Video Understanding Using Audio and Visual Modalities

Jaesung Huh

St Cross College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2024

Abstract

This thesis explores the field of audio-visual learning in the context of video understanding, focusing on two main aspects: developing novel methods for effectively integrating audio and visual modalities, and curating high-quality audio-visual datasets automatically. By leveraging the complementary nature of these modalities and addressing the challenge of dataset creation, we enhance machine perception and comprehension of video content. We overcome the limitations of single-modality processing and improve performance in various practical applications. Our work also contributes to the efficient generation of large-scale, diverse datasets crucial for progress in this field.

Our research addresses three key areas: action recognition, character-aware subtitle generation for TV shows, and efficient audio-visual dataset curation. In action recognition, we develop models that effectively integrate audio-visual cues to improve accuracy, particularly by utilising temporal context from the video. For subtitle generation, we propose a multimodal approach that not only transcribes speech accurately but also attributes dialogue to correct speakers and aligns subtitles precisely with audio content. In dataset curation, we tackle the challenge of creating large-scale, diverse, and accurately labeled audio-visual datasets, developing efficient methods to accelerate progress in the field.

Throughout this work, we introduce novel architectures and algorithms that effectively combine audio and visual information, as well as propose new datasets and automatic creation pipelines to reduce the cost of data collection and human annotation. Our approach is inspired by psychological research on human multisensory integration and aims to mimic human-like processing of audio-visual information.

Keywords – Video understanding, audio-visual learning, multimodal learning

This thesis is submitted to the Department of Engineering Science, The University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Jaesung Huh, Oct 2024.

Acknowledgement

No great journey is ever undertaken alone. I am grateful for having had fabulous opportunities to work with amazing colleagues. This thesis would not have been possible without their assistance and contributions.

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Andrew Zisserman, for his invaluable guidance and encouragement throughout my doctoral studies. I still remember my first year in Oxford, struggling with a lack of knowledge in computer vision, and he generously gave me an hour-long lecture for answering my question *"What is optical flow?"*. I am profoundly thankful for his continuous mentorship and for teaching me how to conduct research. I could not have found a better captain to guide me through my PhD journey.

To the Oxford-Bristol audio-visual team, those who introduced me to the new field and also being the best collaborators I could ever ask for. To Jacob Chalk for being both a great colleague who happened to work with my indecipherable code and also a great friend who classified clang and clatter sounds with me for months. To Evangelos Kazakos, who taught me how to write a paper with meticulous attention to detail and a considerate attitude towards research. I especially thank Dima Damen, whom I always considered as my co-supervisor. She continued to support me and show me how to conduct research with passion and integrity.

To the VoxSRC team, who have contributed to the research community through their creation of numerous speech datasets and successful hosting of VoxSRC workshops. To Joon Son Chung for introducing me to the field of machine learning research. To Arsha Nagrani and Andrew Brown for their warm welcome when I first joined VGG and their invaluable help in acclimating me to the group. To Jee-weon Jung for his constant willingness to offer advice and hands-on assistance whenever needed. Lastly, to Daniel Garcia-Romero for sharing his vast experience and providing insightful advice on hosting these workshops.

To other VGG group members, Ragav, Prajwal, Noel, Bruno, Guanqi, Shu, Luke, Vadim, Laurynas, Jensen, Junyu, Ray, Suny, Yash, Vladimir, Akam, Andreea, Paul, Max, Tengda, Chuhan, Shivani, Dan, Lili, and everyone else I might have missed, thank you for creating a wonderful research environment and for being great friends.

To my family, thank you for your love and support. To my mom Sunghyun who have stood by me in every decision I have made with endless love and understanding. I will never forget the warm and gentle smile you showed me when you saw me in tears during my difficult times. To my dad, ChangJin, thank you for always sitting down with me to discuss my struggles. Your advice on embracing life with joy and happiness, even when under stress, has been a constant source of strength. To my sister Minjung, who always listened to the challenges I faced during my PhD journey and offered invaluable advice, drawing from her own experiences as a fellow PhD student.

And finally, to Nayoon, who has been with me through every step of my PhD journey. Thank you for believing in my dream before I did. I wasn't even sure what my dream was when I told you, in that cafe inside the NC department store, that I had decided to apply to this graduate program. This thesis is as much yours as it is mine.

Contents

Chapter 1

Introduction and Background

Humans naturally process information from multiple senses simultaneously. For instance, when crossing a street, pedestrians visually scan for oncoming traffic while listening for vehicle sounds. Similarly, during face-to-face conversations, we integrate visual cues from lip movements and hand gestures with auditory information from speech to fully comprehend the message. Audio-visual learning aims to give machines similar capabilities, allowing them to merge data from different sources.

This thesis aims to explore and advance the field of audio-visual learning, particularly in the context of video understanding. The objective is to develop novel methods for effectively integrating audio and visual modalities to enhance machine perception and comprehension of video content. We aim to improve performance over single-modality processing in various practical applications, such as human behavior analysis or understanding the dialogue in TV shows.

Despite the potential of audio-visual learning, integrating audio and visual data presents several challenges. First, current audio-visual video understanding works fail to fully utilise the temporal context of the video, limiting their ability to capture long-term dependencies and relationships between modalities and events over time. Humans tend to relate the present to the past and the future to understand video content by watching and listening. However, there is a lack of efficient approaches that effectively leverage multiple modalities in conjunction with temporal context. In addition, existing approaches often focus on solving isolated subtasks rather than aiming for a holistic understanding of video content, particularly in

complex scenarios like human conversations. This fragmented approach hinders the development of comprehensive audio-visual understanding systems. Lastly, there is a critical shortage of both efficient dataset curation pipelines and the datasets themselves in the audio-visual domain. The lack of cost-effective methods for collecting and annotating multimodal data, coupled with the scarcity of large-scale, high-quality audio-visual datasets, impedes the development of more advanced audio-visual learning models.

To address these challenges, this thesis proposes novel approaches for audio-visual integration in video analysis. We focus on three key areas: (i) action recognition leveraging temporal context of video, (ii) character-aware subtitle generation for TV shows, and (iii) efficient audio-visual dataset curation pipelines. Our research introduces novel architectures and algorithms that effectively combine audio and visual information, as well as datasets and their automatic creation pipeline to reduce the cost of data collection and human annotation. Through these contributions, we aim to advance the field of audio-visual learning, enhancing machine understanding of complex, real-world scenarios.

1.1 Motivation and Background

Motivation from psychology The motivation for audio-visual learning in artificial intelligence is rooted in our understanding of human cognition and perception. Psychological research has established that humans process information through multiple sensory channels simultaneously, a phenomenon known as multisensory integration [Spence 2007; Stein and Stanford 2008; Mudrik et al. 2014]. This complex process, where information from different senses interacts and influences overall perception, is exemplified by the McGurk effect [McGurk and MacDonald 1976]. It demonstrates how visual information from lip movements can alter the perception of speech sounds, highlighting the interplay between auditory and visual processing in the brain. Multimodal perception are not only present in adults but are also crucial during infant development. Infants as young as five months old also exhibit the McGurk effect [Rosenblum et al. 1997] and they recognise the correspondence between auditorially and visually presented speech sounds [Kuhl and Meltzoff 1982]. Furthermore, [Teinonen et al. 2008] demonstrates that infants

learn phonetic contrasts more effectively in auditory-visual conditions compared to auditory-only conditions. This suggests that the ability to integrate information from multiple senses is an innate and fundamental aspect of human cognitive development, highlighting the potential benefits of incorporating multimodal learning in artificial intelligence systems from the ground up.

The effectiveness of multimodal processing is further evidenced by its widespread application in educational settings. Audio-visual aids [Madhuri 2013], teaching tools or materials that combine both auditory and visual elements to enhance the learning experience, have long been recognised as powerful tools in the education system. They are proven to be effective in learning language [Mathew and Alidmat 2013; Wazeema and Kareema 2017] or science [Ho and Intai 2017; Ojelade et al. 2020] By engaging multiple senses simultaneously, these aids can improve comprehension, retention, and recall of information.

This thesis is inspired by these psychological and educational background. We study how to train deep learning models to leverage the visual and aural modalities as complementary to each other for tackling video understanding tasks.

Audio-visual deep learning Audio-visual deep learning has emerged as a powerful approach for understanding complex human behaviours and interactions. In the domain of human action analysis, combining audio and visual information has led to significant advancements. Action recognition, which involves identifying and classifying human activities in videos, has benefited by integrating audio cues such as object interactions and environmental sounds with visual data [Korbar et al. 2018; Kazakos et al. 2019; Gao et al. 2020; Nagrani et al. 2021; Nagrani et al. 2020c; Yunhua Zhang et al. 2022]. Similarly, action detection [Tian et al. 2018; Bagchi et al. 2021; Ramazanova et al. 2023] and anticipation [Zhong et al. 2023] tasks have seen improvements, allowing systems to not only locate actions in time and space but also predict future actions based on audio-visual context.

The analysis of human dialogue has also been revolutionised by audio-visual deep learning techniques. Speech recognition accuracy [Afouras et al. 2019a; Shi et al. 2022b; Gabeur et al. 2022; P. Ma et al. 2023] has improved significantly when visual cues such as lip movements are considered alongside audio input, especially in noisy environments. Moreover, speaker diarisation, the process of partitioning

an audio stream into homogeneous segments according to speaker identity, has been enhanced by incorporating visual information such as speakers' faces or lip movements [Ding et al. 2020; J. S. Chung et al. 2020c; E. Z. Xu et al. 2022; M.-K. He et al. 2022]. This multimodal approach has proven particularly effective in challenging scenarios with multiple speakers or background noise. Audio-visual models have also shown promise in speech separation tasks [Afouras et al. 2018b; Ephrat et al. 2018; Gao and Grauman 2021; Rahimi et al. 2022], where the goal is to isolate individual speakers from a mixture of voices, by leveraging both acoustic and visual information.

Beyond human-centric tasks, audio-visual deep learning has opened up new possibilities for comprehensive video understanding. Sounding object localisation [H. Zhao et al. 2018; Arandjelovic and Zisserman 2018; H. Chen et al. 2021; Xixi Hu et al. 2022], which aims to identify and locate the sources of sounds within a video frame, has become more precise by combining auditory features with spatial CNN feature maps from visual streams. Sound source separation [Gao et al. 2018; Gao and Grauman 2019; Tzinis et al. 2021; Majumder and Grauman 2022], which involves isolating individual sound sources from a mixture, has also benefited from audio-visual approaches, leveraging visual cues to enhance separation quality. The field has also seen advancements in cross-modal generation tasks, such as synthesising plausible audio for silent videos [Davis et al. 2014; K. Su et al. 2021; Du et al. 2023] or generating visual content based on audio input [L. Chen et al. 2017; J. S. Chung et al. 2017; Tang et al. 2018]. Additionally, audio-visual representation learning [Aytar et al. 2016; Korbar et al. 2024; Korbar et al. 2018; Owens and Efros 2018; Afouras et al. 2020b; Lian et al. 2023], which aims to learn a joint embedding space for audio and visual data has been widely studied and applied to various downstream applications such as source separation [Afouras et al. 2020b], object detection [Afouras et al. 2022] and action recognition [Korbar et al. 2018].

Learning from context in video Learning from temporal context in video analysis is a crucial aspect of developing more accurate and robust video models. Context provides additional information that can significantly enhance the understanding of individual actions or events within a video sequence. As illustrated in Fig. 1.1, the action "wash aubergine" can be inferred with higher certainty when we take into account the adjacent actions, such as "turn on tap" preceding and "turn

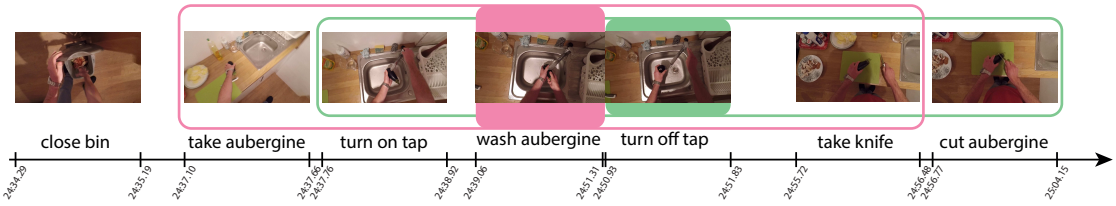


Figure 1.1: Demonstrating two temporal context windows (pink, green), centred around the action to be recognised. We can infer 'wash aubergine' with higher accuracy if we know that the tap was turned on before and turned off afterwards.

off tap" following. This temporal context allows the system to make more informed predictions by considering the logical sequence of actions in a given scenario.

The importance of contextual learning extends beyond simple action recognition. By analysing the relationships between different actions and objects, models can better interpret ambiguous situations, predict future events, and provide a more comprehensive understanding of the video content. In complex video understanding tasks, such as action anticipation [Furnari and Farinella 2019; Sener et al. 2020; Girdhar and Grauman 2021] or object detection [Beery et al. 2020; Bertasius and Torresani 2020], the ability to leverage contextual information becomes even more critical. This approach mimics human cognition, where we naturally use contextual clues to make sense of our surroundings and anticipate likely outcomes.

Audio-visual datasets The internet is flooded with video content in our current era. In 2024, there are 3.9 billion videos on Youtube and 6 hours of videos are uploaded every second [1]. Moreover, with the emergence of other video-based social media platforms like Instagram and TikTok, the internet has come to include even more video data. These data, which often include both audio and video components simultaneously, serve as a vital resource for creating audio-visual datasets. However, methodologies for annotating these video data have not yet been extensively researched. While many video datasets [Nagrani et al. 2017; Carreira and Zisserman 2017; Bain et al. 2019; Patraucean et al. 2024] are publicly available, most of these are collections of videos annotated with labels to solve specific problems, and even annotating these small amounts of labels requires enormous resources. In particular, audio-visual datasets require even more resources than typical video datasets because they involve transcribing *two* different modalities. This thesis investigates various methodologies for creating new datasets to address

[1] https://photutorial.com/how-many-videos-on-youtube/

desired problems such as speaker recognition, speaker diarisation, and audio event recognition, aiming to solve these resource-intensive issues.

1.2 Key Objectives

Audio-visual action recognition Action recognition is a fundamental task in computer vision and video understanding that aims to identify and classify human actions or activities within video sequences. In the context of untrimmed videos, given a video segment with a start and end time in a video, the task is to predict the action in this segment. The objective of audio-visual action recognition is to develop models that can effectively process both visual and auditory information to identify actions accurately.

This task becomes particularly complex when dealing with egocentric videos, where the camera moves with the wearer and captures a first-person perspective of their actions. Egocentric datasets like EPIC-KITCHENS [Damen et al. 2022] or Ego4D [Grauman et al. 2022] contain videos collected by having people wear portable cameras on their heads. These datasets typically include videos recorded over long periods, often exceeding an hour, rather than short clips of around 10 seconds. Both the video and audio inputs are clear, as they are recorded in close proximity to the camera wearer's actions. These characteristics make such datasets excellent benchmarks for evaluating audio-visual action recognition models in untrimmed videos.

Character-aware subtitle generation for TV shows Subtitle generation for TV shows involves automatically creating accurate, time-synchronised text captions for the spoken content in video broadcasts. This task goes beyond simple speech recognition, requiring not only accurate transcription but also precise timing information for each utterance.

Character-aware subtitle generation aims to assign the specific character names per each utterances. The objective is to develop models that can not only transcribe speech accurately but also attribute dialogue to the correct speakers and align subtitles precisely with the audio. Fig. 1.2 shows an example of character-aware subtitle. This research has practical applications in generating Subtitles for Deaf

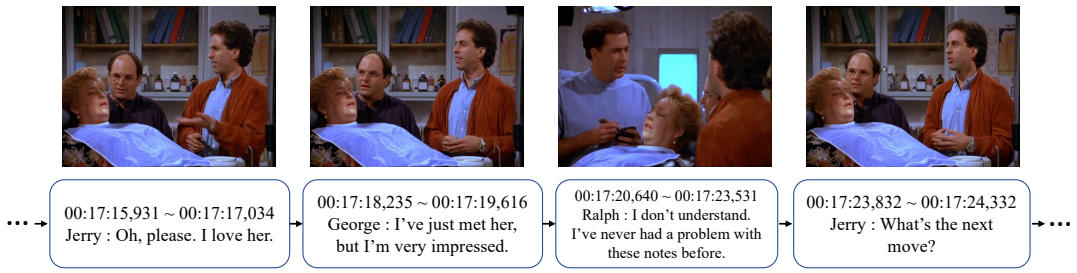


Figure 1.2: The example of character-aware subtitle. It contains speech transcripts with timestamps, and assigns the character who spoke it.

and Hard-of-hearing (SDH) [Szarkowska 2020] and an in-depth understanding of dialogue content in TV shows.

Efficient audio-visual dataset curation The curation of high-quality audio-visual datasets is a critical foundation for advancing research in multimodal machine learning. This process involves collecting, organising, and annotating large volumes of audio-visual data to create resources that can be used to train and evaluate AI models. The objective is to develop efficient methods for assembling datasets that are not only large-scale but also diverse, accurately labeled, and representative of real-world scenarios.

1.3 Thesis Outline and Contributions

This section provides an outline of the rest of the thesis.

Part I. Audio-visual action recognition in egocentric videos In Chapter 2, we propose a transformer-based multimodal model that ingests video and audio as input modalities to recognise human actions in egocentric videos. We explore how to leverage the action's temporal context and propose a method that learns to attend to surrounding actions, when their temporal boundaries are provided, in order to improve recognition performance. We also investigate utilising not only the action's temporal context in the data stream, but also the temporal context from the *labels* of neighbouring actions, with an explicit language model which provides action sequence context to enhance the predictions.

In Chapter 3, we extend the previous work and introduce audio-visual network which utilises temporal context *without* the additional information such as temporal boundaries of neighbouring actions. We propose the Time Interval Machine

(TIM) where a modality-specific time interval poses as a query to a transformer encoder that ingests a long video input. The encoder then attends to the specified interval, as well as the surrounding context in both modalities, in order to recognise the ongoing action. We test TIM on three long audio-visual video datasets: EPIC-KITCHENS [Damen et al. 2022], Perception Test [Patraucean et al. 2024], and AVE [Tian et al. 2018], reporting state-of-the-art for action recognition.

Part II. Audio-visual character-aware subtitle generation for TV shows This part focuses on automatic character-aware audio visual subtitle generation for TV shows. Given a video and a minimal amount of metadata (*e.g.* castlist per each video), our goal is to generate a full transcript of the dialogue, with precise speech timestamps, and the character speaking identified.

Chapter 4 introduces a method which exploits audio-visual cues to select a set of high-precision audio exemplars for each character, and then uses these exemplars to classify all speech segments by speaker identity. The method does not require face detection or tracking method. We evaluate our method's performance over a variety of TV shows, including Seinfeld, Frasier and Scrubs.

Chapter 5 presents an advanced framework from previous chapter for character-aware subtitling. This approach brings two significant enhancements to the table. It showcases how audio-visual synchronisation can be leveraged to pinpoint the speaking character amidst multiple faces in a video clip, thereby associating an identity with the corresponding speech segment. Additionally, the approach demonstrates that speaker identification for short segments can be achieved by examining the temporal context of dialogue within a scene. We utilise local voice embeddings extracted from the nearby audio, along with reasoning based on large language models applied to the text transcription. The method is tested on a dataset of 12 TV shows, thus demonstrating its ability to generalise across TV shows from a variety of genres.

Part III. Audio-visual dataset curation Chapter 6 proposes VoxConverse, a large-scale speaker diarisation dataset collected from 'in-the-wild' videos. We introduces an automatic audio-visual diarisation method which consists of active speaker detection, speech enhancement and audio-only speaker verification. We

then integrate this method into a semi-automatic dataset creation pipeline and create VoxConverse dataset. VoxConverse consists of 448 multi-speaker videos covering 3,830 minutes.

Chapter 7 proposes a novel, challenging speaker recognition dataset called VoxMovies. The chapter introduces an audio-visual data collection pipeline, adopted from the one used to in VoxCeleb [Nagrani et al. 2017] dataset. This modified pipeline is applied to an existing movie dataset [Bain et al. 2019] to collect speaker recognition dataset for 856 identities from around 4,000 movie clips. We also investigate how well state-of-the-art speaker recognition models which are typically trained on Youtube videos can adapt to recognising human speech from movies. Results demonstrate that these models can generalise to movies by applying simple domain adaptation techniques.

Chapter 8, we introduce Epic-Sounds, a large-scale dataset of audio annotations capturing temporal extents and class labels within the audio stream of the egocentric videos. We identify actions that can be discriminated purely from audio, through grouping these free-form descriptions of audio into classes. An annotation pipeline is proposed where annotators temporally segment the audio and describe the action that is happening in the audio. Epic-Sounds includes 78.4k categorised segments of audible events and actions, distributed across 44 classes as well as 39.2k non-categorised segments. We highlight the importance of audio-only labels and the limitations of current models to recognise actions that sound.

1.4 Publications

Chapter 2 to Chapter 8 are all peer-reviewed conference papers. We make no modifications to the content of these papers, except for the format to fit the style of the thesis. The corresponding statement of authorship for each publication can be found in Appendix A. The papers included in this thesis are:

($*$ indicates equal contribution)

- Chapter 2 – With a Little Help from my Temporal Context: Multimodal Egocentric Action Recognition

  Evangelos Kazakos, **Jaesung Huh**, Arsha Nagrani, Andrew Zisserman.

Published in the *British Machine Vision Conference*, 2021.

- Chapter 3 – TIM: A Time Interval Machine for Audio-Visual Action Recognition

  Jacob Chalk∗, **Jaesung Huh**∗, Evangelos Kazakos, Andrew Zisserman, Dima Damen.

  Published in the *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

- Chapter 4 – Look, Listen and Recognise: character-aware audio-visual subtitling

  Bruno Korbar∗, **Jaesung Huh**∗, Andrew Zisserman.

  Published in the *International Conference on Acoustics, Speech, and Signal Processing*, 2024.

- Chapter 5 – Character-aware audio-visual subtitling in context

  Jaesung Huh, Andrew Zisserman.

  Published in the *Asian Conference on Computer Vision*, 2024.

- Chapter 6 – Spot the conversation: speaker diarisation in the wild

  Joon Son Chung∗, **Jaesung Huh**∗, Arsha Nagrani∗, Triantafyllos Afouras, Andrew Zisserman.

  Published in the *Interspeech*, 2020.

- Chapter 7 – Playing a Part: Speaker Verification at the Movies

  Andrew Brown∗, **Jaesung Huh**∗, Arsha Nagrani∗, Joon Son Chung, Andrew Zisserman.

  Published in the *International Conference on Acoustics, Speech, and Signal Processing*, 2021.

- Chapter 8 – Epic-Sounds: A Large-scale Dataset of Actions that Sound

  Jaesung Huh∗, Jacob Chalk∗, Evangelos Kazakos, Dima Damen, Andrew Zisserman.

  Published in the *International Conference on Acoustics, Speech, and Signal Processing*, 2023.

Publications not included

- The VoxCeleb Speaker Recognition Challenge: A Retrospective

  Jaesung Huh, Joon Son Chung, Arsha Nagrani, Andrew Brown, Jee-weon

Jung, Daniel Garcia-Romero, Andrew Zisserman.

Published in the *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

- WhisperX: Time-Accurate Speech Transcription of Long-Form Audio

  Max Bain, **Jaesung Huh**, Tengda Han, Andrew Zisserman.

  Published in the *Interspeech*, 2023.

Part I

Audio-visual action recognition in egocentric videos

Chapter 2

With a Little Help from my Temporal Context: Multimodal Egocentric Action Recognition

The paper has been accepted for publication at the British Machine Vision Conference (BMVC), 2021.

