2018.

[219] Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[220] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *Int. Conf. Computer Vision (ICCV)*, 2019.

[221] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Eur. Conf. Computer Vision (ECCV)*, 2018.

[222] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 8856–8865, 2019.

[223] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Int. Conf. Computer Vision (ICCV)*, 2019.