With a Little Help from my Temporal Context: Multimodal Egocentric Action Recognition

Evangelos Kazakos[1] Jaesung Huh[2] Arsha Nagrani[2]
Andrew Zisserman[2] Dima Damen[1]
[1]University of Bristol [2]VGG, University of Oxford

Abstract

In egocentric videos, actions occur in quick succession. We capitalise on the action's temporal context and propose a method that learns to attend to surrounding actions in order to improve recognition performance. To incorporate the temporal context, we propose a transformer-based multimodal model that ingests video and audio as input modalities, with an explicit language model providing action sequence context to enhance the predictions. We test our approach on EPIC-KITCHENS and EGTEA datasets reporting state-of-the-art performance. Our ablations showcase the advantage of utilising temporal context as well as incorporating audio input modality and language model to rescore predictions. Code and models at: https://github.com/ekazakos/MTCN.

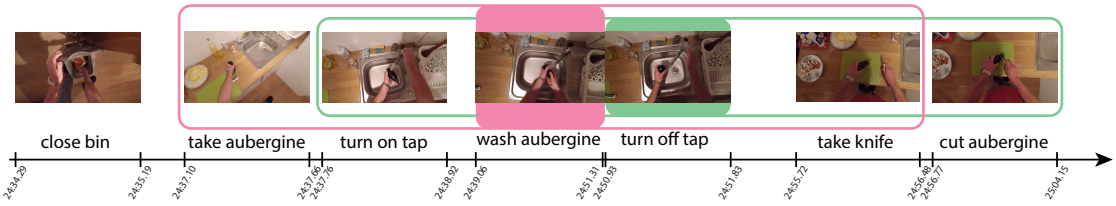


Figure 2.1: Egocentric video demonstrating two temporal context windows (pink, green), centred around the action to be recognised. We can infer 'wash aubergine' with higher accuracy if we know that the tap was turned on before and turned off afterwards.

2.1 Introduction

Action recognition in egocentric video streams from sources like EPIC-KITCHENS poses a number of challenges that differ substantially from those of conventional third-person action recognition – where training and evaluation is on 10 second video clips and classes are quite high-level [Kay et al. 2017]. Actions are fine-grained (e.g. 'open bottle') and noticeably short, often one second or shorter. Along with the challenge, the footage offers an under-explored opportunity, as actions are captured in long untrimmed videos of well-defined and at-times predictable sequences. For example the action 'wash aubergine' can be part of the following sequence – you first 'take the aubergine', 'turn on the tap', 'wash the aubergine' and finally 'turn off the tap' (Fig. 2.1). Furthermore, the objects (the aubergine and tap in this case) are persistent over some of the neighbouring actions.

In this work, we investigate utilising not only the action's temporal context in the data stream, but also the temporal context from the labels of neighbouring actions. We propose a model that attends to neighbouring actions[1]. Concretely, we use the attention mechanism of a multimodal transformer architecture to take account of the context in both the data and labels, using three modalities: vision, audio and language. We are motivated by previous works that demonstrate the significance of audio in recognising egocentric actions [Kazakos et al. 2019; W. Wang et al. 2020; Xiao et al. 2020], and thus include the auditory temporal context in addition to the visual clips. We also utilise context further, by training a language model on the sequence of action labels, inspired by the success of language models [Devlin et al. 2019; Zhilin Yang et al. 2019; T. Brown et al. 2020] in re-scoring model outputs for speech recognition [Chan et al. 2016; Mikolov et al. 2010; Hannun et al. 2014]

[1]Note that these action start/end times are readily available in labelled datasets for untrimmed videos and do not require additional labels. We just leverage these.

and machine translation [Gulcehre et al. 2017] (Fig. 2.2).

Our main contributions are summarised as follows: First, we formulate temporal context as a sequence of actions surrounding the action in a sliding window. Second, we propose a novel framework able to model multimodal temporal context. It consists of a transformer encoder that uses vision and audio as input context, and a language model as output context operating on the action labels. Third, we obtain state-of-the-art performance on two datasets: (i) the large-scale egocentric dataset EPIC-KITCHENS, outperforming high-capacity end-to-end transformer models; and (ii) the EGTEA dataset. Finally, we include an ablation study analysing the importance of the extent of the temporal context and of the various modalities.

2.2 Related Work

Action Recognition. There is a rich literature of seminal works in action recognition innovating temporal sampling [Limin Wang et al. 2016; B. Zhou et al. 2018; Kazakos et al. 2019], multiple streams [Feichtenhofer et al. 2019], spatio-temporal modelling [Tran et al. 2018; J. Lin et al. 2019; Girdhar et al. 2019] or modelling actions as transformations from initial to final states [X. Wang et al. 2016]. Our work is related to the more recent transformer-based approaches [Girdhar et al. 2019; Neimark et al. 2021; Bertasius et al. 2021; Arnab et al. 2021; Bulat et al. 2021; Patrick et al. 2021; Nagrani et al. 2021]. We compare our model with four recent works [Arnab et al. 2021; Bulat et al. 2021; Patrick et al. 2021; Nagrani et al. 2021]. [Arnab et al. 2021] investigated spatio-temporal attention factorisation schemes, while in [Bulat et al. 2021] the authors propose full-attention within a temporal window. [Patrick et al. 2021] proposes temporal attention along trajectories with learnt tokens. [Nagrani et al. 2021] proposes fusion bottlenecks for cross-modal attention. These approaches require training on large-scale datasets and strong data augmentation for generalisation. Our model operates on pre-extracted features which are processed with a lightweight transformer and outperforms these works by relying on multimodal temporal context.

In egocentric action recognition, researchers propose a range of techniques to address its unique challenges [Sudhakaran et al. 2019; Baradel et al. 2018; Yanghao Li et al. 2021; Kazakos et al. 2019; Munro and Damen 2020]. [Sudhakaran et al. 2019]

considered long-term understanding through an LSTM with attention to focus on relevant spatio-temporal regions, but the approach operates within the action clip solely. [Baradel et al. 2018] shows that modelling hand-object interactions is beneficial, while in [Yanghao Li et al. 2021] the authors pre-train egocentric networks by distilling egocentric signals from large-scale third-person datasets. [Kazakos et al. 2019] shows that audio is key to egocentric action recognition due to sounds produced from interactions with objects. In [Munro and Damen 2020], the authors propose a multimodal unsupervised domain adaptation approach to tackle the distribution shift between environments. We build on these works but propose the first approach to incorporate both audio-visual and language-model predictions.

Temporal Context for Video Recognition Tasks. A few works have considered temporal context to improve the models' performance for action anticipation [Sener et al. 2020; Furnari and Farinella 2019], action recognition [C.-Y. Wu et al. 2019; C. Zhang et al. 2021; Ng and Fernando 2019; Cartas et al. 2021], and object detection [Beery et al. 2020; Bertasius and Torresani 2020]. [Sener et al. 2020] proposes a model that operates on multi-scale past temporal context for action anticipation while they also test on action recognition by modifying the architecture to consider both past and future context. In [C. Zhang et al. 2021], a set of learnable query vectors attends to dense temporal context to identify events in untrimmed videos. In [Ng and Fernando 2019], an encoder-decoder LSTM is proposed for classifying sequences of human actions, effectively attending to the relevant temporal context of each action. Closest to our work is [C.-Y. Wu et al. 2019], where long-term features from the past and the future of an untrimmed video are utilised, to improve the recognition performance of the ongoing action. A key difference is that we exploit both the temporal bounds and predicted labels of neighbouring actions.

Multimodal transformers. The self attention mechanism of transformers provides a natural bridge to connect multimodal signals. Applications include audio enhancement [Ephrat et al. 2018; Tzinis et al. 2021], speech recognition [Harwath et al. 2016], image segmentation [Linwei Ye et al. 2019; Tzinis et al. 2021], cross-modal sequence generation [Gan et al. 2020; R. Li et al. 2021; Jiaman Li et al. 2020], video retrieval [Gabeur et al. 2020] and image/video captioning/classification [J. Lu et al. 2019; C. Sun et al. 2019b; C. Sun et al. 2019a; G. Li et al. 2019;

Iashin and Rahtu 2020; Jaegle et al. 2021]. A common paradigm (which we also adapt) is to use the output representations of single modality convolutional networks as inputs to the transformer [Lee et al. 2021; Gabeur et al. 2020]. Unlike these works, we use transformers to combine modalities in two specific ways – we first combine audio and visual inputs to predict actions based on neighbouring context, and then *re-score* these predictions with the help of a language model applied on the outputs.

Language models for action detection. There are a few works that incorporate a language model in action detection. The most relevant work is [Richard and Gall 2016], which utilises a statistical n-gram language model along with a length model and an action classifier. [M. Lin et al. 2017] combines the Connectionist Temporal Classification (CTC) [Graves et al. 2006] with a language model to learn relationships between actions. Both of these works improve the results by considering the contextual structure of the sequence of actions, albeit relying on a statistical language model, whereas in this work we utilise a neural language model which has shown better performance [Y. Bengio et al. 2003; Doval and Gómez-Rodríguez 2019; Y. Kim et al. 2016].

2.3 Multimodal Temporal Context Network (MTCN)

Given a long video, we predict the action in a video segment by leveraging the *temporal context* around it. We define the temporal context as the sequence of neighbouring actions that precede and succeed the action, and aim to leverage that information, when useful, through learnt attention. We utilise multimodal temporal context both at the input and the output of our model. An audio-visual transformer ingests a temporally-ordered sequence of visual inputs, along with the corresponding sequence of auditory inputs. We use modality-independent positional encodings as well as modality-specific encodings. The language model, acting on the output of the transformer learns the prior temporal context of actions, i.e. the probability of the sequence of actions, using a learnt text embedding space.

Inspired by similar approaches [Wray et al. 2019], and instead of using a single summary embedding as in prior works for image [Dosovitskiy et al. 2021] and action classification [Arnab et al. 2021], the audio-visual model utilises two separate

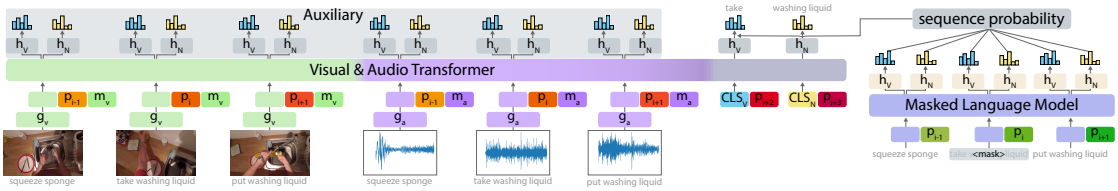


Figure 2.2: The Multimodal Temporal Context Network (MTCN). Visual and auditory tokens are tagged with positional and modality encodings. An audio-visual transformer encoder attends to the sequence. Verb and noun summary embeddings with independent positional encodings, predict the action at the centre of the window ('take washing liquid'). The classifier also predicts the sequence of actions to train an auxiliary loss that enhances the prediction of the centre action. The language model filters out improbable sequences.

summary embeddings to attend to the action (i.e. verb) class and the object (i.e. noun) class. This allows the model to attend independently to the temporal context of verbs vs objects. For example, the object is likely to be the same in neighbouring actions while the possible sequences of verbs can be independent of objects (e.g. 'take' → 'put'). Each summary embedding uses a different learnt classification token, and the classifier predicts a verb and a noun from the summary embeddings. The predictions of the audio-visual transformer are then enhanced by filtering out improbable sequences using the language model. We term the proposed model Multimodal Temporal Context Network (MTCN).

In the next three subsections, we detail the architectural components of MTCN, as well as our training strategy. An overview of MTCN is depicted in Fig. 2.2.

2.3.1 Audio-Visual Transformer

Let $X_v \in \mathbb{R}^{w \times d_v}$ be the sequence of visual inputs from a video, and $X_a \in \mathbb{R}^{w \times d_a}$ the corresponding audio inputs[2], for w consecutive actions in the video (i.e. the temporal context window), with d_v, d_a being the input dimensions of the two modalities respectively. X_v and X_a correspond to features extracted from visual and auditory networks, respectively. Our temporal window is centred around an action b_i with surrounding action segments, excluding any background frames. That is, each action b_j within the window, $i - \frac{w-1}{2} \leq j \leq i + \frac{w-1}{2}$ is part of the transformer's input.

Encoding layer. Our model first projects the inputs X_v, X_a to a lower dimension

[2]Video and audio inputs/features are synchronised, therefore they have the same length w.

D and tags each with positional and modality encodings. Then, an audio-visual encoder performs self-attention on the sequence to aggregate relevant audio-visual temporal context from neighbouring actions. Because all self-attention operations in a transformer are permutation invariant, we use positional encodings to retain information about the ordering of actions in the sequence. We use w learnt absolute positional encodings, shared between audio-visual features to model corresponding inputs from the two modalities. Modality encodings, m_v, $m_a \in \mathbb{R}^D$, are learnt vectors added to discriminate between audio and visual tokens.

A classifier predicts the action b_i, using two summary embeddings, acting on the learnt verb/noun tokens. We use the standard approach of appending learnable classification tokens to the end of the sequence but use two tokens, one for verbs and one for nouns, denoted as CLS_V, $\text{CLS}_\text{N} \in \mathbb{R}^D$, with unique positional encodings. To summarise, the encoding layer transforms the inputs X_v and X_a as follows:

$$X^e_{v_j} = g_v(X_{v_j}) + p_j + m_v \qquad X^e_{a_j} = g_a(X_{a_j}) + p_j + m_a \qquad \forall j \in [1, ..., w] \tag{2.1}$$

$$\text{CLS}^e_\text{V} = \text{CLS}_\text{V} + p_{w+1} \qquad \text{CLS}^e_\text{N} = \text{CLS}_\text{N} + p_{w+2} \qquad X^e = [X^e_v; X^e_a; \text{CLS}^e_\text{V}; \text{CLS}^e_\text{N}], \tag{2.2}$$

where $[;]$ denotes input concatenation and $p \in \mathbb{R}^{(w+2)\times D}$ are the positional encodings. $g_v(\cdot) : \mathbb{R}^{d_v} \mapsto \mathbb{R}^D$ and $g_a(\cdot) : \mathbb{R}^{d_a} \mapsto \mathbb{R}^D$, are fully-connected layers projecting the visual and audio features, respectively, to a lower dimension D. The input to the transformer is $X^e \in \mathbb{R}^{(2w+2)\times D}$. In Section 2.6.5, we compare the absolute positional encodings with relative [Shaw et al. 2018] and Fourier feature positional encodings [Jaegle et al. 2021].

Transformer and classifier. We use a transformer encoder $f(\cdot)$ to process sequential audio-visual inputs, $Z = f(X^e)$. We share the weights of the transformer encoder layer-wise. In Section 2.6.5, we compare this to a version without weight sharing. Weight sharing uses 2.7× less parameters with comparable results. A two-head classifier $h(\cdot)$ for verbs and nouns then predicts the sequence of w actions from both the transformed visual and audio tokens $\hat{Y} = h(Z_{1:2w})$, and the action b_i from the summary embeddings $\hat{y} = h(Z_{2w:2w+2})$.

Loss function. Recall that our goal is to classify b_i, the action localised at the

centre of our temporal context. Nevertheless, we can leverage the ground-truth of neighbouring actions within w for additional supervision to train the audio-visual transformer. Our loss is composed of two terms, the main loss for training the model to classify the action at the centre of our temporal context $i = \frac{w}{2}$, and an auxiliary loss to predict all actions in the sequence:

$$L_{\mathrm{m}} = CE(Y_i^{\mathrm{V}}, \hat{y}^{\mathrm{V}}) + CE(Y_i^{\mathrm{N}}, \hat{y}^{\mathrm{N}}) \tag{2.3}$$

$$L_{\mathrm{a}} = \sum_{j=1}^{w} \left(CE(Y_j^{\mathrm{V}}, \hat{Y}_j^{\mathrm{V}}) + CE(Y_j^{\mathrm{V}}, \hat{Y}_{|w|+j}^{\mathrm{V}}) + CE(Y_j^{\mathrm{N}}, \hat{Y}_j^{\mathrm{N}}) + CE(Y_j^{\mathrm{N}}, \hat{Y}_{|w|+j}^{\mathrm{N}}) \right) \tag{2.4}$$

$$loss = \beta L_{\mathrm{m}} + (1 - \beta) L_{\mathrm{a}}, \tag{2.5}$$

where $CE()$ is a cross-entropy loss, and $Y = (Y_1, ..., Y_w)$ is the ground-truth of the sequence, while $\hat{Y}_1, ..., \hat{Y}_w$ and $\hat{Y}_{w+1}, ..., \hat{Y}_{2w}$ correspond to predictions from the transformed visual and auditory inputs respectively. We use β to weight the importance of the auxiliary loss.

2.3.2 Language Model

In addition to input context from the visual and audio domains, we introduce output context using a language model. Language modelling, commonly applied to predict the probability of a sequence of *words*, is a fundamental task in NLP research [Peters et al. 2018; Devlin et al. 2019; T. Brown et al. 2020; Zhilin Yang et al. 2019]. Our language model predicts the probability of a sequence of *actions*. We use the language model to improve the predictions of the audio-visual transformer by filtering out improbable sequences.

We adopt the popular Masked Language Model (MLM), introduced in BERT [Devlin et al. 2019]. We train this model independently from the audio-visual transformer. Specifically, given a sequence of actions $Y = (Y_{i-\frac{w-1}{2}}, ..., Y_{i+\frac{w-1}{2}})$, we randomly mask any action Y_j and train the model to predict it. For example, an input sequence to the model (for $w = 5$) would be: ('turn on tap', 'wash hands', <MASK>, 'pick up towel', 'dry hands'). Without any visual or audio input, the language model is tasked to learn a high prior probability for 'turn off tap', which

is masked. Note that the model is trained using the ground-truth sequence of actions. For input representation, we split the action into verb and noun tokens (e.g. 'dry hands' → 'dry' and 'hand'), convert them to one-hot vectors and input them into separate word-embedding layers[3]. MLM takes as input the concatenation of verb and noun embeddings. The outputs are scores for verb and noun classes using a two-head classifier, and the model is trained with a cross-entropy loss per output.

2.3.3 Inference

Given the the scores of the sequence, $\hat{S} = (\hat{y}_{i-\frac{w-1}{2}}, ..., \hat{y}_{i+\frac{w-1}{2}})$, from the audio-visual model[4], we apply a beam-search of size K to find the K most probable action sequences $\hat{S}_b$. Therefore, $\hat{S}_b$ is of size $K \times w$. In inference, the trained language model takes as input $\hat{S}_b$, i.e. it operates on sequences predicted from the audio-visual transformer.

For each sequence l, we calculate the probability of the sequence $p_{LM}(\hat{S}_b^l)$ from the language model by utilising the method introduced in [Shin et al. 2019]. We mask actions, one at a time, and predict their probability. $p_{LM}(\hat{S}_b^l)$ is the sum of log probabilities of all actions in l. We also calculate the probability $p_{AV}(\hat{S}_b^l)$ by summing the log probabilities of all predicted actions in l by the audio-visual model. Then, we combine the probabilities of sequences of the audio-visual and language models:

$$p(\hat{S}_b^l) = \lambda p_{LM}(\hat{S}_b^l) + (1 - \lambda) p_{AV}(\hat{S}_b^l). \tag{2.6}$$

Sequences are then sorted in descending order by $p(\hat{S}_b^l)$. The score of the centre action from the sequence with the highest probability, is used as the final prediction.

[3]Learning a word embedding outperforms pretrained embeddings.

[4]These are temporally ordered predictions from the summary embeddings, and thus different from $\hat{Y}$.

2.4 Experiments

2.4.1 Implementation Details

Datasets. EPIC-KITCHENS-100 [Damen et al. 2022] is the largest egocentric audio-visual dataset, containing unscripted daily activities, thus offering naturally variable sequences of actions. There are on average 129 actions per video (std 163 actions/video and maximum of 940 actions/video). This makes the dataset ideal for exploring temporal context. The length of sequences of $w = 9$ actions (this is our default window length) is 34.4 seconds of video on average with an std of 27.8 seconds[5]. EGTEA [Yingwei Li et al. 2018] is another video-only egocentric dataset. There are 28 hours of untrimmed cooking activities with 10K annotated action segments. Although the dataset does not contain audio, it has sequential actions annotated within long videos, and we use it to train part of our approach (vision and language).

Visual features. For EPIC-KITCHENS, we extract visual features with SlowFast [Feichtenhofer et al. 2019], using the public model and code from [Damen et al. 2022]. We first train the model with slightly different hyperparameters, where we sample a clip of 2s from an action segment, do not freeze batch normalisation layers, and warm-up training during the first epoch starting from a learning rate of 0.001. We note that this gave us better performance. All unspecified hyperparameters remain unchanged. For feature extraction, 10 clips of 1s each are uniformly sampled for each action segment, with a center crop per clip. The resulting features have a dimensionality of $d_v = 2304$. The SlowFast visual features are used for all the results in this paper, apart for the comparison with the state of the art in Table 2.3 where we additionally experiment with features from Mformer-HR [Patrick et al. 2021]. These are extracted from the EPIC-KITCHENS pretrained model using a single crop per clip. The resulting features have a dimensionality of d_v=768. For EGTEA, see Section 2.6.6.

Auditory features. We use Auditory SlowFast [Kazakos et al. 2021b] for audio feature extraction when present. Similarly to the visual features, we extract 10 clips of 1s each uniformly spaced for each action segment, with average pooling and concatenation of the features from the Slow and Fast streams, and the resulting

[5]minimum of 3.4 seconds to a maximum of 720.2 seconds.

features have the same dimensionality, $d_a = 2304$.

Architectural details. Both the audio-visual transformer encoder and the language model consist of 4 layers with shared weights, 8 attention heads and a hidden unit dimension of 512. In the audio-visual transformer, positional/modality encodings as well as verb/noun tokens have also dimensionality $D = 512$ and are initialised to $\mathcal{N}(0, 0.001)$. The layers $g_v(\cdot)$ and $g_a(\cdot)$ reduce the features to the common dimension $D = 512$. In the encoding layer, dropout is applied at the inputs of $g_v(\cdot)$ and $g_a(\cdot)$ as well as at X^e. In the language model, both verb and noun word-embedding layers have a dimension of 256, and positional encodings have a dimension of 512, while dropout is applied to its inputs.

Scheduled sampling. We modify the scheduled sampling from [S. Bengio et al. 2015] to train the language model. At each training iteration, we randomly mask an action, predict it, and replace the corresponding ground-truth with the prediction.

Train / Val details. For EPIC-KITCHENS, the audio-visual transformer is trained using SGD, a batch size of 32 and a learning rate of 0.01 for 100 epochs. Learning rate is decayed by a factor of 0.1 at epochs 50 and 75. In the loss function, we set $\beta = 0.9$. For regularisation, a weight decay of 0.0005 is used and a dropout 0.5 and 0.1 for the encoding layers and transformer layers respectively. We use mixup data augmentation [Hongyi Zhang et al. 2018] with $\alpha = 0.2$. The language model is trained for the same number of epochs with a batch size of 64, Adam optimiser with initial learning rate of 0.001 and the learning rate is decreased by a factor of 0.1 when validation accuracy saturates for over 10 epochs. The values of dropout and weight decay are the same as those of the audio-visual model. For inference, we tune λ in Eq. (2.6) on the validation set with grid-search from the set $\{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$, and we use a beam size of $K = 10$. For training the audio-visual transformer, we randomly sample 1 out of 10 features per action. For testing, we feed all 10 features per action to the transformer and we share the positional encoding corresponding to an action with all 10 features. We also tried single feature per action followed by averaging 10 predictions but observed no difference in performance. For the train/val details in EGTEA, please see Section 2.6.6.

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
w	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
1	67.93	52.29	41.30	90.53	76.47	61.52	61.13	44.60	32.58	42.05	27.42	21.48
3	69.80	55.24	43.52	91.30	79.04	63.25	61.41	46.48	33.71	39.09	32.58	23.06
5	70.38	56.16	45.13	**91.67**	**79.47**	**64.14**	61.97	46.95	34.74	**43.12**	32.53	24.54
7	70.43	56.19	45.01	91.23	79.13	63.52	62.63	**47.14**	34.84	41.31	32.79	24.12
9	**70.60**	**56.26**	**45.48**	91.14	79.06	63.06	**63.76**	**47.14**	**35.87**	41.36	32.84	**24.70**
11	70.55	55.74	44.68	91.18	79.23	63.02	62.91	46.57	34.74	41.82	**33.58**	24.44

Table 2.1: Analysis of temporal context extent for MTCN on EPIC-KITCHENS.

Evaluation metrics. For EPIC-KITCHENS-100, we follow [Damen et al. 2022] and report top-1 and top-5 accuracy for the validation and test sets separately (Test results and results on EPIC-KITCHENS-55 in Section 2.6.1 and Section 2.6.2). We also follow [Damen et al. 2022] and report results for two subsets within val/test: unseen participants and tail classes. For EGTEA, we follow [Min and Corso 2021; M. Lu et al. 2019; Kapidis et al. 2019] and report top-1 accuracy and mean class accuracy using the first train/test split.

2.4.2 Analysis of temporal context length

We first analyse the importance of the temporal context extent by varying the size of w, i.e. the length of window of actions that our model observes. We use the validation set of EPIC-KITCHENS for this analysis as well as for the ablations in Section 2.4.3, and report EGTEA analysis in Section 2.6.7. We perform this analysis both for MTCN containing all modalities as shown in Table 2.1, as well as for the language model as shown in Table 2.2. Varying the length of the window has a big impact on the model's accuracy showcasing that MTCN successfully utilises temporal context. Using temporal context outperforms $w = 1$, i.e. no temporal context. As the window length increases the performance also increases. Overall top-1 accuracy increases up to $w = 9$ while top-5 up to a window $w = 5$. Performance on unseen participants and tail classes also increases up to $w = 9$.

In Table 2.2, experiments are conducted by masking the centre action and measuring how well the model predicts it. We use ground-truth for the other actions, as in this experiment we are interested in the maximum possible performance of the language model, i.e. assuming correct predictions from the audio-visual model. Here $w = 1$ corresponds to a language model that randomly guesses the masked

	Overall		
	Top-1 Accuracy (%)		
w	Verb	Noun	Action
1	19.32	3.74	0.82
3	38.08	45.56	23.85
5	42.15	**50.36**	29.48
7	42.93	50.35	**29.91**
9	**43.06**	50.22	29.41
11	41.89	49.96	29.14

Table 2.2: Analysis of the temporal context on LM.

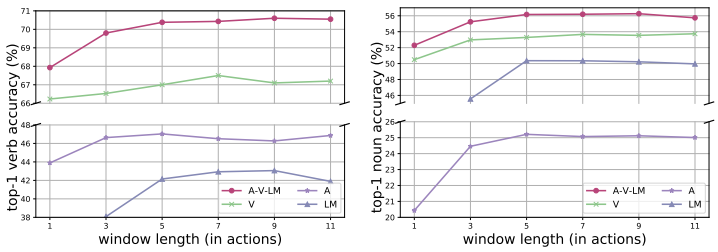


Figure 2.3: Effect of temporal context on verb (left) and noun (right) accuracy of individual modalities and for MTCN on EPIC-KITCHENS. Y-axis cut to emphasise details.

action without any context. When varying the length of the window for the language model, verb performance increases when enlarging the temporal context from $w = 3$ to $w = 9$ while for nouns optimal temporal context is $w = 5$, and $w = 7$ for actions. Interestingly, the language model is performing particularly well for nouns, since the same object is often used for the entire sequence.

Finally, Fig. 2.3 shows the effect of temporal context extent on verb and noun accuracy of the individual modalities as well as of MTCN. For verbs, visual modality performance increases up to 7 actions and then decreases while for nouns, it steadily increases up to 11 actions, possibly because larger context is able to resolve ambiguities due to occlusion. Audio can better capture temporal context for verbs than the language model, showcasing that the progression of sounds conveys more useful information about the action. For nouns, the language model outperforms audio. Moreover, the language model performs better on nouns because of repetitions of the same object in the sequence. MTCN utilising all modalities (A-V-LM) benefits the most from larger temporal context, particularly for verbs. With this analysis completed, we fix w to 9 in all subsequent experiments.

2.4.3 Results and Ablations

We compare our approach with the state-of-the-art (SOTA) approaches on EPIC-KITCHENS-100 as shown in Table 2.3. MTCN significantly outperforms convolutional approaches [Limin Wang et al. 2016; J. Lin et al. 2019; Kazakos et al. 2019; Feichtenhofer et al. 2019]. We outperform the audio-visual TBN [Kazakos et al. 2019] by 8% on top-1 actions, and SlowFast [Feichtenhofer et al. 2019] by 6% (using the same visual features). We also outperform very recent transformer-

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Model	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
TSN [Limin Wang et al. 2016]	60.2	46.0	33.2	89.6	72.9	55.1	47.4	38.0	23.5	30.5	19.4	13.9
TBN [Kazakos et al. 2019]	66.0	47.2	36.7	90.5	73.8	57.7	59.4	38.2	29.5	39.1	24.8	19.1
TSM [J. Lin et al. 2019]	67.9	49.0	38.3	91.0	75.0	60.4	58.7	39.6	29.5	36.6	23.4	17.6
SlowFast [Feichtenhofer et al. 2019]	65.6	50.0	38.5	90.0	75.6	58.6	56.4	41.5	29.7	36.2	23.3	18.8
ViViT-L/16x2 [Arnab et al. 2021]	66.4	56.8	44.0	-	-	-	-	-	-	-	-	-
X-ViT (16x) [Bulat et al. 2021]	68.7	56.4	44.3	-	-	-	-	-	-	-	-	-
Mformer-HR [Patrick et al. 2021]	67.0	58.5	44.5	-	-	-	-	-	-	-	-	-
MBT [Nagrani et al. 2021]	64.8	58.0	43.4	-	-	-	-	-	-	-	-	-
MTCN - v.f. SlowFast [Feichtenhofer et al. 2019]	70.6	56.3	45.5	**91.1**	79.1	63.1	**63.8**	47.1	35.9	41.4	32.8	24.7
MTCN - v.f. Mformer-HR [Patrick et al. 2021]	**70.7**	**62.1**	**49.6**	90.7	**83.1**	**68.6**	63.7	**50.9**	**38.9**	**41.9**	**39.2**	**27.7**

Table 2.3: Comparison with SOTA on EPIC-KITCHENS-100 using two visual features ('v.f.')

				Overall						Unseen Participants			Tail-classes		
				Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
V	A	LM	Aux	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
✓	✗	✗	✓	67.10	53.54	41.49	90.62	78.22	62.32	58.87	43.29	30.99	40.97	30.47	22.35
✓	✗	✓	✓	67.84	54.08	42.05	90.63	78.20	60.68	59.06	44.23	31.36	39.77	31.32	22.38
✓	✓	✗	✓	70.23	55.82	45.00	91.13	79.06	**64.58**	63.29	46.38	35.02	**41.76**	32.26	24.41
✓	✓	✓	✗	69.31	55.46	43.81	**91.19**	**79.76**	62.35	61.13	46.01	33.90	39.55	30.74	22.74
✓	✓	✓	✓	**70.60**	**56.26**	**45.48**	91.14	79.06	63.06	**63.76**	**47.14**	**35.87**	41.36	**32.84**	**24.70**
✓	✓	†	✓	71.33	63.56	50.32	92.05	83.16	67.85	62.25	52.68	37.56	41.53	44.16	29.89

Table 2.4: Ablation on multimodal temporal context and auxiliary loss in EPIC-KITCHENS. †: upper bound using ground-truth knowledge as input to the language model.

based approaches [Arnab et al. 2021; Bulat et al. 2021; Patrick et al. 2021; Nagrani et al. 2021], reporting published results (only top-1 accuracy). Note, that MTCN consists of a lightweight transformer that operates on pre-extracted features, while [Arnab et al. 2021; Bulat et al. 2021; Patrick et al. 2021; Nagrani et al. 2021] are high-capacity models trained end-to-end.

Additionally, when we employ visual features from Mformer-HR [Patrick et al. 2021], MTCN improves over all transformer-based approaches, including audio-visual fusion [Nagrani et al. 2021], by 3.5% on top-1 nouns and 5% on actions. We attribute the boost on nouns to the enhanced object recognition performance of the ViT backbone [Dosovitskiy et al. 2021] in Mformer-HR and its large-scale pretraining. These results, however, further demonstrate the potential to boost other methods and features by utilising multimodal temporal context.

We compare MTCN without audio with the state-of-the-art on EGTEA, in Table 2.5. This model uses $w = 3$. Our model improves over previous approaches by 3% in both top-1 and mean class accuracy. Note that MTCN outperforms SlowFast [Feichtenhofer et al. 2019] which we used to extract features. In Section 2.6.7,

Method	Top-1 (%)	MC(%)
Li et al. [Yin Li et al. 2018]	-	53.30
Ego-RNN [Sudhakaran and Lanz 2018]	62.17	-
Kapidis et al. [Kapidis et al. 2019]	68.99	61.40
Lu et al. [M. Lu et al. 2019]	68.60	60.54
SlowFast [Feichtenhofer et al. 2019]	70.43	61.92
MCN [Y. Huang et al. 2020]	55.63	-
Min et al. [Min and Corso 2021]	69.58	62.84
MTCN (V) (Ours)	72.55	64.86
MTCN (V+LM) (Ours)	**73.59**	**65.87**

Table 2.5: Comparative results on EGTEA. MC: Mean Class

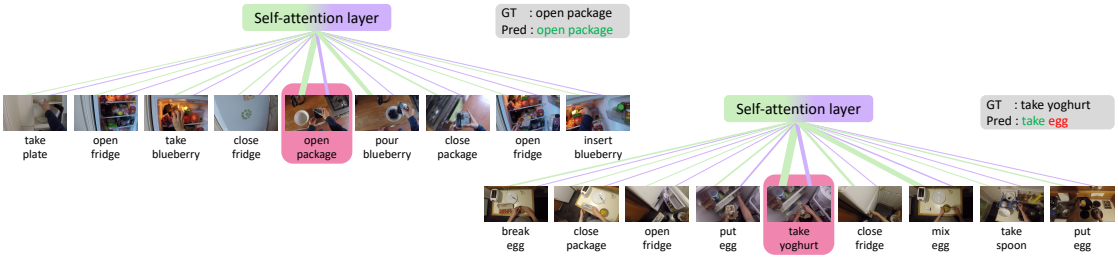


Figure 2.4: Qualitative results of attention weights along with the predictions of our model in EPIC-KITCHENS. Green and purple edges represent attention from the noun embedding to visual and auditory tokens, respectively. Thickness indicates attention weights magnitude to centre (bordered) and temporal context actions.

we ablate the temporal context and language model, where we showcase that the language model provides a bigger boost in performance, possibly due to the absence of the audio modality (also shown in Table 2.5: 'V' vs 'V+LM').

Multimodality Ablation. We offer an ablation to identify the performance impact of the components of our MTCN. Results are shown in Table 2.4. We remove audio and language modalities from our model's input and output respectively to assess their importance. Multimodal context is important, as our proposed model enjoys considerable margins compared with the model trained only with visual context (line 1 in Table 2.4). Audio is beneficial, confirming the finding of prior works.

Although the language model provides smaller boost in performance than audio, it showcases that it is useful to model prior temporal context at the output level, and that its benefits are complementary to audio. We also include upper bound performance improvement from the addition of the language model, where it takes as input the ground-truth preceding/succeeding actions rather than the predictions

from the audio-visual transformer (last line in Table 2.4), effectively a language infilling problem. These results demonstrate the potential margin for improvement, particularly for nouns and accordingly action accuracy as well as tail classes. In Section 2.6.3, we report statistical significance over multiple runs for the language model, and compare it to alternative baselines.

Auxiliary loss. Also from Table 2.4, training MTCN with auxiliary loss boosts its performance almost in all metrics, confirming that utilising supervision from neighbouring actions can improve the performance of the action of interest, i.e. the action at the centre of the window.

While we focus on offline action recognition, we can evaluate our model for online recognition by predicting the last action in the sequence instead of the centre one. Results are included in Section 2.6.4 for different values of w, where $w = 7$ provided best results for EPIC-KITCHENS. Best top-1 action performance using past context solely is 42.96% compared to 45.48% using surrounding context. However, we demonstrate that our model can leverage multimodal temporal context in this setting.

In Fig. 2.4, we visualise the auditory and visual temporal context attention on a correctly recognised sequence (left) – where subsequent actions of pouring and closing are particularly informative, and an incorrectly (right) predicted sequence – where both attention weights are high on actions containing the egg, causing the model to incorrectly predict the noun as egg. More qualitative examples are included in Section 2.6.8.

2.5 Conclusion

We formulate temporal context as a sequence of actions, and utilise past and future context to enhance the prediction of the centre action in the sequence. We propose MTCN, a model that attends to vision and audio as input modality context, and language as output modality context. We train MTCN with additional supervision from neighbouring actions. Our results showcase the importance of audio and language as additional modality context. We report SOTA results on two egocentric video datasets: EPIC-KITCHENS and EGTEA. An extension to MTCN would incorporate an actionness score of neighbouring frames, to distinguish background

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Model	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
TSN [Limin Wang et al. 2016]	59.03	46.78	33.57	87.55	72.10	53.89	53.11	42.02	27.37	26.23	14.73	11.43
TBN [Kazakos et al. 2019]	62.72	47.59	35.48	88.77	73.08	56.34	56.69	43.65	29.27	30.97	19.52	14.10
TSM [J. Lin et al. 2019]	65.32	47.80	37.39	89.16	73.95	57.89	59.68	42.51	30.61	30.03	16.96	13.45
SlowFast [Feichtenhofer et al. 2019]	63.79	48.55	36.81	88.84	74.49	56.39	57.66	42.55	29.27	29.65	17.11	13.45
Ego-Exo [Yanghao Li et al. 2021]	66.07	51.51	39.98	**89.39**	76.31	60.68	59.83	45.50	32.63	33.92	22.91	16.96
MTCN - v.f. SlowFast [Feichtenhofer et al. 2019]	**68.44**	55.41	44.10	88.74	78.04	61.69	**61.82**	47.62	34.94	34.77	28.60	20.45
MTCN - v.f. Mformer-HR [Patrick et al. 2021]	67.88	**60.02**	**46.83**	88.69	**81.84**	**66.48**	61.07	**55.21**	**38.98**	**35.16**	**34.70**	**22.79**

Table 2.6: Results on the test set of EPIC-Kitchens-100.

	Top-1 Accuracy (%)			Top-5 Accuracy (%)		
Model	Verb	Noun	Action	Verb	Noun	Action
LFB [C.-Y. Wu et al. 2019]	60.0	45.0	32.7	88.4	71.8	55.3
G-Blend [W. Wang et al. 2020]	66.7	48.5	37.1	88.9	71.4	56.2
AV-SlowFast [Xiao et al. 2020]	65.7	46.4	35.9	89.5	71.7	57.8
Ego-Exo [Yanghao Li et al. 2021]	65.97	47.99	37.09	**90.32**	70.72	56.32
MTCN (Ours)	**69.12**	**51.30**	**40.77**	90.18	**73.53**	**59.15**

Table 2.7: Comparison with SOTA on the Seen split (S1) of EPIC-KITCHENS-55.

frames and learn from action sequences in untrimmed videos without temporal bounds during testing. This would bridge the problems of recognition and detection, utilising multimodality and temporal context. We will be exploring this in future work.

2.6 Appendix

2.6.1 EPIC-KITCHENS-100: Results on the Test Set

In Table 2.3 of the main paper, we compare to published works on the validation set of EPIC-KITCHENS-100. Unfortunately, most works do not report on the leaderboard test set. In Table 2.6, we provide results on the test set comparing our model to baselines from [Damen et al. 2022], as well as Ego-Exo [Yanghao Li et al. 2021] that distills knowledge from a much larger training set. MTCN outperforms all other methods, including the competitive method of [Yanghao Li et al. 2021], showcasing that multimodal temporal context from consecutive actions is more beneficial than pretraining large models (ResNet101) using egocentric signals from third-person datasets.

2.6.2 EPIC-KITCHENS-55 Results

We also compare our model to works that report on the earlier version of this dataset, namely EPIC-KITCHENS-55 [Damen et al. 2018] in Table 2.7. We report

	Top-1 Accuracy (%)			Top-5 Accuracy (%)		
LM	Verb	Noun	Action	Verb	Noun	Action
✗	70.26 ± 0.27	55.70 ± 0.22	44.90 ± 0.20	91.12 ± 0.13	**79.03 ± 0.18**	**64.79 ± 0.17**
✓	**70.52 ± 0.25**	**56.08 ± 0.21**	**45.25 ± 0.18**	**91.13 ± 0.13**	**79.03 ± 0.18**	64.58 ± 0.18

Table 2.8: Mean and standard deviation of multiple runs both w. & w/o language model in the validation set of EPIC-KITCHENS-100.

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Model	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
No LM	70.23	55.82	45.00	91.13	**79.06**	**64.58**	63.29	46.38	35.02	**41.76**	32.26	24.41
N-gram	70.23	55.84	45.02	91.13	**79.06**	64.49	63.29	46.38	35.02	**41.76**	32.26	24.41
BiLSTM	70.57	55.97	45.09	**91.14**	**79.06**	**64.55**	63.29	46.76	35.31	40.68	32.47	24.15
Transformer enc. (proposed)	**70.60**	**56.26**	**45.48**	**91.14**	**79.06**	63.06	**63.76**	**47.14**	**35.87**	41.36	**32.84**	**24.70**

Table 2.9: Performance of MTCN in the validation set of EPIC-KITCHENS-100 using different language models.

results for the Seen split (S1). We opted to include these in the appendix to avoid confusion in the main paper as the results are not comparable across these two dataset versions. We compare MTCN with two audio-visual approaches [W. Wang et al. 2020] and [Xiao et al. 2020], as well as [C.-Y. Wu et al. 2019] which was one of the first works to utilise temporal context. We also report the performance of [Yanghao Li et al. 2021] which evaluates their method on both EPIC-KITCHENS-55 and EPIC-KITCHENS-100. Our MTCN outperforms all approaches.

2.6.3 Language model analysis and baselines

In this section, we assess the statistical significance of our language model and compare the performance of our MTCN to variants using baseline language models. All the experiments in this section as well as in Section 2.6.4 and Section 2.6.5 and the visualisations in Section 2.6.8 are performed on the validation set of EPIC-KITCHENS-100 and using the SlowFast visual features.

Statistical significance of LM. We train 10 audio-visual transformers and 10 corresponding language models with different random seeds. Table 2.8 shows the mean and standard deviation top-1 and top-5 accuracy without and with the language model. Utilising the language model improves performance on average with a low std, demonstrating that the improvement from the language model is statistically significant. We further showcase that by conducting T-tests on verb, noun and action top-1 accuracies, obtaining a p-value of $3.6e-2$, $6.0e-4$, $9.7e-4$, respectively.

Baselines comparison. We compare our MTCN that uses a transformer based language model to two baselines, N-gram and Bi-directional LSTM (BiLSTM). For N-gram, we follow a similar procedure to natural language processing. In particular, from all action sequences of length 9 in the training set, we derive the heuristic probability of occurrence of the centre action given the preceding and succeeding actions. We train a BiLSTM with 3 layers and a hidden size of 512. The rest hyperparameters are the same as the transformer encoder.

Results are shown in Table 2.9. It turns out that only a few preceding-succeeding action sequences in the training set also appear in the validation set, resulting in no difference in performance when N-gram is added comparing to not using a language model. Our transformer-based Masked Language Model (MLM) outperforms both the N-gram and BiLSTM, showcasing that it is beneficial to use a deep neural network language model over a heuristic prior and that MLM with transformers outperforms recurrent architectures in this problem.

2.6.4 Online recognition

The focus of this work is to leverage both past and future context to predict an action. In this section however, we explore the performance of our model in online recognition, i.e. using only the preceding actions as context to predict the current action. This approach can be used to recognise actions in an online fashion for streaming videos. For this setting, we train the audio-visual transformer to predict the last action in the sequence. We do not train a new language model for this task; we simply mask and predict the last action in the sequence instead of the centre one.

Results are demonstrated in Table 2.10 by varying w. Our model can also utilise temporal context in this setting, as performance improves for $w > 1$ with optimal top-1 accuracy at $w = 7$ and optimal accuracy on tail-classes at $w = 9$. Compared to our original proposal that utilises also future context (see Table 2.1 on main paper), the overall performance degrades, indicating that leveraging future context is beneficial.

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
w	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
1	67.93	52.29	41.30	90.53	76.47	61.52	61.13	44.60	32.58	42.05	27.42	21.48
3	68.42	54.15	42.59	91.20	78.52	61.10	**61.69**	**44.41**	**32.11**	40.11	31.26	22.58
5	68.58	54.04	42.75	**90.96**	78.27	62.04	59.81	43.94	**32.11**	39.49	31.11	22.48
7	**68.88**	**54.31**	**42.96**	90.89	77.87	62.39	61.41	43.38	32.02	40.51	32.00	23.61
9	68.77	54.28	42.77	90.66	77.72	**62.44**	60.38	45.07	31.83	**40.80**	**32.68**	**23.86**
11	67.83	54.04	42.13	90.63	**78.85**	62.10	57.46	43.94	31.46	36.88	30.74	21.96

Table 2.10: Online action recognition results by varying temporal context length in the validation set of EPIC-KITCHENS-100.

		Overall						Unseen Participants			Tail-classes		
		Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Layers	Shared	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
1	-	69.58	55.04	43.71	91.27	79.02	63.96	61.03	46.01	33.33	42.10	32.42	24.09
2	✗	69.49	55.41	43.94	**91.13**	78.96	63.86	**62.72**	46.57	34.37	**41.42**	**32.32**	23.90
4	✗	**71.01**	**56.55**	**46.04**	90.98	**79.28**	**63.97**	62.35	**47.61**	**35.96**	39.94	31.89	**24.22**
6	✗	69.58	55.68	44.89	90.28	78.17	63.37	61.31	45.63	34.46	38.75	32.21	23.90
2	✓	69.82	55.37	43.81	91.09	78.97	**64.26**	61.50	44.32	32.68	**42.05**	32.58	23.74
4	✓	**70.60**	**56.26**	**45.48**	**91.14**	**79.06**	63.06	**63.76**	**47.14**	**35.87**	41.36	**32.84**	**24.70**
6	✓	69.58	55.68	44.89	90.28	78.17	63.37	61.31	45.63	34.46	38.75	32.21	23.90

Table 2.11: Analysis of performance using different number of layers, both w. and w/o weight sharing. Results are shown in the validation set of EPIC-KITCHENS-100.

2.6.5 Architecture ablations

In Table 2.11, we explore different number of layers in MTCN, both without and with (layer-wise) weight sharing, and compare each case with a single layer. Note that we use the same number of layers and sharing strategy for both AV and LM. We use bold to indicate best performance within each group rather than overall. Best results are obtained using four layers in most metrics, both without & with weight sharing. These outperform a single layer, demonstrating that is beneficial to use a multi-layered transformer. Although MTCN without weight sharing performs slightly better, our proposed model has 2.7× less parameters with only a minor drop in performance.

In Table 2.12, we compare the effect of different types of positional encodings. Particularly, we replace our chosen absolute learnt positional encoding with relative positional encodings [Shaw et al. 2018] and Fourier feature positional encodings [Jaegle et al. 2021]. Fourier feature positional encodings replace our learnable absolute positional encodings with non-learnable ones represented as a vector of log-linearly spaced frequency bands up to a maximum frequency. Relative po-

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Pos. enc.	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
Fourier PE	69.60	56.13	44.63	90.65	78.86	63.31	63.29	45.63	35.12	38.86	32.53	23.09
Relative PE	70.32	**56.30**	45.37	91.01	**79.35**	**64.04**	61.41	46.67	33.99	**41.42**	**33.74**	**24.96**
Absolute PE (Proposed)	**70.60**	56.26	**45.48**	**91.14**	79.06	63.06	**63.76**	**47.14**	**35.87**	41.36	32.84	24.70

Table 2.12: Comparison of different positional encodings (PE) using the validation set of EPIC-KITCHENS-100.

sitional encodings replace our absolute positional encodings of the inputs, with positional encodings representing distances between tokens and placed within the self-attention layers. As shown in the table our proposed absolute learnable positional encodings outperform Fourier feature positional encodings in all metrics (except top-5 action accuracy). Comparing to relative positional encodings, our positional encodings are slightly better in top-1 verbs and actions, as well as in unseen participants, while relative positional encodings perform slightly better in top-5 accuracy and tail classes. Overall, there are no notable differences between the different choices of positional encodings.

2.6.6 EGTEA Implementation Details

Visual features. For EGTEA, we train SlowFast [Feichtenhofer et al. 2019] using the EPIC-KITCHENS pre-trained model, by sampling a clip of 2s from an action segment similar to EPIC-KITCHENS. We use a learning rate of 0.001, no warm-up, and we keep the batch normalisation layers frozen. All unspecified hyperparameters remain unchanged. For feature extraction, we follow the same procedure as EPIC-KITCHENS, except that we use clips of 2s rather than 1s.

Train/Val Details. Here, we discuss differences in the architecture for training/evaluating EGTEA. Remember that for EGTEA we train only vision and language as EGTEA does not contain audio. First, as there is no audio input to the transformer, we do not use modality encodings either. Second, following previous methods [Yin Li et al. 2018; Min and Corso 2021; M. Lu et al. 2019; Sudhakaran and Lanz 2018; Sudhakaran et al. 2019; Y. Huang et al. 2020] that train using a single head for actions and report only action accuracy, we use a single summary embedding for actions, rather than verb/noun embeddings. Accordingly, the language model utilises a single word-embedding for actions, with a dimension of 512. For training the visual-only transformer, we use a learning rate of 0.001,

	Visual		Visual + LM	
w	Top-1(%)	Mean Class (%)	Top-1(%)	Mean Class (%)
1	72.26	64.98	72.26	64.98
3	72.55	64.86	**73.59**	65.87
5	**73.10**	**65.42**	73.49	65.57
7	72.26	64.38	73.19	65.31
9	72.55	64.86	73.44	**66.02**

Table 2.13: Ablation of temporal context extent and language model in the first test split of EGTEA.

train the model for 50 epochs and decay the learning rate at epochs 25 and 38, while keeping all other hyperparameters unchanged. We use same hyperparameters for the language model. For evaluation, differently than EPIC-KITCHENS, we average the predictions of the 10 clips per action, rather than feeding all 10 clips in the transformer.

2.6.7 Ablation of temporal context and language model in EGTEA

We study the effect of the temporal context length both with and without the language model on the first test split of EGTEA. Results are shown in Table 2.13. For the visual only model, top-1 accuracy increases when we increase the length of temporal context from $w = 1$ to 5, and optimal results for both top-1 and mean class accuracy are obtained for $w = 5$. When the language model is incorporated top-1 accuracy increases from $w = 1$ to 3 and then decreases while best mean class accuracy is obtained at $w = 9$. These findings showcase that our model successfully utilises context in this dataset as well. The language model is helpful for EGTEA, and provides a bigger boost in performance than EPIC-KITCHENS, possibly due to the absence of audio modality. Finally, it is worth noting that after the addition of the language model best performance is obtained at a shorter temporal context, showing that shorter sequences of actions provide a stronger prior in this dataset.

2.6.8 Attention Visualisation

In Fig. 2.5, we show additional qualitative examples, similar to Fig. 2.4 in the main paper. These demonstrate how our model attends to temporal context. In the first three examples, the model predicts the centre action correctly, while in

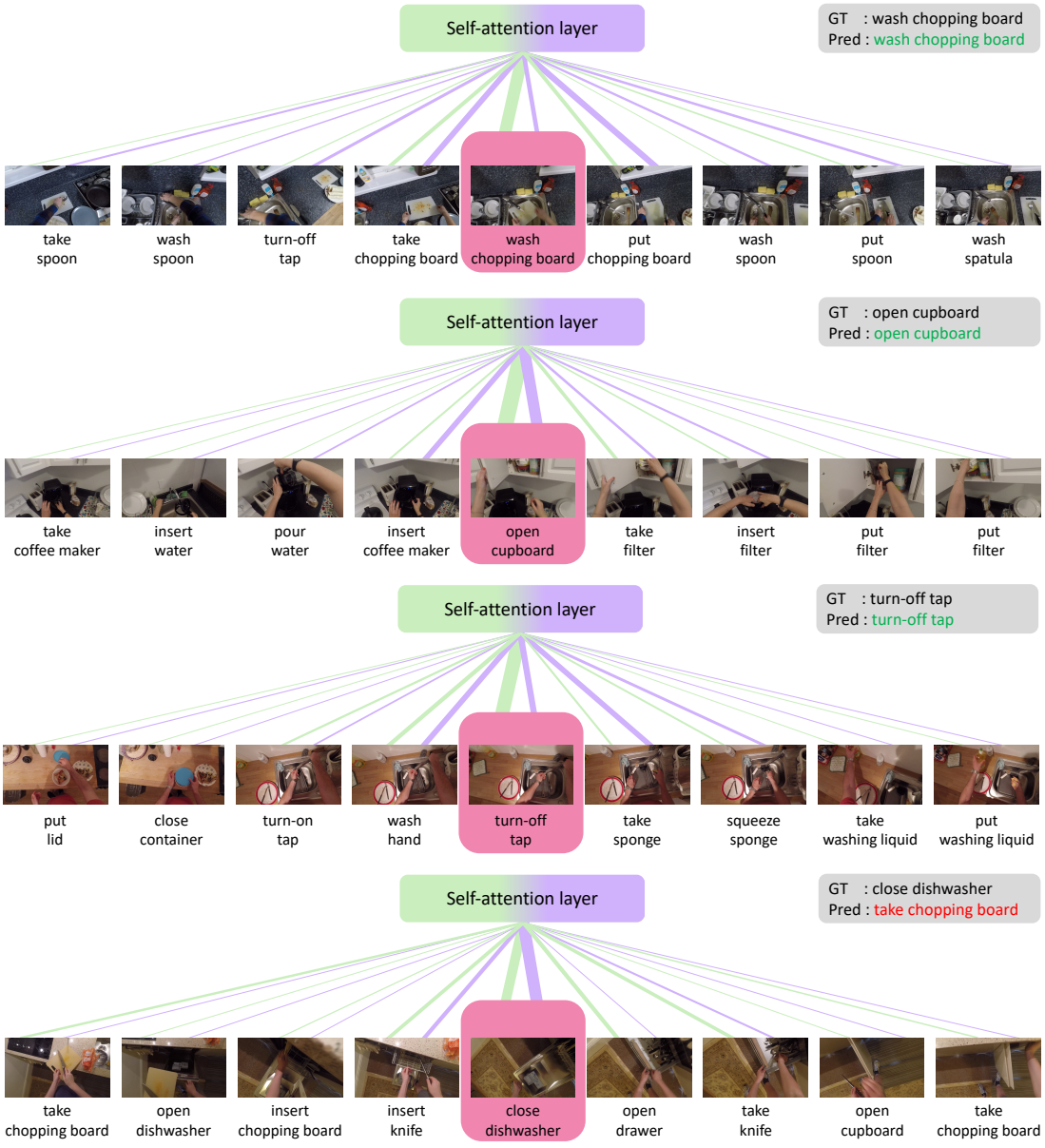


Figure 2.5: Additional qualitative results of attention weights along with the predictions of our model. Green and purple edges represent attention to visual and auditory tokens, respectively, from the noun summary embedding. Thickness indicates attention weights magnitude to centre (bordered) and temporal context actions.

the last one it gives incorrect predictions. In the 'wash chopping board' example, the model particularly attends to actions containing the chopping board. For 'open cupboard', the model has high audio-visual attention to the centre action, and high attention to the audio of the previous action ('insert coffee maker'), showing that at times audio provides useful temporal context. The importance of audio is also apparent in the third example. A source of error in the model results from confusing the centre action with another action in the sequence; in the fourth example 'close dishwasher' is predicted as 'take chopping board' which

corresponds to the first and last actions in the temporal context.

Chapter 3

TIM: A Time Interval Machine for Audio-Visual Action Recognition

The paper has been accepted for publication at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

TIM: A Time Interval Machine for Audio-Visual Action Recognition

Jacob Chalk[1*] Jaesung Huh[2*] Evangelos Kazakos[3]
Andrew Zisserman[2] Dima Damen[1]
[1]University of Bristol [2]VGG, University of Oxford
[3] Czech Technical University in Prague

Abstract

Diverse actions give rise to rich audio-visual signals in long videos. Recent works showcase that the two modalities of audio and video exhibit different temporal extents of events and distinct labels. We address the interplay between the two modalities in long videos by explicitly modelling the temporal extents of audio and visual events. We propose the Time Interval Machine (TIM) where a modality-specific time interval poses as a query to a transformer encoder that ingests a long video input. The encoder then attends to the specified interval, as well as the surrounding context in both modalities, in order to recognise the ongoing action.

We test TIM on three long audio-visual video datasets: EPIC-KITCHENS, Perception Test, and AVE, reporting state-of-the-art (SOTA) for recognition. On EPIC-KITCHENS, we beat previous SOTA that utilises LLMs and significantly larger pre-training by 2.9% top-1 action recognition accuracy. Additionally, we show that TIM can be adapted for action detection, using dense multi-scale interval queries, outperforming SOTA on EPIC-KITCHENS-100 for most metrics, and showing strong performance on the Perception Test. Our ablations show the critical role of integrating the two modalities and modelling their time intervals in achieving this performance. Code and models at: https://github.com/JacobChalk/TIM.

*Equal technical contribution.

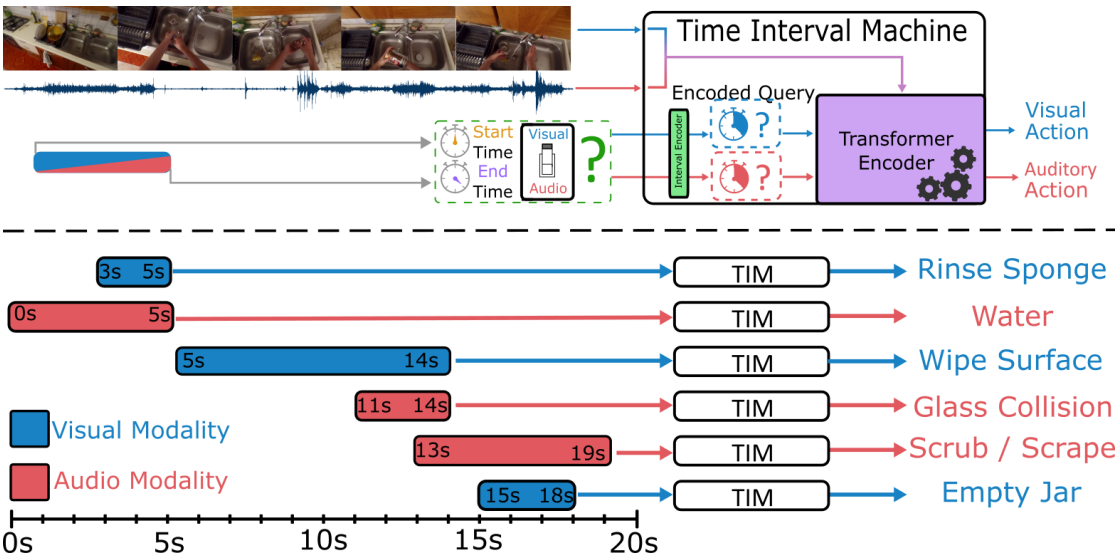


Figure 3.1: Time Interval Machine (TIM): **Top:** Given a visual and auditory stream input, the ongoing action in a particular *time interval* is determined by a query specifying the start and end time of the interval, along with the modality of interest. **Bottom:** TIM can query for visual (e.g. 'Rinse Sponge') and auditory (e.g. 'Water') action classes, as well as distinguish between overlapping actions within the same modality ('Glass Collision' and 'Scrub / Scrape').

3.1 Introduction

Long videos exhibit a quick succession of auditory and visual events. The latest attempts to annotate events in these modalities separately [Huh et al. 2023; Patraucean et al. 2024], showcase that both the temporal extents and class labels differ between the two. However, these events still remain correlated – identifying temporally close events in both modalities can improve recognition of actions in both visual and audio.

Furthermore, most methods to date typically only utilise the exact temporal extent of an action; a precise, trimmed clip of the action is fed into a convolutional [Limin Wang et al. 2016; Carreira and Zisserman 2017; Feichtenhofer et al. 2019] or transformer-based [Arnab et al. 2021; Girdhar et al. 2022; Z. Liu et al. 2022] backbone, which predicts the action taking place. Even when the surrounding context is utilised to improve action recognition [C.-Y. Wu et al. 2019; Sener et al. 2020; Kazakos et al. 2021a], again, this context is supplied in the form of exact clips of neighbouring actions, rather than the untrimmed long input video.

In this paper, we propose an approach that encodes multiple events that occur in both visual and auditory streams of a long video input. We achieve this by elevating *time intervals* to first-class citizens, utilising them to specify a query

within an accompanying modality. We term this mechanism a **Time Interval Machine (TIM)**. It can output the actions that occur *within the queried intervals of the queried modalities*.

This approach has the following benefits:

- TIM is able to receive a long video input, and utilises the temporal context of the video.
- TIM can distinguish between different, potentially overlapping, events within the same input by querying the time interval of a particular event within a given modality.
- TIM can be trained with a large-scale pre-training dataset, since we can chop the random video clips within the dataset and use the same interval for all modalities.

Consider the example in Fig. 3.1. The input contains the sound of water running while a sponge is being rinsed, which is then used to wipe a surface. These distinct events may vary significantly in duration and may be more prominent in the audio or visual modality. Despite differences between these events, there are likely many correlations between them and the surrounding context, which may be beneficial to recognising a given event (e.g. the sound of water is relevant to rinsing a sponge, providing useful information for recognising the visual action). TIM is able to exploit this by accessing the context within both modalities, including the background when no events occur.

We test TIM on three challenging audio-visual **recognition** datasets consisting of long videos: EPIC-KITCHENS [Damen et al. 2022], which recently offered distinct audio annotations through EPIC-SOUNDS [Huh et al. 2023], the Perception Test [Patraucean et al. 2024], and AVE [Tian et al. 2018]. We show that TIM can effectively learn both visual and auditory classes within a long input, outperforming the current SOTA top-1 accuracy on EPIC-KITCHENS by 2.9% and 1.4% on EPIC-SOUNDS, despite competing methods for the former using far larger pre-training datasets, large language models or higher resolution input. We also outperform models pre-trained with public datasets on AVE by 0.6% and improve over a strong baseline on the Perception Test in visual and audio action recognition by 9.9% and 3.2% respectively.

Additionally, we adapt TIM to action **detection**, through fixed multi-scale dense querying with an added *interval regression* loss. We report strong detection results on EPIC-KITCHENS and the Perception Test, outperforming Action Former [C.-L. Zhang et al. 2022] by 1.6 and 4.3 mAP respectively.

Our contributions are summarised as: (i) we propose the TIM query mechanism for attending to modality-specific intervals in long videos. (ii) we efficiently train TIM to encode/query multiple audio-visual actions using time intervals. (iii) we showcase the value of TIM for both visual and auditory action recognition, and adapt it for detection with an added interval regression loss. (iv) we achieve new SOTA in both video and multi-modal recognition on multiple datasets.

3.2 Related Works

Audio-visual action recognition. A number of works have employed audio and visual modalities for action recognition [Xiao et al. 2020; Gao et al. 2020; Nagrani et al. 2021; W. Wang et al. 2020; Kazakos et al. 2019; Kazakos et al. 2021a]. Some introduce new architectures to effectively fuse modalities [Kazakos et al. 2019; Xiao et al. 2020; Nagrani et al. 2021; Kazakos et al. 2021a]; others propose unique training techniques to solve problems occurring while training multi-modal models, such as Gradient Blending [W. Wang et al. 2020], to tackle overfitting at different speeds for each modality, or contrastive learning for cross-modal discrimination [Morgado et al. 2021]. However, these works use the same set of semantic and temporal labels for both modalities. Recent works have shown that both the temporal intervals and semantics of events differ between modalities [Huh et al. 2023; Patraucean et al. 2024]. [Tian et al. 2020] temporally annotates visual and auditory events independently, although they share the same set of labels. In this work, we train with distinct labels for each modality to leverage discriminative audio and visual actions.

Leveraging temporal context. Several works have considered incorporating temporal context [Ng and Fernando 2019; C. Zhang et al. 2021; C.-Y. Wu et al. 2019; C.-Y. Wu et al. 2022; Kazakos et al. 2021a], a direction orthogonal to employing multiple modalities and particularly useful in untrimmed videos. An auto-regressive LSTM-based encoder-decoder is proposed in [Ng and Fernando

2019] for action sequence classification, effectively leveraging past action context to predict the current action. The Temporal Query Network [C. Zhang et al. 2021] uses learnable query vectors that correspond to specific attributes of a long video, allowing the model to attend to the aspects of the video and its surrounding context to produce a response for each attribute. [C.-Y. Wu et al. 2019] proposes to enhance the representation of the action by aggregating temporal context from neighbouring action clips using a Long-Term Feature Bank along with an attention mechanism. [C.-Y. Wu et al. 2022] crafts a more sophisticated memory bank by storing keys and values of all the intermediate layers of a transformer to aggregate the past context. Lastly, [Kazakos et al. 2021a] exploits multi-modal temporal context from surrounding actions using vision, audio, and language.

[C.-Y. Wu et al. 2022; C.-Y. Wu et al. 2019; Kazakos et al. 2021a] are the closest to our approach, in that the common goal is to enrich the representation of the action of interest using surrounding context from the untrimmed video, rather than neighbouring clips. Nevertheless, [C.-Y. Wu et al. 2022; C.-Y. Wu et al. 2019] are single modality models, recognising visual actions solely. [Kazakos et al. 2021a] assumes the temporal extents of all actions are known, including for the test set, which is restrictive.

Queries in visual models. Learning visual queries with Transformer architectures has gained recent attention [Carion et al. 2020; Locatello et al. 2020; C. Zhang et al. 2021; Herzig et al. 2022; Jia et al. 2022]. Commonly, approaches employ a set of learnable vectors that are used to inquire about the presence of a concept in the input. For example, in [Locatello et al. 2020; Carion et al. 2020] the learnable queries correspond to different objects, whereas in [Herzig et al. 2022] they are used for multi-task learning and each learnable query corresponds to a different task. [Jia et al. 2022] has incorporated learnable queries for adapting a pre-trained model while keeping the rest of its parameters frozen. Closest to our motivation is [C. Zhang et al. 2021], where the queries correspond to events and their attributes for fine-grained action recognition in videos. The authors note that the queries also have the role of temporally localising the events in untrimmed videos.

Different from [C. Zhang et al. 2021] and other works, our queries are primarily temporal with no semantic interpretation and are applied to multiple modalities.

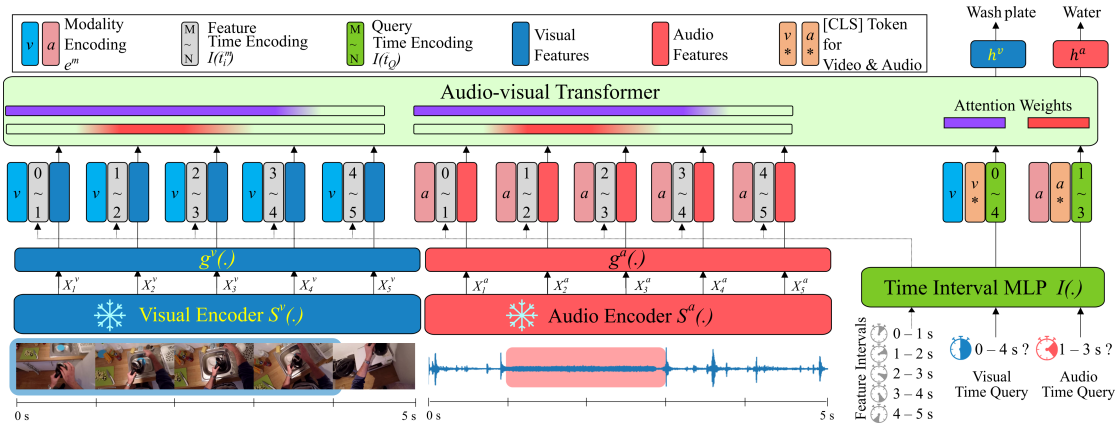


Figure 3.2: **Overview of the Time Interval Machine (TIM)**. The model ingests a sequence of audio and visual features from a video, with each feature time-stamped by the temporal interval it spans, and encoded with its modality. To infer the action occurring over a temporal interval (a visual or audio event) a query is formed specifying the interval and modality of interest.

Importantly, since time is continuous, we cannot use a predefined set of queries. Instead, we employ an MLP architecture to encode time, in a form akin to a universal clock. We present our approach next.

3.3 Time Interval Machine

In this section, we describe the **Time Interval Machine** (TIM), a multi-modal transformer encoder architecture where all inputs, both features and queries, are encoded with their associated *time intervals*. A time interval incorporates the duration and position of each audio and visual feature and is also used to *query* the network for any action occurring within the given time interval.

The architecture of TIM is illustrated in Fig. 3.2. It ingests a large video input, represented as a sequence of audio and visual features, and outputs the ongoing auditory or visual action label for the provided query time intervals.

3.3.1 Model architecture

Input. The input to TIM is a long crop of the untrimmed video, represented by extracted features. When considering two modality inputs, such as video and audio, each modality is embedded separately as follows: for each modality m, let $\mathbf{X}^m = [X_1^m, \cdots, X_{N^m}^m]$ be N^m temporally-ordered feature representations of the input video, obtained from a pre-trained feature extractor $S^m(\cdot)$. We feed

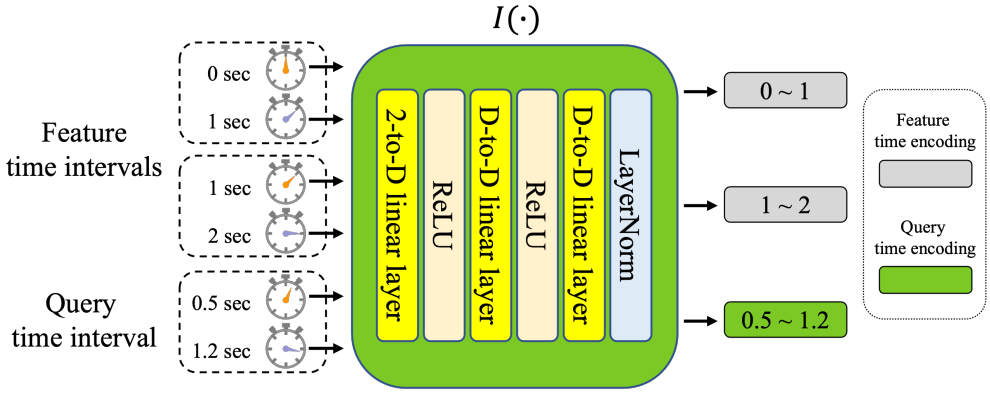


Figure 3.3: Illustration of the Time Interval MLP $I(\cdot)$. It inputs the **two** dimensional vector, start and end times of an interval, and produces a single vector, which can be concatenated along the channel dimension to either input features or `[CLS]` tokens. The figure shows three time interval inputs and three corresponding outputs. Note that in practice, time intervals are ingested simultaneously.

the features through modality-specific embedding layers $g^m(\cdot)$, projecting them to a lower, common dimension D across all modalities. The embedded features[1] are then tagged with modality encodings and time interval encodings, forming the input to the transformer encoder. We now detail how we encode the time intervals.

Encoding Time Intervals. In this work, we introduce a new type of learnt query network, the **Time Interval MLP**, which produces a single D-dimensional vector representing a given time interval. This network is used within TIM to encode the time intervals of the input features and the time interval we wish to query, and later classify. Fig. 3.3 illustrates the concept of this network.

The Time Interval MLP $I(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}^D$ receives a time interval, represented by start and end time, and produces a single D-dimensional encoding. Note that this is distinct from encoding the start and end times separately. Specifically, let t_s and t_e be the start and end time of an interval of interest, normalised by the length of the long video input. $I(\cdot)$ receives the interval $\tilde{t} = [t_s, t_e]$ as input, and outputs a D-dimensional vector encoding of that interval. This vector encodes both the relative position of the time interval within the input, as well as its duration. This vector then acts as a *query* for the model concerning the action taking place within the interval. Furthermore, each feature $\{X_i^m\}$ spans a certain time interval within the input. Thus, it is important to also encode the time intervals of the features.

In summary, the Time Interval MLP acts as an *universal clock*, which encodes the

[1]Note that the number of features can differ between modalities

temporal extent of features, from any modality, within the input. Note that it is critical that the same Time Interval MLP is used for encoding all time intervals of the input features and queries across both modalities to accurately encode universal time. It is also important to note that Time interval MLP can cover *continuous* time intervals, whereas traditional positional encoding only covers a fixed set of positions of the input features. It also enables to query any time interval within the input, even for overlapping events which is not possible with traditional positional encoding. The Time Interval MLP is trained end-to-end along with the transformer.

Transformer Feature Inputs. Let $\tilde{\mathbf{t}}^m = [\tilde{t}_1^m, \cdots, \tilde{t}_{N^m}^m]$ be the corresponding time intervals of the video's features $\mathbf{X}^m$ from the modality m. We inject the encoded time interval $I(\tilde{\mathbf{t}}^m)$ into the embedded features via channel-wise concatenation. A learnable modality-specific encoding $e^m \in \mathbb{R}^{2D}$ is then summed to the temporally-encoded features to discriminate between each modality. In summary, the feature inputs $\mathbf{E}^m$ for TIM are computed by,

$$E_i^m = [g^m(X_i^m), I(\tilde{t}_i^m)] + e^m \quad \forall i \in [1, ..., N^m] \tag{3.1}$$

where $[\cdot, \cdot]$ indicates concatenation.

Transformer Query Inputs. To query for an action within an interval of interest, we adopt a standard approach of appending a learnable classification token to the input sequence, $\texttt{CLS}^m$. If $\tilde{t}_Q$ is an interval of interest, we concatenate the time interval representation $I(\tilde{t}_Q)$ to this classification token along the channel dimension, which acts as a query for the network in order to predict the corresponding action happening within $\tilde{t}_Q$. We also add the modality-specific encoding e^m to each classification token, as a flag to distinguish between which modality we are querying. The encoded $\texttt{[CLS]}^m$ tokens can be more formally defined as:

$$\texttt{[CLS]}^m = [\texttt{CLS}^m, I(\tilde{t}_Q)] + e^m \tag{3.2}$$

During training, we add a classification token for *each* action within the input video, resulting in multiple `[CLS]` tokens across both modalities.

Transformer Encoder. We use a transformer encoder to perform self-attention

on the input sequence to aggregate relevant temporal context and cross-modal relations.

We form the transformer input sequence with the encoded feature inputs $\mathbf{E}^m$ and **one or more** classification tokens $\texttt{[CLS]}^m$, representing each time interval query, and feed these into the encoder. Note that we recognise all actions from any modality simultaneously by appending multiple $\texttt{CLS}^m$ tokens to the input. The transformer output representation of $\texttt{[CLS]}^m$, namely $Z^m_{\texttt{CLS}}$, is then passed to the corresponding linear classifier to predict the action labels.

Importantly, we use an attention mask to prevent queries from attending to one another, and we similarly prevent input features from attending to queries. This ensures each query is recognised without the privileged knowledge of any other query, or action boundary, during inference.

3.3.2 Training and Testing in TIM

To train TIM, we consider all long segments of W seconds and stride H_w across the entirety of the untrimmed videos. We randomly select batches from these. For each window, we query all annotated audio and visual actions that overlap with the window by more than $\delta = 0.2$ seconds.

All queries in the window are encoded and concatenated to separate $\texttt{CLS}$ tokens. To classify queries, let $h^m_{\texttt{CLS}}(\cdot)$ be a linear classifier for modality m, and let $\hat{y}^m_{\texttt{CLS}} = h^m(Z^m_{\texttt{CLS}})$ be the predicted action of the output representation $Z^m_{\texttt{CLS}}$. We train TIM by using a cross-entropy classification loss $CE(\cdot)$ on the ground truth $y^m_{\texttt{CLS}}$ by:

$$L^m = \frac{1}{N_Q} \sum^{N_Q} CE(\hat{y}^m_{\texttt{CLS}}, y^m_{\texttt{CLS}}) \tag{3.3}$$

where N_Q is the number of queries within the batch.

Temporal Distance Loss. In addition to the standard classification loss, we introduce a Temporal Distance (TD) loss as an auxiliary loss for training TIM. Inspired by [Y. Liu et al. 2021], where relative patch positions in token embeddings are learnt using self-supervision, we similarly train the network to take two transformer outputs and predict the elapsed time between their corresponding time intervals.

Let the $\mathbf{Z}_{1:\sum_m N^m}$ be the transformer outputs of the features from all modalities. We randomly sample a set of feature pairs $\mathbb{B} \subset \mathbf{Z}_{1:\sum_m N^m}$ from these outputs, concatenate along the channel dimension and feed them to the temporal distance regression head $h_{\tilde{t}}(\cdot) : \mathbb{R}^{4D} \rightarrow \mathbb{R}^1$ to predict the time interval difference between each pair. Note that feature pairs can be sampled both *within* and *across* modalities. In our case, we sample across modalities by pairing one visual feature with another audio feature. This helps the model to learn the temporal relations between modalities.

Formally, a TD loss L^{td} is computed as:

$$L^{td} = \sum_{\{Z_i, Z_j\} \in \mathbb{B}} |h_{\tilde{t}}(Z_i, Z_j) - d_{ij}| \tag{3.4}$$

where d_{ij} is the temporal distance between intervals $\tilde{t}_i, \tilde{t}_j$.

Training objective and regime. For our final training loss, we sum the losses across modalities along with the TD loss:

$$L^{total} = \left(\sum_{m \in \mathbb{M}} \lambda^m L^m \right) + \lambda^{td} L^{td} \tag{3.5}$$

where $\mathbb{M}$ is a set of modalities, λ^m controls the strength of each modality's loss and λ^{td} is a hyperparameter that controls the strength of the TD loss.

Test-Time Augmentation. We use test-time augmentation, as this generally increases prediction robustness and performance [Shanmugam et al. 2021; Patrick et al. 2021]. In TIM, we use a sliding window across the untrimmed video, thus feeding the same interval query with varying contexts. We then aggregate predictions of the same interval query across windows to make the final prediction.

3.3.3 Adapting for Detection

While primarily designed for recognition, we can adapt TIM for detection. The backbone remains largely unchanged from recognition, but there are two main differences. First, we construct dense multi-scale interval queries spanning the entirety of the video input at each scale. These are used as interval queries in both training and detection inference. The multi-scale intervals allow for detecting both

long and short actions. Second, we introduce an additional interval regression head, which regresses the query interval to the action's exact temporal duration.

During training, we deem any query in the multi-scale pyramid that overlaps with a ground truth action by more than some IoU threshold as a positive query. In addition to classifying the query, we train a DIOU regression loss [Z. Zheng et al. 2020] to predict the exact interval of the action. Both classification and interval regression losses are trained jointly. We provide full details in the ArXiv appendix.

3.4 Experiments

This section describes the datasets used to evaluate our model, implementation details and results along with a comparison to the state-of-the-art methods.

3.4.1 Dataset

EPIC-KITCHENS-100 [Damen et al. 2022] is a large-scale video dataset, including 700 egocentric videos recording actions in kitchens. It consists of 89,977 segments of fine-grained actions. Inspired by prior works [Girdhar et al. 2022; Sudhakaran et al. 2021; Tai et al. 2022], we directly predict the action out of 3806 classes present in the train and validation set to avoid predicting invalid actions.

EPIC-SOUNDS [Huh et al. 2023] offers audio annotations capturing temporal extents and class labels within the audio stream of EPIC-KITCHENS-100. The annotations contain 78,366 labelled audio events. We combine the visual annotations from EPIC-KITCHENS with the audio annotations from EPIC-SOUNDS to train our audio-visual model. TIM can recognise actions from both datasets using a single model.

AVE [Tian et al. 2018] contains 4,143 videos covering a range of real-life scenes and labelled with 27 categories, such as church bell, male speaking and dog barking. Each video is equally divided into 10 segments, each 1 second in length. We evaluate TIM on the supervised audio-visual event localisation task. Given a 1 second segment, we recognise the ongoing action out of the 27 categories *plus* a background class.

Perception Test [Patraucean et al. 2024] is a recent multimodal video bench-

mark of 11,620 videos with an average length of 23 seconds, and provides both temporal action and sound annotations. There are 73,503 visual annotations spanning 63 classes, versus 137,128 sound annotations over 16 classes.

3.4.2 Implementation Details

Architectural Details. Visual and audio embedding layers g_m consist of a single 512-D feed-forward layer, followed by a GELU [Hendrycks and Gimpel 2016] activation and layer normalisation [Ba et al. 2016] are used to project the features to a common space. The Time Interval MLP I consists of three linear layers with 512-D hidden dimension, followed by ReLU activations, with layer normalisation after the output of the last linear layer. We include 512-D learnable `[CLS]` tokens: $\texttt{[CLS]}^m_{action}$ for each query in each modality which become 1024-D after concatenation with the encoded time interval. They are then summed with 1024-D modality encodings; e^m.

The audio-visual transformer contains four encoder layers, each with 8 attention heads, GELU activations, and 1024-D keys, queries and values. A dropout rate of $p = 0.1$ is applied within the encoder layers. We also apply channel-wise dropout with $p = 0.5$ directly to the raw input features, as well as to the encoded transformer input. The temporal distance head consists of two linear layers with a hidden dimension of 1024 and a third which outputs a single number corresponding to the elapsed time between each time interval. We include the architectural ablations on encoder layers and the temporal distance head in the ArXiv appendix.

Training / Validation Details. We train each model for 100 epochs, usingAdamW [Loshchilov and Hutter 2017] with a batch size of 64 and a weight decay of 1e-4. A linear learning rate warm-up is applied for the first two epochs, starting from 1e-6 to a target learning rate, and we use a cosine learning rate scheduler. We set the TD loss weight λ^{td} to 0.3. We pad the queries for each window in the batch to the maximum number of queries in a single window in each dataset. We provide implementation details per dataset in the ArXiv appendix.

3.4.3 Results

We compare TIM with SOTA models for each dataset.

EPIC-KITCHENS / EPIC-SOUNDS Results. We train a single model on both visual and audio labels of the EPIC-KITCHENS videos, reporting results on both datasets.

For the visual features, we concatenate Omnivore [Girdhar et al. 2022] and VideoMAE-L [Tong et al. 2022] features along the channel dimension, forming 2048-D features. For the audio features, we use Auditory SlowFast [Kazakos et al. 2021b], which generalises well across diverse audio domains [Luyu Wang et al. 2022]. For both modalities, we extract 1 second features every 0.2s. For training, we extract additional augmented feature sets - with RandAugment [Cubuk et al. 2020] for visual and SpecAugment [D. S. Park et al. 2019] for audio features.

Table 3.1 compares TIM with the SOTA models on EPIC-KITCHENS-100. We outperform M&M Mix [Xiong et al. 2022] by 5.1% on verb, 0.9% on noun and 3.9% on action. Compared to our model, both MTV and M&M Mix are trained with an additional private dataset [Stroud et al. 2020] which contains 194K hours of 70 million videos while we only use the open-source visual backbone pre-trained with public datasets. We also outperform LaViLa [Y. Zhao et al. 2023] and AVION [Y. Zhao and Krähenbühl 2023] which leverage pre-trained LLMs to learn video representations.

We note that we outperform all prior works, often without additional techniques that boost performance. For example, we use short-sided cropped 224×224 images while [Xiong et al. 2022] uses 420×420, which enlarges the spatial resolution of objects in the egocentric video, enabling better noun recognition. We expect a further performance boost when implementing any of: higher resolution feature extractors, additional large-scale pre-training and the introduction of a LLM. We leave this as an avenue for future work.

Table 3.2 compares TIM against prior results on EPIC-SOUNDS, where TIM outperforms SOTA by 1.4%.

For detection, we show that TIM can produce competitive results when compared to models primarily designed for this task in Table 3.3. TIM adapted for detection outperforms ActionFormer [C.-L. Zhang et al. 2022] by 2.3 mAP on verb and 1.6 mAP on noun using the same set of features. Interestingly, our features outperform the ActionFormer features by 2.8 mAP on verb and 7.5 mAP on noun when using

Model	*xp*	LLM	Verb	Noun	Action
Visual-only models					
MFormer-HR [Patrick et al. 2021]	336p	✗	67.0	58.5	44.5
MoViNet-A6 [Kondratyuk et al. 2021]	320p	✗	72.2	57.3	47.7
MeMViT [C.-Y. Wu et al. 2022]	224p	✗	71.4	60.3	48.4
Omnivore [Girdhar et al. 2022]	224p	✗	69.5	61.7	49.9
MTV [Yan et al. 2022]	280p	✗	69.9	63.9	50.5
LaViLa (TSF-L) [Y. Zhao et al. 2023]	224p	✓	72.0	62.9	51.0
AVION (ViT-L) [Y. Zhao and Krähenbühl 2023]	224p	✓	73.0	65.4	54.4
TIM (ours)	224p	✗	**76.2**	**66.4**	**56.4**
Audio-visual models					
TBN [Kazakos et al. 2019]	224p	✗	66.0	47.2	36.7
MBT [Nagrani et al. 2021]	224p	✗	64.8	58.0	43.4
MTCN [Kazakos et al. 2021a]	336p	✗	70.7	62.1	49.6
M&M [Xiong et al. 2022]	420p	✗	72.0	66.3	53.6
TIM (ours)	224p	✗	**77.1**	**67.2**	**57.5**

Table 3.1: Comparisons to state-of-the-art *recognition* models on the EPIC-KITCHENS validation set. We report the top-1 accuracy for verb, noun and action (%). LLM: large language model is used during pre-training. xp: input resolution of $x \times x$.

Model	SSAST [Yuan Gong et al. 2022]	ASF [Kazakos et al. 2021b]	DiffSED [Bhosale et al. 2023]	**TIM (A)**	**TIM (A+V)**
Top-1 acc	53.5	53.8	56.9	55.7	**58.3**

Table 3.2: Comparisons to state-of-the-art *sound recognition* models on EPIC-SOUNDS. We report the top-1 accuracy (%) on Val. The performance of SSAST and ASF are from [Huh et al. 2023].

the same architecture. The original paper uses SlowFast [Feichtenhofer et al. 2019] features whereas ours use the combination of Omnivore [Girdhar et al. 2022] and VideoMAE [Tong et al. 2022] features. Omnivore is trained with a mixture of image and video datasets, resulting in the model focusing on the visual objects. The original paper already shows a higher boost on noun compared to verb (See Table 6 in [Girdhar et al. 2022]) resulting in the higher performance of TIM on noun.

AVE Results. As this dataset contains joint audio-visual labels, we train TIM by duplicating the query, i.e. using a `[CLS]` for each modality, and combine their logits during training and inference. We use the pre-trained publicly available models from [Tian et al. 2018] for a fair comparison with other works. We also apply AVGA [Tian et al. 2018] to spatial visual features from VGG-19 before feeding them to the transformer.

Table 3.4 shows our results on the AVE dataset. Combining audio and video significantly improves the performance on TIM. The results from [Feng et al. 2023]

Model	V	A	Average Precision (AP)					
			Task	@0.1	@0.2	@0.3	@0.4	@0.5	Avg.
G-TAD [M. Xu et al. 2020]	✓	✗	Verb	12.1	11.0	9.4	8.1	6.5	9.4
			Noun	11.0	10.0	8.6	7.0	5.4	8.4
ActionFormer [C.-L. Zhang et al. 2022]	✓	✗	Verb	26.6	25.4	24.2	22.3	19.1	23.5
			Noun	25.2	24.1	22.7	20.5	17.0	21.9
ActionFormer - Our Features	✓	✗	Verb	29.6	28.8	26.9	24.4	21.6	26.3
			Noun	34.3	32.6	30.2	27.4	22.6	29.4
TIM	✓	✓	Verb	**32.9**	**31.6**	**29.6**	**27.0**	**22.2**	**28.6**
			Noun	**36.4**	**34.8**	**32.1**	**28.7**	**22.7**	**31.0**

Table 3.3: Comparisons to state-of-the-art *detection* models on the EPIC-KITCHENS validation set. We report the average precision at IOU thresholds $[0.1, 0.2, 0.3, 0.4, 0.5]$ as well as their average across all thresholds on verb, noun.

Model	PSP	CPSP	CSSNet	TIM			
	[Jinxing Zhou et al. 2021]	[Jinxing Zhou et al. 2022]	[Feng et al. 2023]†	V	A	AV	AV⋆
Top-1 acc	77.8	78.6	**80.5**	62.8	65.5	79.2	79.8

Table 3.4: Top-1 event classification accuracy (%) on AVE Test set. †: no official code or public model provided to replicate results. We show the models trained only with publicly available datasets. ⋆: results with Omni+ASF features.

perform best but could not be replicated. We also report TIM using the Omnivore visual features and Auditory Slowfast features used for EPIC-KITCHENS, which achieves a 0.6% boost in performance.

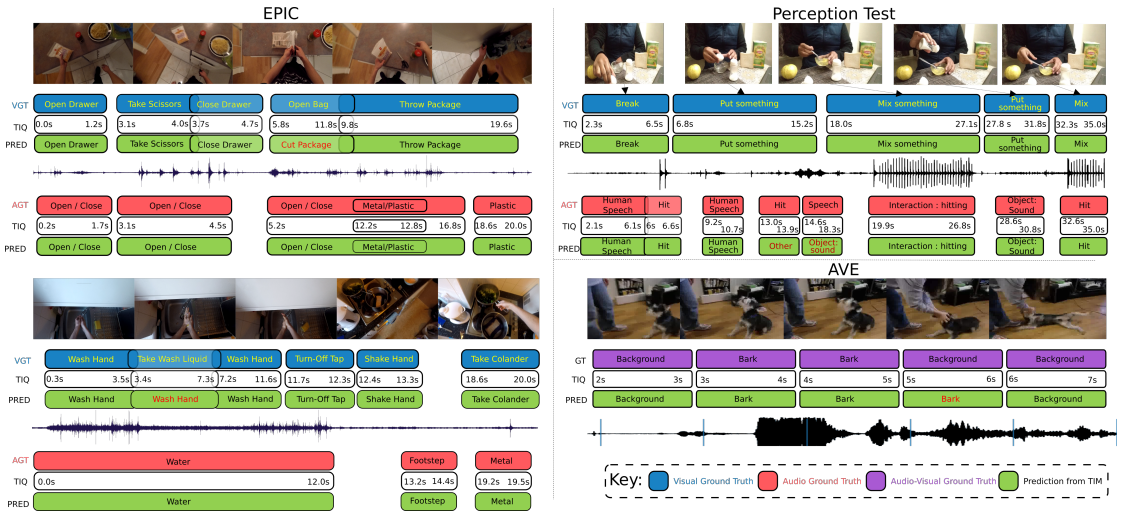


Figure 3.4: Qualitative results for all datasets. **PRED:** Prediction by TIM, **TIQ:** Time Interval Queries, **V/AGT:** Visual/Audio Ground Truth.

Perception Test Results. We use the same backbone for the Omnivore features and Auditory Slowfast features and train a single model using both the visual and audio labels. Table 3.5 compares the results on the newly introduced Perception Test. We train an MLP classifier, with two linear layers and a ReLU activation, on the features directly as a baseline. We also evaluate on an audio-visual model that does use context with MTCN. Compared to these methods TIM clearly shows

Perception Test Action				
Model	MLP (V)	MTCN [Kazakos et al. 2021a](A+V)	**TIM (V)**	**TIM (A+V)**
Top-1 acc	43.7	51.2	56.1	**61.1**
Perception Test Sound				
Model	MLP (A)	MTCN [Kazakos et al. 2021a](A+V)	**TIM (A)**	**TIM (A+V)**
Top-1 acc	50.6	52.9	54.8	**56.1**

Table 3.5: Comparisons to trained recognition baselines on the Perception Test validation split. We show both action and sound recognition and the benefit of including audio-visual in TIM for both challenges. **V** : visual and **A** : audio input features. MLP is the result by training an MLP classifier with the features directly.

Model	**Average Precision (AP)**					
	@0.1	@0.2	@0.3	@0.4	@0.5	Avg.
Perception Test Action						
ActionFormer [C.-L. Zhang et al. 2022]	27.8	27.6	25.2	23.0	20.0	24.5
TIM	**33.5**	**32.2**	**29.8**	**26.4**	**22.0**	**28.8**
Perception Test Sound						
ActionFormer [C.-L. Zhang et al. 2022]	34.7	31.3	27.5	22.7	**17.7**	26.8
TIM	**37.5**	**33.1**	**27.9**	**22.8**	17.2	**27.7**

Table 3.6: Comparisons to strong *detection* models on the Perception Test validation set for action and sound localisation. We report the average precision at IOU thresholds $[0.1, 0.2, 0.3, 0.4, 0.5]$ as well as the average across all thresholds.

significant improvements. Results are improved over MTCN by 9.9% and 3.2% on visual and audio recognition tasks respectively. We also provide detection results in Table 3.6. TIM improves over ActionFormer [C.-L. Zhang et al. 2022] by 3.3 average mAP on visual actions and by 0.9 average mAP on sound, when using the same features.

Cross-Modality in TIM. When referring to our previous results, we see that including the additional modality provides a performance boost in all cases, highlighting TIM's ability to utilise and distinguish between different modalities. For example, on EPIC-KITCHENS-100, including audio improves visual action accuracy by 0.9%. For EPIC-SOUNDS, the visual modality further improves accuracy by 2.6%. In the Perception Test, including the audio modality improves visual recognition by 5.0%, and visual increases sound recognition by 1.3%. Finally, for AVE, we see a significant improvement, where an audio-visual model increases accuracy by 13.7% from audio-only.

Qualitative Results. We present qualitative results in Fig. 3.4. We see that in EPIC-KITCHENS, TIM can competently recognise actions across the two modalities including overlapping queries. Furthermore, we see consecutive actions are

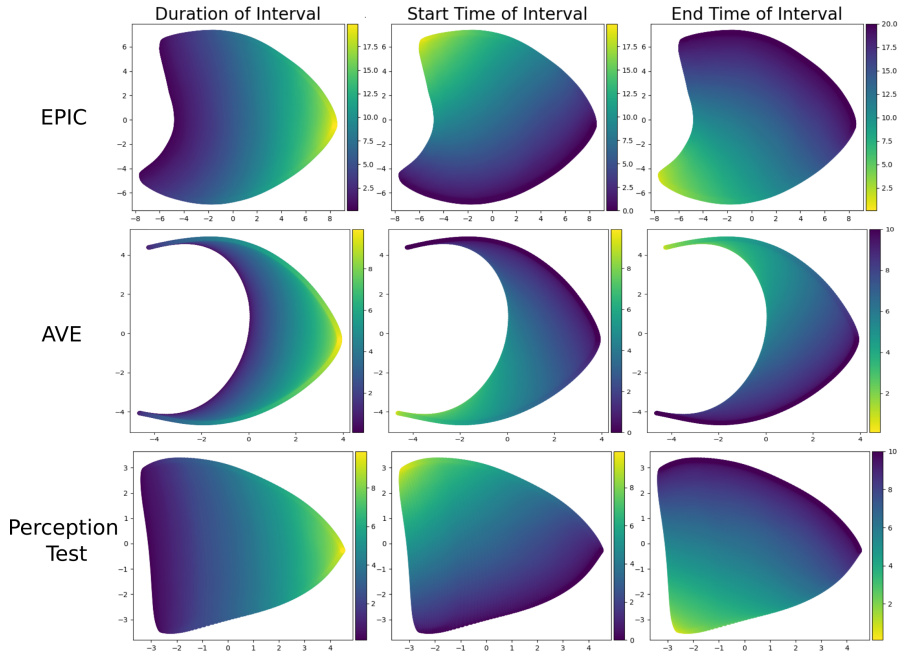


Figure 3.5: TSNE plot for time encodings $I(\cdot)$ on all datasets. In each plot, we use colour maps to indicate encodings of the time interval's duration (left), start time (middle) and end time (right).

correctly recognised with varying interval lengths, such as the 'open / close' audio actions between 0.2s and 4.5s. For AVE, TIM is able to distinguish between the background and a 'barking' audio-visual event based on the time interval query. For the Perception Test, we see that TIM can distinguish between heavily overlapping actions across both modalities, such as between 'break', 'human speech', 'hit' and 'put something'. However, there are also failure cases, such as in EPIC-KITCHENS when the action 'take washing up liquid' is recognised as 'wash hand', as the model is likely confused by the context predominantly associated with the highly overlapping 'wash hand' actions.

3.4.4 Analysing Time Intervals

We showcase the importance of effectively encoding time intervals, as well as how they differ from alternative strategies. We perform this analysis on the EPIC-KITCHENS-100 and EPIC-SOUNDS recognition tasks.

Time Encoding Representation. To show TIM encodes time intervals across all datasets in Fig. 3.5. We use three colour maps on the same TSNE projection to show three properties of the encoded interval: duration, start time and end time. Interestingly, the 1D time encoding perfectly captures all three attributes

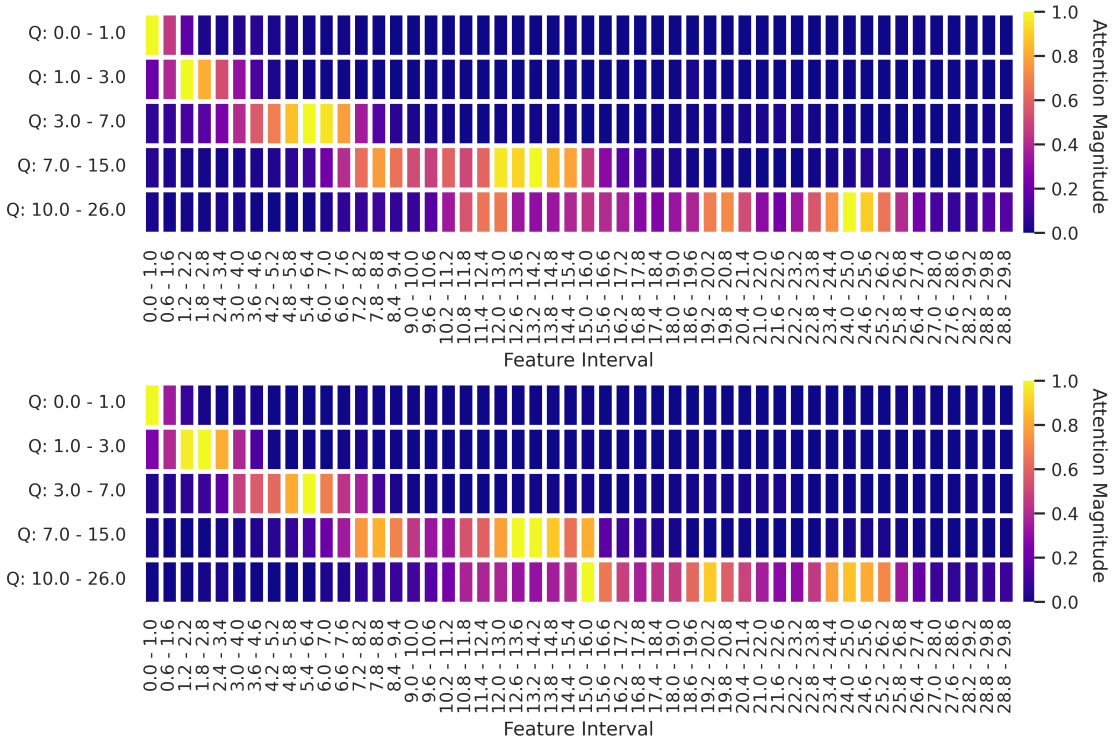


Figure 3.6: Attention heatmaps of the second encoder layer for two random 30s clips in EPIC-KITCHENS. **x-axis**: input feature time intervals; **y-axis** query time intervals of varying position and duration. The attention magnitude relates to the query CLS token.

and across the datasets. While the encodings differ per dataset, as these differ in the positions and durations of actions, we see clear similarities in the learnt time encoding projections. For example, the duration is perfectly captured across the x-axis of the TSNE plot with lower values indicating longer time intervals.

Interval Query Attention. We plot two attention heatmaps in Fig. 3.6 for 5 separate queries with varying positions and scales in EPIC-KITCHENS-100. We extract the attention weights from the second transformer encoder layer, as this appears to be the most relevant to the interval query. The learnt attention clearly applies to the feature time intervals contained within the query. We note the similarity between the attention in the two randomly selected windows.

Shifting Intervals. To show how TIM effectively encodes the time interval of actions, we shift the time interval queries from their correct action interval by -1.5s to 1.5s, assessing the impact of these adjustments on the performance.

Fig. 3.7 shows the result. We see the performance gradually drops, both in visual and audio, as the query interval moves away from the correct action interval. The drop is also symmetric showcasing no bias. Unsurprisingly, the performance drops

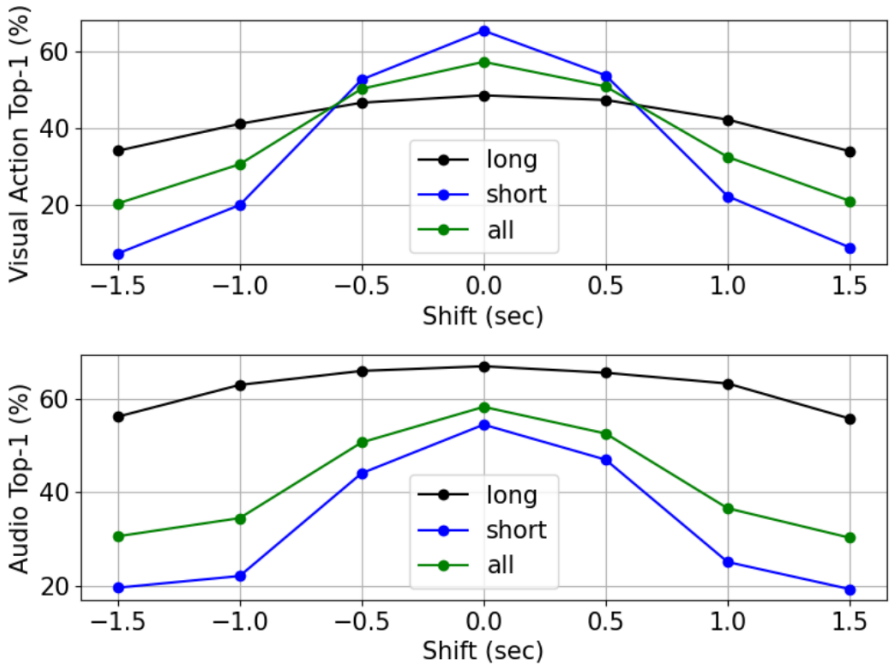


Figure 3.7: The impact of shifting the time interval query on visual performance (top) and audio performance (bottom), both on short actions (< 2 sec) and long actions (> 2 sec) and overall validation set (**all**) for EPIC-KITCHENS-100 and EPIC-SOUNDS.

	EPIC-KITCHENS			**EPIC-SOUNDS**
Encoding	**Verb**	**Noun**	**Action**	**Audio Actions**
Learned	43.8	44.3	29.6	23.7
Sinusoidal	43.8	44.6	30.0	13.4
Centre	74.3	65.8	55.6	56.4
Separate-add	76.0	66.2	56.4	57.7
Interval-add	76.3	66.5	56.9	**58.8**
Separate-cat	76.8	**67.4**	57.1	58.4
Interval-cat (proposed)	**77.1**	67.2	**57.5**	58.3

Table 3.7: Ablating the choice of encoding time intervals.

significantly when shifting short actions both in video (-57.9%) and audio (-35.2%), while it is less extreme in long actions (-14.5% and -11.2%). We assess the impact of scaling the time interval in the ArXiv appendix.

Time Interval Encodings. The Time Interval MLP encodes the interval of the query. Here, we compare this to traditional positional encodings, both sinusoidal and learned. We also experiment on five different variations of the Time Interval MLP, namely: (i) Centre – we only encode the centre timestamp of the interval; (ii) Separate-Add/Cat – we encode the intervals' start and end time separately and add the encoded output vectors together, or concatenate along the channel

dimension; and (iii) Interval-Add/Cat – we encode the start and end time within the same vector and add, or concatenate, the encoded output to the input sequence.

We show the results in Table 3.7. In all cases, the final encoding is of the same dimension for comparable results. Performance is significantly worse with sinusoidal or learned positional encoding, as they are unable to capture the complexity of overlapping actions. There is also a drop when encoding only the centre of the time interval.

Separate-Add/Cat are alternative ways to encode the intervals (and hence include duration information) resulting in comparable performance to the interval counterparts. Our proposed approach to encoding the interval into the MLP shows the best performance for visual and while maintaining strong auditory performance.

3.5 Conclusions

In this paper, we propose to utilise the action's time interval as a query to an audio-visual transformer which learns to recognise the action from its interval and the unaltered surrounding context. We jointly train the model on modality-specific time intervals and label sets, allowing the Time Interval Machine (TIM) to recognise multiple events across both visual and auditory modalities.

TIM is sensitive to the interval's position and duration. This allows the model, as is, to produce competitive results on action detection through multi-scale dense querying.

3.6 Appendix

3.6.1 Further analysis of time intervals – scaling

We show the effect of shifting the time interval queries from their correct action interval in Section 3.4.4 (Fig. 3.7). In Fig. 3.8, we show the analogous figure as we vary the effect of *scaling* a centralised query from the ground truth. Similar to shifting, we also demonstrate a decrease in performance when scaling a query. Performance drops from 57.5% to 54.9% when contracting, and to 55.3% when expanding the query in visual queries. In audio, we see a drop from 58.3% to

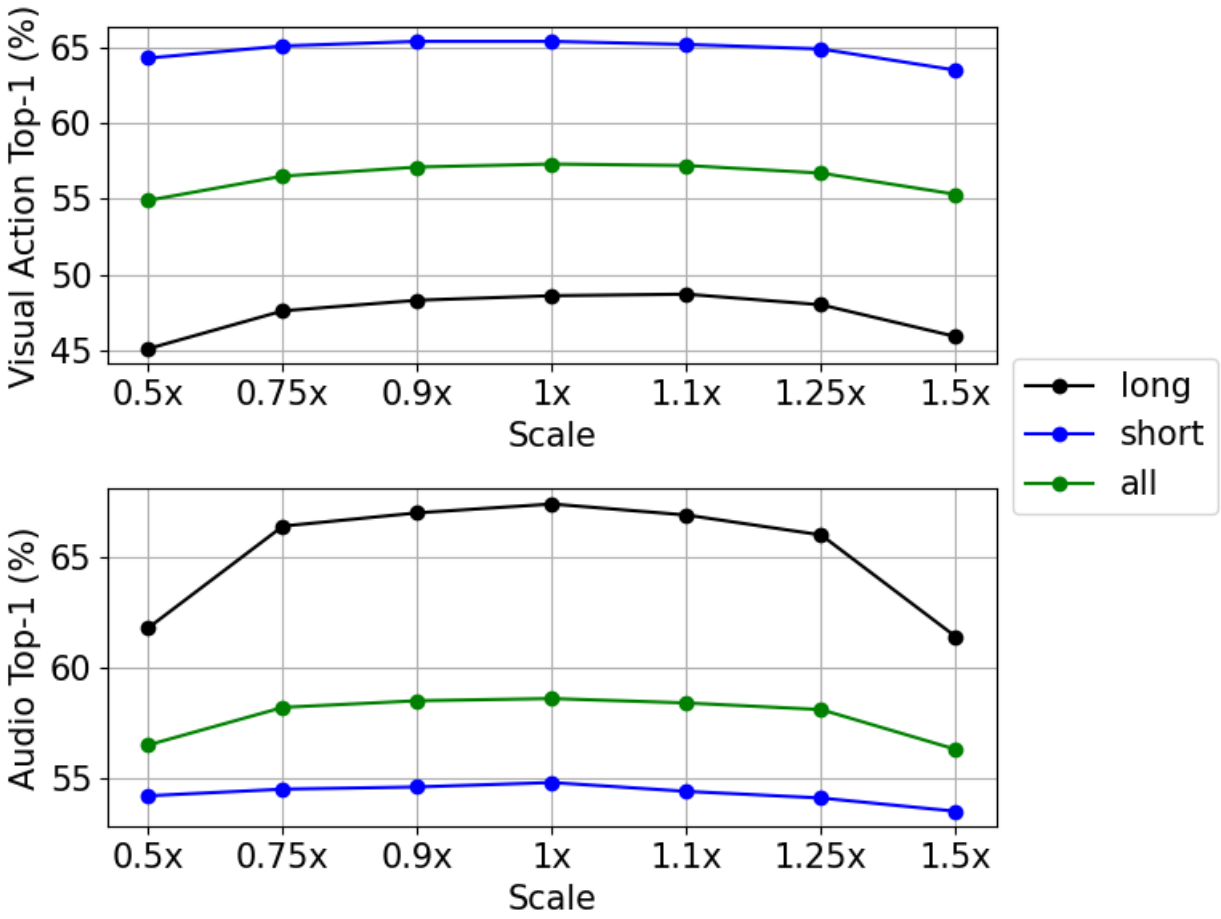


Figure 3.8: The impact of scaling a query centered around the action on both visual and audio performance. A shift of 0.0 sec, or a 1x scale, means querying with original time interval. Both visual and audio performance fall gradually as the query moves away from the original time interval.

56.5% when contracting, and to 56.3% when expanding the query.

Both Fig. 3.7 and Fig. 3.8 combined showcase the ability of TIM to correctly model the time interval of actions. The performance drops steadily, yet smoothly, as queries are changed from the ground truth – whether shifted or scaled.

3.6.2 Test Set Results

In this section we showcase TIM's results on multiple challenges and test sets across EPIC, namely EPIC-KITCHENS-100 recognition, EPIC-Sounds recognition, EPIC-KITCHENS-100 detection and EPIC-Sounds detection.

EPIC-KITCHENS-100 Test Set

In the main paper, TIM is evaluated on the EPIC-KITCHENS-100 validation set, as most state-of-the-art results only report on the validation set, and thus we do

the same for a direct comparison. Here, we evaluate the same model on the test set, by submitting to the leaderboards.

We report the results of our best performing model in Table 3.8. We ensemble six TIM models with input window lengths $W = 15, 30, 36, 40, 45, 60$ seconds with weights $[1.0, 0.9, 0.9, 0.9, 0.9, 0.9]$ respectively. All other parameters/architecture details remain unchanged. Our model achieves SOTA action performance (which ranks the winners) as well as verb performance. TIM remains behind SOTA on noun performance by 0.6%. We also report a single model TIM, without ensembling, and show this is competitive with winners from previous years despite using only a single model. We showcase the submission ranked top on the test set leaderboard in Fig. 3.9.

We also provide results for detection in Table 3.9. Note that this challenge also requires action predictions i.e. a combination of a verb prediction and noun prediction. To achieve this, we combine the predictions of each query from our verb and noun model, resulting in a two-stream architecture. We then follow [C.-L. Zhang et al. 2022] and re-weight the confidence and action boundaries of each proposal by:

$$\begin{aligned} \mathbf{p}(action) &= \mathbf{p}(verb)^{\alpha}\ \mathbf{p}(noun)^{(1-\alpha)}, \\ \mathbf{d}(action) &= \omega\mathbf{d}(verb) + (1-\omega)\mathbf{d}(noun) \end{aligned} \tag{3.6}$$

where $\alpha = 0.45$ and $\omega = \mathbf{p}(verb)/(\mathbf{p}(verb) + \mathbf{p}(noun))$. We can see that TIM sets a new SOTA in noun and action detection by 3.1 and 1.7 mAP respectively, only falling slightly behind on verb. For this method we ensemble 6 models using context windows $W = 15, 30, 45$ for both verb and noun streams. Evidence of our new SOTA method is shown in Fig. 3.10.

EPIC-Sounds Test Set

Here, we evaluate TIM on the test set by submitting to leaderboards. Again, we showcase results for both single and ensemble models in Table 3.10, using the same configuration described previously in the EPIC-KITCHENS-100 Action Recognition Challenge. Our model achieves SOTA performance across all metrics.

Method	Ensemble	Verb	Noun	Action
ctai	✓	69.4	63.3	50.0
hrgdscs	✓	71.0	61.3	50.4
Jaesung	✓	70.6	63.9	52.3
xxiong	✓	70.9	**66.2**	52.8
TIM (ours)	✗	73.1	64.1	53.0
yzhao	✓	71.7	65.8	54.3
TIM (ours)	✓	**73.8**	65.6	**54.5**

Table 3.8: Comparisons to state-of-the-art *recognition* models on EPIC-KITCHENS test set. We report the top-1 accuracy for verb, noun and action (%).

Test set

#	User	Entries	Date of Last Entry	Team Name	SLS			Top-1 Accuracy (%)			Top-5 Accuracy (%)			Unseen Participants Top-1 (%)			Tail Classes Top-1 (%)		
					PT	TL	TD	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
1	TIM_method	3	03/28/24	Oxford+Bristol	2.0 (4)	3.0 (2)	3.0 (2)	73.8 (1)	65.6 (3)	54.5 (1)	90.4 (6)	84.4 (4)	69.6 (5)	67.8 (1)	59.5 (2)	45.4 (1)	38.5 (3)	39.6 (4)	28.1 (3)
2	yzhao	5	06/01/23	UTDL	4.0 (2)	3.0 (2)	3.0 (2)	71.7 (2)	65.8 (2)	54.3 (2)	81.7 (21)	77.0 (14)	65.2 (11)	64.9 (3)	58.5 (3)	44.5 (3)	37.1 (5)	40.0 (3)	28.2 (2)
3	xxiong	8	06/01/22	Google Research Grenoble	5.0 (1)	3.0 (2)	4.0 (1)	70.9 (4)	66.2 (1)	52.8 (3)	91.1 (4)	86.8 (1)	71.7 (1)	64.5 (4)	61.6 (1)	44.8 (2)	39.7 (1)	43.7 (1)	28.5 (1)
4	Jaesung	17	06/01/23		2.0 (4)	3.0 (2)	4.0 (1)	70.6 (6)	63.9 (4)	52.3 (4)	89.8 (8)	84.6 (3)	71.3 (2)	64.9 (2)	57.9 (4)	43.9 (4)	35.0 (9)	36.7 (6)	25.4 (4)
5	hrgdscs	3	05/29/22		2.0 (4)	3.0 (2)	4.0 (1)	71.0 (3)	61.3 (7)	50.4 (5)	91.2 (2)	83.8 (5)	70.7 (3)	63.9 (5)	54.8 (8)	41.2 (6)	37.5 (4)	33.9 (8)	24.8 (6)
6	ctai	8	06/01/23	ctai	2.0 (4)	3.0 (2)	4.0 (1)	69.4 (8)	63.3 (5)	50.0 (6)	87.6 (17)	83.1 (7)	68.5 (7)	62.8 (9)	57.1 (6)	42.0 (5)	39.1 (2)	42.7 (2)	24.7 (7)

Figure 3.9: Screenshot of the EPIC-KITCHENS Action Recognition leaderboard (March 2024) showcasing our TIM_method ranked top.

Our single model does not perform quite as well as the visual counterpart in top-1 accuracy, but still outperforms all other methods with regards to mean average precision and per-class accuracy. It is also worth noting that the model selection was visually biased i.e. we chose the best performing visual model, instead of audio. Again, we showcase the submission ranked top on the test set leaderboard in Fig. 3.11.

We also provide detection results in Table 3.11, where we convincingly outperform the ActionFormer baseline across all metrics, notably by 4.2 mAP, setting a new

Test Set (Mean Average Precision - mAP)

#	User	Entries	Date of Last Entry	Team Name	SLS			mAP@0.1 (%)			mAP@0.2 (%)			mAP@0.3 (%)			mAP@0.4 (%)			mAP@0.5 (%)			Avg. mAP (%)		
					PT	TL	TD	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
1	TIM_method	1	04/06/24	Oxford+Bristol	2.0 (1)	3.0 (1)	3.0 (2)	32.14 (1)	34.88 (1)	28.13 (1)	30.01 (2)	32.99 (1)	26.74 (1)	27.84 (2)	30.57 (1)	25.01 (1)	25.24 (2)	26.60 (1)	22.29 (1)	20.37 (3)	21.78 (1)	18.86 (1)	27.12 (2)	29.36 (1)	24.21 (1)
2	mzs	5	05/27/23	mzs	2.0 (1)	3.0 (1)	4.0 (1)	31.01 (2)	30.32 (3)	25.54 (2)	30.04 (1)	28.76 (3)	24.54 (2)	28.01 (1)	27.20 (2)	23.16 (2)	25.44 (1)	24.28 (2)	21.04 (2)	22.32 (1)	20.74 (2)	18.35 (2)	27.36 (1)	26.26 (2)	22.52 (2)
3	lijun	18	06/01/22		2.0 (1)	3.0 (1)	4.0 (1)	30.67 (3)	30.96 (2)	24.57 (3)	29.40 (3)	29.36 (2)	23.50 (3)	26.81 (3)	26.78 (3)	21.94 (3)	24.34 (3)	23.27 (3)	19.65 (3)	20.51 (2)	18.80 (3)	16.74 (4)	26.35 (3)	25.83 (3)	21.28 (3)

Figure 3.10: Screenshot of the EPIC-KITCHENS Action detecion leaderboard (April 2024) showcasing our TIM_method ranked top.

Method	Average Precision (AP)						
	Task	@0.1	@0.2	@0.3	@0.4	@0.5	Avg.
lijun	Verb	30.7	29.4	26.8	24.3	20.5	26.4
	Noun	31.0	29.4	26.8	23.3	18.8	25.8
	Action	24.6	23.5	21.9	19.7	16.7	21.3
mzs	Verb	31.1	28.0	26.5	**25.4**	**22.3**	**27.3**
	Noun	30.3	28.8	27.2	24.3	20.7	26.3
	Action	25.5	24.5	23.2	21.0	18.4	22.5
TIM	Verb	**32.1**	**30.0**	**27.8**	25.2	20.4	27.1
	Noun	**34.9**	**33.0**	**30.6**	**26.6**	**21.8**	**29.4**
	Action	**28.1**	**26.7**	**25.0**	**22.3**	**18.9**	**24.2**

Table 3.9: Comparisons to state-of-the-art *visual action detection* models on the EPIC-KITCHENS test set. We report the average precision at IOU thresholds $[0.1, 0.2, 0.3, 0.4, 0.5]$ as well as their average across all thresholds on verb, noun and action.

Method	Ensemble	Top-1 Acc.	PCA	mAP
TIM (ours)	✗	54.9	22.8	31.9
Yuqi_Li	✓	55.1	21.0	26.2
audi666	✗	55.1	21.1	26.0
stevenlau	✗	55.4	21.8	27.0
TIM (ours)	✓	**55.9**	**23.0**	**32.2**

Table 3.10: Comparisons to state-of-the-art *audio recognition* models on EPIC-Sounds test set. We report the top-1 accuracy for audio interactions, along with the per-class accuracy (PCA) and mean average precision (mAP).

SOTA in this challenge.

3.6.3 Ablation studies

This section contains ablation studies on the proposed TIM architecture in various aspects and loss functions. We perform all ablations on the EPIC-KITCHENS (visual action recognition) and EPIC-SOUNDS (audio actions recognition). In all tables, we highlight our main reported results in grey.

Number of encoder layers. Here, we ablate the number of transformer encoder layers in TIM, varying from 1 to 6, on the performance. Other hyperparameters and model configuration remain fixed, as described in our main paper. Table 3.12 shows the result.

The best visual action performance is obtained by using four layers, while verb and noun performance is comparable to the models with only three layers. Interestingly, audio performance is best when using three layers. This is likely due

Test set												
#	User	Entries	Date of Last Entry	SLS			Training Modality	Top-1 Accuracy (%)	Top-5 Accuracy (%)	Per-Class Accuracy (%)	Mean Average Precision (%)	Mean Area Under Curve
				PT ▲	TL ▲	TD ▲	T_MOD ▲	Interaction ▲	Interaction ▲	Avg. ▲	Avg. ▲	Avg. ▲
1	TIM_method	1	04/06/24	2.0 (1)	2.0 (2)	3.0 (2)	2.0 (1)	55.86 (1)	86.26 (1)	22.97 (1)	32.23 (1)	0.894 (1)
2	stevenlau	6	05/31/23	2.0 (1)	3.0 (1)	4.0 (1)	0.0 (2)	55.43 (2)	85.52 (3)	21.84 (3)	26.98 (2)	0.877 (2)
3	audi666	3	06/01/23	2.0 (1)	3.0 (1)	4.0 (1)	0.0 (2)	55.11 (3)	85.40 (4)	21.14 (4)	25.96 (5)	0.856 (4)
4	Yuqi_Li	5	06/01/23	2.0 (1)	3.0 (1)	4.0 (1)	0.0 (2)	55.07 (4)	85.61 (2)	20.95 (5)	26.20 (3)	0.859 (3)

Figure 3.11: Screenshot of the EPIC-Sounds Audio-Based Interaction Recognition leaderboard (April 2024) showcasing our TIM_method ranked top.

Method	**Average Precision (AP)**					
	@0.1	@0.2	@0.3	@0.4	@0.5	Avg.
ActionFormer Baseline	9.6	8.5	7.4	6.2	5.1	7.4
TIM	**15.7**	**13.3**	**11.4**	**9.3**	**7.3**	**11.4**

Table 3.11: Comparisons to state-of-the-art *audio detection* models on the EPIC-Sounds test set. We report the average precision at IOU thresholds $[0.1, 0.2, 0.3, 0.4, 0.5]$ as well as their average across all thresholds.

to overfitting of the audio input compared to the visual. It is well known that multimodal training is susceptible to differences between the two modalities [W. Wang et al. 2020]. However, our training regime remains relatively stable between the two modalities. The difference between the top performance audio (3 layer) and our reported results (4 layers) is only 1.0%.

Number of transformer heads. We also ablate the number of transformer heads. We experiment with 2, 4, 8 and 16, keeping other hyperparameters fixed. Table 3.13 shows the results of this ablation.

Peak visual and audio performance is obtained when using 8 heads. This is the performance we report in the main paper. Interestingly, changing the number of

Test set													
#	User	Entries	Date of Last Entry	SLS			Training Modality	mAP@0.1 (%)	mAP@0.2 (%)	mAP@0.3 (%)	mAP@0.4 (%)	mAP@0.5 (%)	Avg. mAP (%)
				PT ▲	TL ▲	TD ▲	T_MOD ▲	Interaction ▲	Interaction ▲	Interaction ▲	Interaction ▲	Interaction ▲	Interaction ▲
1	TIM_method	1	04/06/24	2.0 (1)	3.0 (1)	3.0 (1)	2.0 (1)	15.71 (1)	13.27 (1)	11.36 (1)	9.34 (1)	7.30 (1)	11.40 (1)
2	EPIC_ACTIONFORMER	1	03/01/24	2.0 (1)	3.0 (1)	3.0 (1)	0.0 (2)	9.57 (2)	8.51 (2)	7.38 (2)	6.22 (2)	5.05 (2)	7.35 (2)

Figure 3.12: Screenshot of the EPIC-Sounds Audio-Based Interaction Detection leaderboard (April 2024) showcasing our TIM_method ranked top.

	EPIC-KITCHENS			EPIC-SOUNDS
Depth	**Verb**	**Noun**	**Action**	**Audio Actions**
1 Layer	75.8	65.0	55.4	58.4
2 Layers	76.5	66.2	56.5	58.4
3 Layers	77.0	66.9	57.2	**59.3**
4 Layers	**77.1**	**67.2**	**57.5**	58.3
5 Layers	76.6	66.7	56.9	58.2
6 Layers	76.9	66.6	57.0	57.5

Table 3.12: Effect of changing the number of transformer encoder layers. Number of transformer head is fixed to 16. The highlighted row is the performance we report in the main paper.

	EPIC-KITCHENS			EPIC-SOUNDS
# Head	**Verb**	**Noun**	**Action**	**Audio Actions**
2	77.0	65.9	56.6	**58.3**
4	76.7	66.7	56.9	57.9
8	**77.1**	**67.2**	**57.5**	**58.3**
16	77.0	**67.2**	57.1	58.1

Table 3.13: Effect of changing the number of heads in transformer. Number of transformer layer is fixed to 4. The highlighted row is the performance we report in the main paper.

heads has a comparable impact on performance to that when changing the number of layers reported in Table 3.12.

Temporal distance regression head architecture. We also ablate the structure of the Temporal distance regression head $h_{\tilde{t}}$ in Eq. (3.4), by varying the number of layers from 1 to 4. Results are shown in Table 3.14. The results are similar across all depths, but we find using 3 layers gives the best compromise across all metrics and these are the results we report in the paper.

Input length and feature density. We set $W \geq 10$ seconds. These long segments from untrimmed videos are complex and contain multiple overlapping actions. For example in EPIC-KITCHENS-100, a 30 second window contains on average 16 audio-visual annotated events with a maximum of 81 queries in the training set. Additionally, 28.1% of all actions overlap.

Table 3.15 shows the effect of changing the input visual and audio features for TIM. We experiment with the window size W, which is affected by the number of features within the window (N^m) and the stride between the features (H_f). We also experiment with the window stride (H_w) which affects how many windows

	EPIC-KITCHENS			EPIC-SOUNDS
Depth	**Verb**	**Noun**	**Action**	**Audio Actions**
1 Layer	77.0	66.8	57.3	58.1
2 Layers	**77.2**	66.9	56.9	58.4
3 Layers	77.1	**67.2**	**57.5**	58.3
4 Layers	76.8	66.9	**57.5**	**58.7**

Table 3.14: Effect of temporal distance head structure. The highlighted row is the performance we report in the main paper.

fit within an entire untrimmed video, and hence the extent of the temporal context around a given action. We separate the table into 4 sections, separated by horizontal lines, to showcase different variations.

First we ablate on the number of features, while keeping the feature hop size constant. Increasing the number of features will increase the window size. We see that using 50 features with a stride of 0.6 seconds works best, resulting in a window size of 30 seconds. This time frame likely provides enough relevant context to the action without injecting redundant information through additional features too distant from the action.

We then ablate the feature stride while keeping the number of features constant. In this case, a larger hop size results in a larger input window. We see that a stride of 0.6 seconds, resulting in a 30 second window, performs the best. This outperforms a 30 second window with 75 features with stride 0.4 seconds, as the sparser sampling likely removes redundant information.

We also experiment with the feature density, by fixing the window size to 30 seconds, but varying both the number of features and the feature stride. In this case, we see that our proposed feature density of $N^m = 50$ performs best. Increasing the number of features increases redundancy, whereas a sparser number does not benefit from sufficient neighbouring context.

Finally, we experiment on the stride of the input window. A smaller stride results in increased overlap between the input features. Compared to the stride of 1.0, used in our results, an increased stride clearly drops visual performance.

Time Interval MLP structure. We also ablate the structure of the Time Interval MLP $I(.)$. We experiment with varying the number of linear layers. As shown in Table 3.16, TIM seems to favour a depth of 3 within the Time Interval

				EPIC-KITCHENS			EPIC-SOUNDS
W	**N^m**	**H_f**	**H_w**	**Verb**	**Noun**	**Action**	**Audio Actions**
15.0	25	0.6	1.0	76.8	67.0	57.3	**59.0**
45.0	75	0.6	1.0	76.6	67.1	57.0	57.4
60.0	100	0.6	1.0	76.5	66.8	57.1	57.3
10.0	50	0.2	1.0	76.2	66.1	55.9	58.4
20.0	50	0.4	1.0	76.7	66.7	56.8	58.7
30.0	50	0.6	1.0	**77.1**	**67.2**	**57.5**	58.3
40.0	50	0.8	1.0	76.5	66.8	56.8	58.0
50.0	50	1.0	1.0	75.5	65.9	56.2	56.5
30.0	25	1.2	1.0	76.5	66.1	56.4	57.3
30.0	75	0.4	1.0	76.8	66.5	57.3	58.0
30.0	50	0.6	2.0	76.7	66.8	57.2	58.7
30.0	50	0.6	5.0	76.4	66.1	56.4	58.6
30.0	50	0.6	10.0	75.5	65.4	55.6	57.6

Table 3.15: Effect of changing the parameters to alter the feature input to TIM in EPIC-KITCHENS and EPIC-SOUNDS. **W**: Window Size in seconds, **N^m**: Number of features, **H_f**: Feature Stride in seconds, **H_w**: Window Stride in seconds.

	EPIC-KITCHENS			EPIC-SOUNDS
Depth	**Verb**	**Noun**	**Action**	**Audio Actions**
1 Layer	75.5	66.3	56.0	57.2
2 Layers	76.6	66.5	56.5	57.8
3 Layers	**77.1**	**67.2**	**57.5**	58.3
4 Layers	76.5	66.9	57.3	58.0
5 Layers	76.6	67.0	57.2	**58.4**

Table 3.16: Effect of the Time Interval MLP $I(\cdot)$ structure. The highlighted row is the performance we reported in the main paper.

MLP, benefiting from a 1.0% visual and 0.5% audio boost over 2 layers.

Loss ablation. We experiment with varying the λ^{td}, as well as the within-modal and cross-modal sampling variants (Eq. (3.4)). The results are shown in Table 3.17. Introducing the Temporal Distance loss ($\lambda^{td} > 0$) improves the overall performance for visual, but has an adverse effect on audio. We also observe that $\lambda^{td} = 0.3$ with cross-modal sampling shows the highest performance on the visual action metric. The *cross-modal* sampling strategy shows marginally improved results than the *within-modal* strategy for visual, suggesting the distance loss is more beneficial for video than for audio.

		EPIC-KITCHENS			EPIC-SOUNDS
λ^{td}	**Sampling**	**Verb**	**Noun**	**Action**	**Audio Actions**
0.0	-	76.9	66.7	57.2	**58.4**
0.1	*cross-modal*	77.0	66.7	57.1	58.1
0.3	*cross-modal*	77.1	**67.2**	**57.5**	58.3
0.3	*within-modal*	**77.3**	67.0	57.4	**58.4**
0.5	*cross-modal*	76.9	66.8	57.3	58.2

Table 3.17: Effect of Temporal Distance loss on performance. **Sampling** represents the two different ways of sampling pairs $\mathbb{B}$, *cross-modal* means sampling the pairs across modalities and *within-modal* indicates sampling pairs only within the same modality. We report the highlighted row in the main paper.

3.6.4 TIM for Detection

In this section we describe how we adapted TIM for the task of action detection for the results reported in Table 3.3. The backbone remains largely unchanged from the recognition task. However, there are differences in how we obtain queries, as well as an additional interval regression head.

Multi-scale Queries for Detection

While in recognition we can utilise the ground-truth timestamps of the actions to query the input, in detection, we obtain dense *proposal* queries by constructing a query pyramid. These queries cover multiple fixed-size scales, spanning the entirety of the long video at each level, starting from short, dense temporal interval queries to long ones. The pyramid structure allows the model to classify and regress to both long and short actions within the input.

In practice, when constructing our query pyramid, we start from a query interval size of $0.005 * W$ (0.15s for a $W = 30$s window), with dense queries that span the entire window. We then double the query size in the next layer, again spanning the entire window at this resolution, and repeat this process stopping before the query size matches or exceeds the full window size. For a 30s window, this method constructs a query pyramid consisting of 8 layers, with resolutions: [0.15s, 0.3s, 0.6s, 1.2s, 2.4s, 4.8s, 9.6s, 19.2s].

We classify these queries in the same manner as recognition. However, we also introduce a regression head, which predicts the start and end times of the action the query is assigned to. The regression head allows for temporal localisation to be improved over that of the proposal interval and to have greater overlap with

the ground truth.

When obtaining the final sets of detections, we classify and regress all queries in the pyramid across all input windows in the untrimmed video. We then threshold predictions that are below a confidence threshold. We then apply class-dependent Soft-NMS [Bodla et al. 2017] to the filtered predictions to remove highly overlapping proposals, before calculating the precision scores.

Detection Training

During training, we deem any query in the fixed pyramid (multi-scale) set of queries with temporal $IOU \geq 0.6$ with any ground truth action as a positive query. If a query has a temporal overlap above the threshold with multiple ground-truth actions, we only consider the action label with the highest IOU. For all positive queries, we directly predict the assigned action's start and end times (t_s^m, t_e^m) and classify the corresponding action label. For negative queries, we do not regress the interval's duration and set the label as a zero-vector for across all classes e.g. background.

As with recognition, we classify all queries with $h^m_{\texttt{CLS}}(\cdot)$ and obtain predictions $\hat{y}^m_{\texttt{CLS}} = h^m_{\texttt{CLS}}(Z^m_{\texttt{CLS}})$. To classify queries, we train TIM using a Sigmoid Focal Loss [T.-Y. Lin et al. 2020] $F(\cdot)$ to balance the positive and negative samples:

$$L^m_{\texttt{det\_CLS}} = \frac{1}{B}\sum^{B} F(\hat{y}^m_{\texttt{CLS}}, y^m_{\texttt{CLS}}) \tag{3.7}$$

For positive queries, we also feed the encoded `CLS` tokens through a separate regression head $h^m_{\texttt{REG}}$ to predict the querys associated ground truth action start and end time $(\hat{t}_s^m, \hat{t}_e^m) = h^m_{\texttt{REG}}(Z^m_{\texttt{CLS}})$. We train this via a DIOU regression loss [Z. Zheng et al. 2020]:

$$L^m_{\texttt{det\_REG}} = \frac{1}{Q_P}\sum^{Q_P} DIOU((\hat{t}_s^m, \hat{t}_e^m), (t_s^m, t_e^m)) \tag{3.8}$$

where Q_P is the number of positive queries. Finally, we combine both losses to form our detection loss:

$$L^m_{det} = L^m_{\texttt{det\_CLS}} + \lambda_{\texttt{det\_REG}} L^m_{\texttt{det\_REG}} \tag{3.9}$$

Where $\lambda_{\texttt{det\_REG}}$ is a parameter used to weight the regression loss. We set this to 0.5.

3.6.5 Further Implementation Details

Feature Extraction The Omnivore model used is pre-trained with ImageNet [Russakovsky et al. 2015], Kinetics [Kay et al. 2017] and SUN RGB-D [S. Song et al. 2015] datasets. For EPIC experiments, we finetune the model with EPIC-KITCHENS100 visual labels. The VideoMAE-L features are pre-trained on Kinetics [Kay et al. 2017], Something-Something V2 [R. Goyal et al. 2017], AVA [Gu et al. 2018] and WebVid2M, which we also fine-tuned on EPIC-KITCHENS visual labels. The detailed training procedure for Omnivore is available in [Girdhar et al. 2022] and for VideoMAE in [Tong et al. 2022; Y. Wang et al. 2022]. We extract dense features that overlap, so that we can use fine-grained time intervals as a query. Each 1 second Omnivore feature is computed by feeding 32 frames the temporal segment sampling described in [Z. Liu et al. 2022] and whereas we feed 16 frames using the sampling described in [Tong et al. 2022; Y. Wang et al. 2022] for each VideoMAE feature.

For Auditory SlowFast [Kazakos et al. 2021b], we utilise the pre-trained VGGSound [H. Chen et al. 2020] model and change the input length from 2 seconds to 1 second to match the temporal extent of the visual features. Only for EPIC experiments, we finetune the model with EPIC-SOUNDS audio labels. The additional sets used for data augmentation apply SpecAugment with two frequency masks with F = 27 and two time masks with T = 25. Again, this enables data augmentation for audio.

For AVE visual features, we use a VGG-19 [Simonyan and Zisserman 2014] model pre-trained on ImageNet [Russakovsky et al. 2015]. We extract the features from *pool5* layer on VGG-19 to get a spatial feature map per each frame. We average these feature maps per each second by global pooling. For audio features, we adopt a VGG-like [Hershey et al. 2017] network pre-trained on AudioSet [Gemmeke et al. 2017]. Both visual and audio feature cover one second of the visual or audio stream. Additionally, due to the significantly smaller size of the AVE dataset, we reduce the model size for this dataset by halving the hidden dimension of all linear layers (512-D) and applying channel-wise dropout with $p = 0.1$ to the raw input

features, but retain a dropout of $p = 0.5$ on the encoded transformer input.

Model selection scheme. For datasets with distinct visual and audio label sets (EPIC and Perception Test), we train a single model on both sets of labels simultaneously. In these cases, we report results across all metrics in both modalities for the epoch with the best visual performance. We note that we can obtain additional audio performance by adjusting hyperparamters (such as λ^a) to be more biased towards audio. However, when reporting results, we take the audio performance from our best performing visual model, reporting a single model for audio-visual TIM.

EPIC Details. For EPIC-KITCHENS-100 and EPIC-SOUNDS, we include two extra CLS tokens for each visual query: $\texttt{[CLS]}^v_{verb,noun}$, along with classifiers $h^v_{\texttt{CLS}_{verb}}(\cdot)$ and $h^v_{\texttt{CLS}_{noun}}(\cdot)$. We set the learning rate to 1e-4 and apply channel-wise dropout with $p = 0.5$ directly to the raw input features, as well as to the encoded transformer input. We set $\lambda^a = 0.01$ and $\lambda^v = 1.0$. The low value of λ^a is to alleviate early overfitting of the audio data, also observed in other works [Xiao et al. 2020].

AVE Details. Due to the significantly smaller size of the AVE dataset, we reduce the model size for this dataset by halving the hidden dimension of all linear layers (512-D). We use an initial learning rate $5e-4$. We set all dropouts in the model to $p = 0.1$. We set $N^m = 10$ with $H_f = 1.0$ to be consistent with other works. This results in a window size of $W = 10$ seconds which is the full-length of the video in this dataset. We thus do not use any window stride (H_w) for this dataset. We apply AVGA [Tian et al. 2018] to spatial visual features from VGG-19 before feeding it to the transformer. As this dataset does not contain distinct labels for audio and visual, we encourage the model to learn the single label set for both modalities by duplicating the query, i.e. using a `[CLS]` for each modality, and combine their logits for training and inference. We set $\lambda^a = 1.0$ and $\lambda^v = 1.0$.

Perception Test Details. We set the learning rate to 1e-4 and apply channel-wise dropout with $p = 0.1$ to both the raw input features and encoded input sequence. We set $W = 20$ seconds, $\lambda^a = 1.0$, and $\lambda^v = 1.0$.

Detection Details. Due to memory constraints, as opposed to using a single model to jointly train for all sub-tasks in recognition (visual and audio or verb,

noun, action and audio in EPIC), we use a separate model for each individual sub-task, resulting in two different sets of model weights for detection and recognition. We also extend the number of layers in the transformer encoder from 4 to 6. The regression head consists of 2 layers with hidden dimension $D/2$, followed by ReLU activations, and a final layer which outputs 2 numbers relating to the regressed boundaries, followed by a Sigmoid activation to scale the outputs between $[0, 1]$.

For Perception Sound and Action, we train for 100 epochs and use a 0.01 confidence threshold and NMS $\sigma = 0.1$. For EPIC, we train for 35 epochs and use a 0.03 confidence threshold and NMS $\sigma = 0.25$. All other hyper-parameters are consistent with the recognition models.

Part II

Audio-visual character-aware subtitle generation for TV shows

Chapter 4

Look, Listen and Recognise: character-aware audio-visual subtitling

The paper has been accepted for publication at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2024.

Look, Listen and Recognise: Character-Aware Audio-Visual Subtitling

Bruno Korbar* Jaesung Huh* Andrew Zisserman

VGG, University of Oxford

Abstract

The goal of this paper is automatic character-aware subtitle generation. Given a video and a minimal amount of metadata, we propose an audio-visual method that generates a full transcript of the dialogue, with precise speech timestamps, and the character speaking identified. The key idea is to first use audio-visual cues to select a set of high-precision audio exemplars for each character, and then use these exemplars to classify all speech segments by speaker identity. Notably, the method does not require face detection or tracking. We evaluate the method over a variety of TV sitcoms, including Seinfeld, Fraiser and Scrubs. We envision this system being useful for the automatic generation of subtitles to improve the accessibility of the vast amount of videos available on modern streaming services. Project page : https://www.robots.ox.ac.uk/~vgg/research/look-listen-recognise/

*Equal technical contribution.

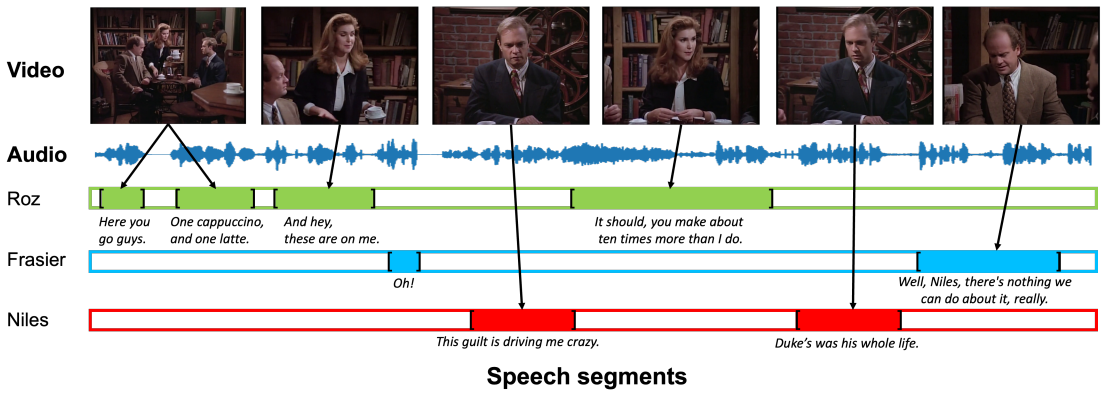


Figure 4.1: Character-aware audio-visual subtitling. The generated data covers *what* is said, *when* it said, and by *whom* it is said.

4.1 Introduction

With the rise of streaming platforms that allow watching videos "on-demand", more video content is made available to the general public and researchers than ever in history. With more than 80% of users of one such platform relying on subtitles [*Netflix player control tests* n.d.], automatic subtitle generation and captioning has become an important research topic in the community [Radford et al. 2023; Bain et al. 2023]. Unfortunately, many subtitles, whether automatically generated or not, do not comply with the standards for Subtitles for Deaf and Hard-of-hearing (SDH): namely, they do not include information about speaker identification, nor do they contain sound effects and music.

In this paper, we take the next step towards automatic generation of SDH – we aim to make the subtitles character-aware. Character-aware subtitles would also be of great benefit to researchers. They would allow for the automatic generation of large-scale video datasets, which could fuel the next generation of visual-language models capable of learning higher-level semantics from the paired data.

There has been a plethora of works using audio-visual networks for speech recognition [Afouras et al. 2019a; Shi et al. 2022b], speaker diarisation [J. S. Chung et al. 2020c; Ding et al. 2020; E. Z. Xu et al. 2022] or character recognition [Sharma and Narayanan 2022a; Everingham et al. 2009; Haurilet et al. 2016; Nagrani and Zisserman 2017] which are subtasks of our main goal. However, these works require additional processing for detecting and tracking faces. We present a simpler method that does not require face detection or tracking and uses only off-the-shelf deep neural network models and the cast list for each episode.

We make the following four contributions: (i) we propose a new task, character-aware audio-visual subtitling, which aims to generate the *what*, *when* and by *whom* for subtitles, with minimal required metadata. (ii) we develop an automatic pipeline for this task that does not require face detection or tracking (Section 4.2); (iii) we curate an evaluation dataset that includes subtitles labelled with characters individually for three different sitcom series: Fraiser, Scrubs and Seinfeld (Section 4.3); and (iv) we assess the method on the evaluation dataset and report the performance (Section 4.4).

4.1.1 Related work

Labelling people in videos. is a well studied topic in computer vision [Everingham et al. 2009; Haurilet et al. 2016; Nagrani and Zisserman 2017]. Often, the availability of various levels of prior information is required such as scripts [Everingham et al. 2009], clean images for actor-level supervision [Nagrani and Zisserman 2017], or ground truth subtitles with correct timestamps [Mocanu et al. 2019; Akahori et al. 2017]. [A. Brown et al. 2021a] relaxes the need for cleaned data and makes their method scalable by gathering a large amount of data via automated image search to obtain the corroborative evidence they use for supervision. Like [A. Brown et al. 2021a], our model retrieves the necessary information via search engines, however, it does not pre-process video frames, save for the transformations required by a neural network.

Audio-only speaker diarisation. Speaker diarisation is the task of identifying "who spoke when" from a given audio file with human speech. There are two branches of works in this area: (i) using existing Voice Activity Detection (VAD) and a speaker model together with clustering [Quan Wang et al. 2018; A. Zhang et al. 2019; Kwon et al. 2021] and (ii) using an end-to-end model which goes from the VAD to assigning speakers [Fujita et al. 2019; Horiguchi et al. 2020]. Both of them suffer when the number of speakers is large such as in TV shows or dramas. Furthermore, the current state-of-the-art speaker recognition models assume that the input is long (> 2 sec), while most of the speeches in TV shows are relatively short including exclamations, which leads to the degradation of speaker clustering performance. In this paper, we include the active speaker detection model and person-identification model, which are strong in short videos, to identify

the character.

Audio-visual speaker diarisation. In the last few years, efforts were made to improve the performance of diarisation by borrowing the power of face recognition models or lipsync models, which are closely related to human speech [J. S. Chung et al. 2020c; E. Z. Xu et al. 2022; J. S. Chung et al. 2019]. [J. S. Chung et al. 2020c] utilises audio-visual active speaker detection model and speech enhancement models, but mostly in celebrity interviews or news segment where the length of speeches are generally short. [E. Z. Xu et al. 2022] introduces an Audio-Visual Relation Network (AVR-Net) that leverages the cross-modal correlation to recognise the speaker's identity. Our approach is different from these works in two ways: (i) we do not use any face detection or tracking; and (ii) we introduce character-aware audio-visual subtitling that builds the character bank within each video and figures out not only the speaker clusters but the speakers' *identity* for each utterances and the speech content.

Datasets. The Bazinga! dataset [Lerner et al. 2022] also provides subtitles labelled with characters for a large number of TV series. However, it is an audio only dataset, and consequently is not directly suitable for applying the audio-visual approach we develop.

4.2 Method

This section explains our approach to creating subtitles for the video and attributing speakers to each speech segment. Our method consists of two distinct stages. First, we detect speech segments from the video, recognise the spoken words, and process the data to create a database of what we refer to as *speech exemplars* – sample video clips where a speaker is clearly audible, visible and identifiable. In the second stage, the speech exemplars for each character are used to assign the identities to *all* speech segments.

In order to label the characters we require the following metadata for each episode: (i) the names of the characters in the show; and (ii) for each character 1–10 sample images of the actor and their names that we can use as visual examples. This metadata can be obtained automatically from online database of movies or TV series [*International Movie Database* n.d.].

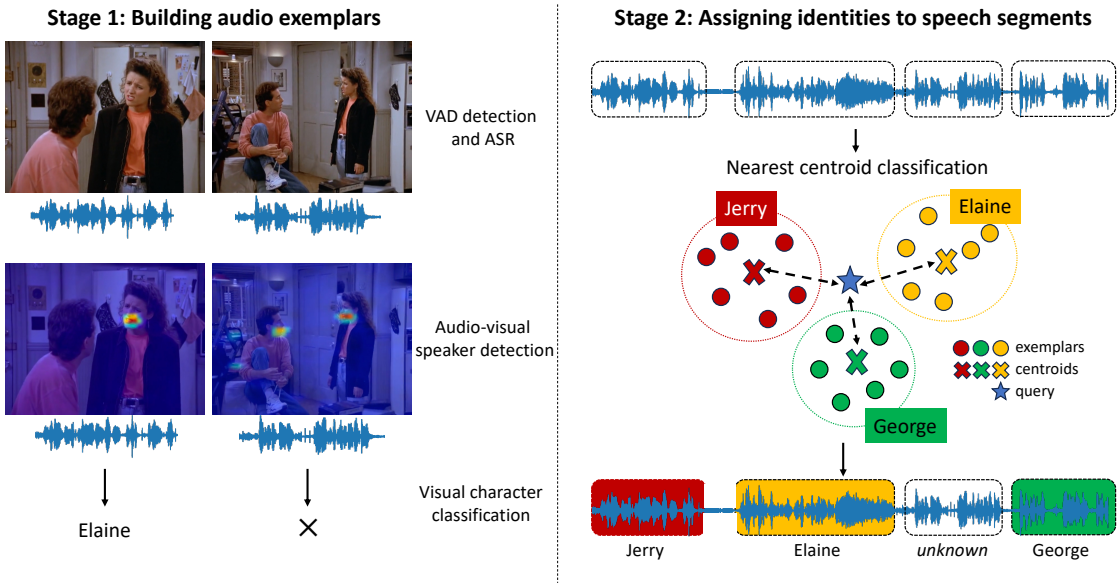


Figure 4.2: Overview of our method. We first build a database of audio exemplars for each character by filtering speech segments until only a high precision set remains (left). Each speech segment is then assigned to a character by comparing its voice embedding to the exemplar embeddings (right).

4.2.1 Stage 1: building audio exemplars

The goal of stage 1 is to create a database of character voices. We take multiple episodes of a TV series, and obtain a set of speech segments for each character.

In order to do this, we first split videos into speech segments, and transcribe them. For each segment we determine if only one speaker is visible and is speaking – a crucial step because it allows us to be confident that the speech segment corresponds to the face in the frame. We collect a set of speech segments for each character that we can confidently recognise from their face, and then filter the samples in each set to remove potential label noise using voice embeddings.

We end up with a set of speech segments for each character that are recognised with high precision, and refer to these as *speech exemplars*. The building of these exemplars is illustrated in Fig. 4.2, and we give details of each sub-step below.

1. VAD detection and Automatic Speech Recognition (ASR). In this stage, we take an entire video and split it into segments where speech is detected and recognised. We first detect the voice regions across the entire dataset and determine the spoken content of each segment. We do this with a language-guided VAD model. We apply the WhisperX [Bain et al. 2023] model on the audio stream of our dataset which detects the speech regions with word-level timestamps. We

concatenate the generated words to obtain the entire transcription per video, then use a sentence tokenizer to separate them by sentences. Assuming each sentence is spoken by a single speaker, we use the start and end times of the sentences as our unit of speech segments.

We also find that most TV shows contain laughter tracks (audience laughter) which are voice regions but are not of interest to this work. Thus, we run a pretrained laughter detector [Gillick et al. 2021] for each of the remaining voice segments and remove the ones from the candidates of exemplars if laughter is detected. After this step, we know precisely when characters in the show are speaking and what they are saying. We don't yet know *who* is saying what.

2. Audio-visual speaker detection.

The goal of this stage is to take speech segments from the previous stage and select only those with a single visible speaker. This will produce a subset of speech segments where we can recognise the speaker. To achieve this, we localise the speaker with an audio-visual synchronisation model [Afouras et al. 2020b] which produces a spatial location of the audible objects and has been shown to detect speakers well. In practice, it generates an audio-guided heatmap over each video frame. We average the heatmaps over the length of each speech segment to avoid unnecessary noise and detect peaks in the heatmap through a combination of maximum filtering and non-maximum suppression. Example heatmap outputs can be seen in Fig. 4.2. When a single peak is visible throughout the video clip, we can assume that only one speaker is present. If there are no detected peaks, or there are multiple ones, we discard that speech segment from the candidates of exemplars.

3. Visual character classification. In this step a character name is assigned to each of the single-speaker speech segments from the previous step where possible. This leaves us with a further reduced set of speech segments, each having a character name associated with it. Character classification is the only step in our annotation process that external data is used. Specifically, the 1–10 sample images of each actor are used to form a visual embedding of that character. Our classification model [Korbar and Zisserman 2022] compares a visual embedding of the frames of a speech segment to a combination of actor visual embedding and

actor name (details are given below). We select the best match or discard the clips which cannot be classified with a high degree of confidence. Note, (i) the comparison is at the frame level, no face detector or cropping is required for this visual recognition; (ii) we compute visual embeddings for all characters in a given season, but only consider ones present in that episode at inference time.

4. Audio filtering. Finally, we group the labelled speech segments from the previous stage by character, and for each character we filter their *voice* samples to remove potential noise from the groupings as follows: we compute voice embeddings for each sample, and consider that a sample is positive for a given character if its 5 nearest neighbours are labelled as the same character. Note that for characters where the number of samples n is smaller than 5, we keep all the samples in our database. This gives us the final exemplar set for a given TV series and hopefully leaves us knowing what each character sounds like.

4.2.2 Stage 2: Assigning characters to speech segments

The aim of this stage is to assign a character name to each of the detected audio segments that we are confident of, regardless of whether a speaker is visible or not. On a high-level, we achieve this by comparing the distance between each speech segment and the audio exemplars for each character. We do not assign an identity if the minimum distance is above a certain threshold.

Specifically, for each character we compute the mean of exemplar embeddings and use it as a centroid representation for that character. To classify speech segments, we embed them with the same model used to generate the exemplar embeddings, and measure distances to class centroids. The segment is assigned to the speaker corresponding to the nearest centroid. However, if the minimum distance between the segment embedding and each centroid is bigger than a threshold d, then that segment is classified as "unknown". This covers uncertainty and also the cases where we don't have exemplars.

4.2.3 Implementation details

We detect speech and perform ASR with an off-the-shelf WhisperX [Bain et al. 2023] model, and the sentences are tokenized with NLTK [Bird 2006] tokenizer.

We use the laughter detector by [Gillick et al. 2021] with a detection threshold of 0.8. All voice embeddings are encoded with ECAPA-TDNN [Desplanques et al. 2020], which is pretrained with VoxCeleb [Nagrani et al. 2019]. For discovery of speaking faces, we use a pretrained LWTNet [Afouras et al. 2020b]. For each generated heatmap we detect 4 peaks, and consider each a positive if it's larger than $\tau_{\text{det}} = 0.7$. For actor face classification, we use the CLIP-PAD model [Korbar and Zisserman 2022] pretrained on VGGFace and VGGFace2 [Parkhi et al. 2015]. Actor text-image embeddings are formed as `"An image of <TKN> Name Surname"` where `<TKN>` is an average representation of query images computed using a face-embedding network, as in [Korbar and Zisserman 2022]. To classify the actors in the scene, we measure the cosine similarity between the visual embedding of the frames and the text-image embedding and choose the ones with highest similarity score where the score is over threshold $\tau_{\text{rec}} = 0.85$ as positives. All hyperparameters are determined via grid search on the three validation episodes, and kept fixed otherwise.

4.3 Evaluation Dataset

In this section, we describe a semi-automatic annotation pipeline used to generate the ground truth character names, timestamps and subtitles for speech segments. The goal is to annotate the identities for all subtitles with accurate time intervals in the video.

4.3.1 Annotation procedure

The dataset collection process consists of two stages: (i) automatic initial annotations by aligning a transcript with timed subtitles; and (ii) human annotators reviewing and further refining these annotations. Note that our dataset differs from other speaker diarisation datasets since we are also interested in the *identity* of each speaker and speech transcriptions.

Aligning transcripts and timestamps. To associate character names with corresponding temporal timestamps, we leverage two readily accessible source of textual video annotation: original transcripts and subtitles with word-level timestamps. Transcripts are obtained from multiple online sources [*The Frasier Archives*

Dataset	# episode	duration	# IDs	speech %	# spks
Seinfeld	6	2h 09m	36	60.6	6 / 9.2 / 12
Frasier	6	2h 11m	29	59.5	6 / 9.2 / 12
Scrubs	6	2h 02m	48	67.9	13 / 15.7 / 18

Table 4.1: Evaluation dataset statistics. # **episode**: number of episodes, **duration**: total duration of the dataset, #**IDs**: total number of characters, **speech %**: percentage of video time that is speech and # **spks**: min / mean / max of number of speakers per video.

2024; *Seinfeld scripts dot com* 2024; *Scrubs fandom* 2024]. They include spoken lines and information about who is speaking. However, they do not provide any timing information beyond the order in which the lines are spoken. We use WhisperX [Bain et al. 2023] to obtain the timed subtitles. We find this suitable since its transcription and timestamps are highly accurate, whereas the timestamps in subtitles from other online sources often do not align with the actual speech in the video. To align the original transcripts and timed subtitles, we employ the approach from [Everingham et al. 2006]. We use Dynamic Time Warping (DTW) to obtain the word-level alignment between the transcript and timed subtitles to associate the speaker with each of these words. Please refer to the original paper for the detailed process.

Manual correction. The output of the automatic pipeline is prone to several errors such as (i) a mismatch between the text of the transcript and WhisperX's transcription results; and (ii) mispredicted timestamps. We correct any errors in timestamps and character names manually using the VIA Video Annotator [Abhishek Dutta and Zisserman 2019].

4.3.2 Dataset statistics

Three TV series datasets are used to evaluate our method. We annotate the first six episodes of Season 2 of **Frasier**, Season 2 of **Scrubs** and Season 3 of **Seinfeld**. We utilise the sixth episode in each season as our validation set, while the remaining episodes serve as our test set. The detailed statistics are shown in Table 4.1.

Step	# of exemplars	% of total
VAD detection	2107	100.0
Audio-visual speaker detection	1271	60.3
Visual character classification	806	38.3
Audio filtering	407	19.3

Table 4.2: Exemplar yield after steps in Stage 1 (on Seinfeld).

Char. name	# exemplars	# correct	Acc (%)
Total	407	406	99.8
Jerry	273	272	99.6
Elaine	30	30	100
Kramer	12	12	100
George	14	14	100
others	78	78	100

Table 4.3: Exemplar recognition performance for named characters in Stage 1 in Seinfeld. 'others' is a group of 21 characters, all named correctly.

4.4 Results

This section provides a detailed analysis of Stage 1 and 2, followed by the overall result on our test set.

4.4.1 Detailed analysis of Stage 1 and 2

Performance evaluation of Stage 1. We evaluate the yield and classification accuracy of the speech exemplars on the five episodes of Seinfeld in our test set. In Table 4.2, it can be seen that **19.3%** of voice activity segments can be considered as exemplars. We also evaluate the performance quantitatively by manually inspecting the exemplars. The results, shown in Table 4.3, demonstrate that the accuracy of Stage 1 is almost perfect, being **100%** correct for most characters. There are 11 characters for which we have no exemplars in the 5 episodes of Seinfeld. They cover only 1.8% of speech segments – most of them speak less than five sentences in the episodes.

Performance evaluation of Stage 2. We demonstrate the trade-off between the Proportion of Classified Segments (POCS) and overall precision by varying the threshold d used in the nearest centroid voice classification to assign speech segments as "unknown". True positives are the segments that overlap with the ground truth segments and the character is correctly identified. Fig. 4.3 shows the result. It can be seen that precision decreases as we classify more segments. Also,

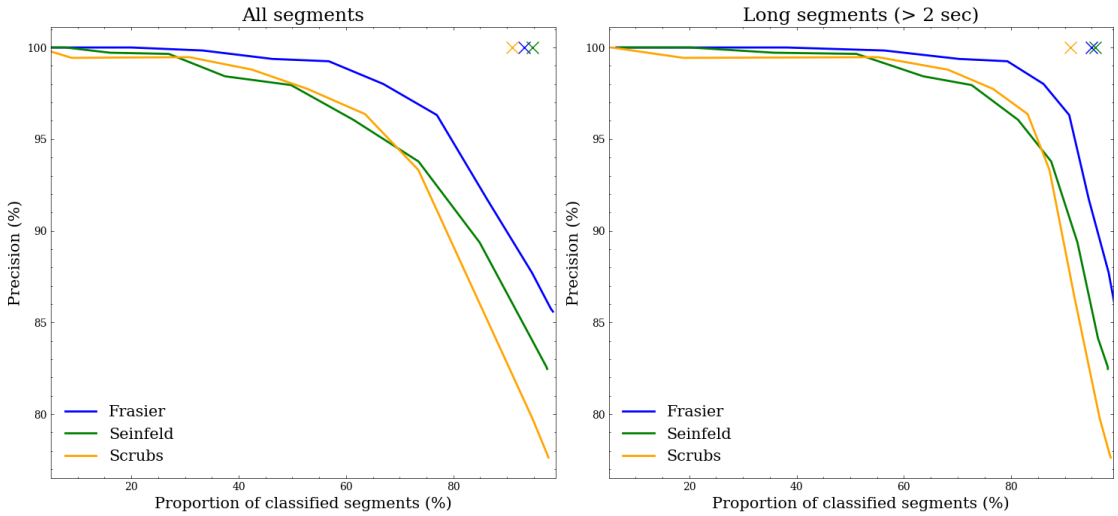


Figure 4.3: Stage 2 Precision-POCS Curves for the test set of the three TV series, obtained by varying the threshold d (for classification as "unknown"). The left figure shows the performance using all detected speech segments. The right figure shows the performance only for the long segments (> 2 sec). We also show the oracle points ('x' in each graph) for each TV series. The oracle point is where all segments for which there are character exemplars are correctly classified, and other segments are classified as "unknown".

long segments show higher precision in all three TV series at any given POCS, which shows that the speaker model produces better representations for longer segments.

4.4.2 Overall performance on the test set

Performance measures. In addition to the traditional diarisation metric of Diarisation Error Rate (DER), we report the overall character recognition accuracy as well as the average of the per-character precision and recall metrics for the characters of each show. We use a 0.25-second collar to calculate DER. Accuracy is calculated for the segments that overlap with one of the ground truth segments.

The results are given in Table 4.4. We can see that the model performs best on Frasier and worst on Scrubs in all metrics. This is due to the difference in size of the casts in each dataset. Scrubs has more characters than Frasier ($48 > 29$) for a similar total duration (see Table 4.1). Thus, Scrubs provides more potential assignments for each segment, making identification more challenging.

We also report the diarisation performance with and without the overlapping speech in Table 4.4. The difference in DER for these two categories is small in Seinfeld and Frasier, meaning that there is not much overlapping speech within

Showname	DER↓	DER(O)↓	Acc↑	Ppc↑	Rpc↑
Seinfeld	29.6	29.7	81.2	0.922	0.841
Frasier	**23.8**	**24.3**	**83.1**	**0.933**	**0.888**
Scrubs	32.6	36.4	76.1	0.883	0.853

Table 4.4: Performance on the test set. We report the Diarisation Error Rate both with and without consideration of the overlapping regions, **DER(O)** and **DER** respectively. **Acc** denotes a character recognition accuracy for the segments that overlap with the groundtruth. **Ppc** and **Rpc** are the average per-character precision and recall, respectively.

Model	Version	Seinfeld	Frasier	Scrubs
Wav2Vec2.0 [Baevski et al. 2020]	`ASR_BASE_960H`	45.0	36.9	36.3
Whisper [Radford et al. 2023]	`medium.en`	13.2	13.5	10.6
WhisperX [Bain et al. 2023]	`medium.en`	**11.8**	**11.2**	**9.2**

Table 4.5: Word Error rate (WER) (%) on each dataset.

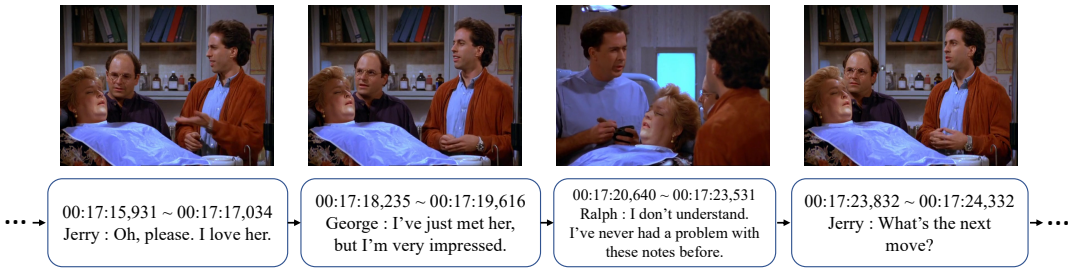


Figure 4.4: Qualitative example. Our method produces the speech segments with timestamps, and assigns the character who spoke it.

these two shows.

Speech transcription performance. Our method uses the WhisperX ASR model which also produces the speech transcription results. We compare the performance with the state-of-the-art models in Table 4.5. Word Error Rate (WER) is computed after applying the Whisper text normaliser to both ground truth and predictions which can be found in the original paper [Radford et al. 2023]. We see that WhisperX outperforms both Wav2vec2.0 and Whisper. This is because the VAD Cut & Merge preprocessing reduces the hallucination of Whisper, which is also mentioned in the original paper [Bain et al. 2023].

Qualitative example. We show a qualitative example of our results in Fig. 4.4. As can be seen, our method assigns the character for each speech segment, as well as timestamps and the transcription.

4.5 Conclusions

In this work, we show promising first steps towards a model for character-aware subtitling, which we hope will be beneficial for improving accessibility, and facilitating further research in video understanding. Our method is not perfect, however. Our recognition efforts fail on short segments such as exclamations and also do not deal with overlapping speech – though the latter does not appear to be a serious limitation in practice. Furthermore, to generate the true SDH subtitles, we would need to classify and categorise every sound, not just speech – something our model is not yet capable of.

Chapter 5

Character-aware audio-visual subtitling in context

The paper has been accepted for publication at the Asian Conference on Computer Vision (ACCV), 2024.

Character-aware audio-visual subtitling in context

Jaesung Huh Andrew Zisserman

VGG, University of Oxford

Abstract

This paper presents an improved framework for character-aware audio-visual subtitling in TV shows. Our approach integrates speech recognition, speaker diarisation, and character recognition, utilising both audio and visual cues. This holistic solution addresses what is said, when it's said, and who is speaking, providing a more comprehensive and accurate character-aware subtitling for TV shows. Our approach brings improvements on two fronts: first, we show that audio-visual synchronisation can be used to pick out the talking face amongst others present in a video clip, and assign an identity to the corresponding speech segment. This audio-visual approach improves recognition accuracy and yield over current methods. Second, we show that the speaker of short segments can be determined by using the temporal context of the dialogue within a scene. We propose an approach using local voice embeddings of the audio, and large language model reasoning on the text transcription. This overcomes a limitation of existing methods that they are unable to accurately assign speakers to short temporal segments. We validate the method on a dataset with 12 TV shows, demonstrating superior performance in speaker diarisation and character recognition accuracy compared to existing approaches.

5.1 Introduction

Character-aware audio-visual subtitling is an emerging area that aims to automatically generate subtitles for TV shows and movies, including the corresponding speaker names. This task involves determining three key aspects: *what* is being said, *when* it is said, and *who* is saying it. This capability is essential for the audio-impaired, so that they can follow video material – indeed it is a requirement of Subtitles for Deaf and Hard-of-hearing (SDH [Szarkowska 2020]) that the subtitles include information about speaker identification, as well as information on sound effects and music. It also enables the annotation of large-scale video datasets for training the next generation of visual-language models, capable of learning a higher-level story understanding of video material.

The task builds on developments in three specialised areas: *Automatic speech recognition* (ASR, or speech-to-text) that is primarily concerned with transcribing spoken words into text – determining 'what is spoken'; *Speaker diarisation*, that aims to organise multi-speaker audio into homogeneous single speaker segments, effectively solving 'who spoke when'; and *Character recognition*, that aims to identify the characters appearing in the video clips. Each of these areas is well explored, and can use single modality methods (i.e. audio only or visual only) or audio-visual methods. For example, ASR can be audio-only, e.g. [Radford et al. 2023; Hsu et al. 2021; Gulati et al. 2020], or audio-visual, e.g [Afouras et al. 2019a; Shi et al. 2022a; P. Ma et al. 2023; Gabeur et al. 2022]. Similarly, common methods can be used across the areas. For example, voice embeddings can be used for diarisation by clustering [Quan Wang et al. 2018; A. Zhang et al. 2019; Kinoshita et al. 2021], and for recognition by matching to a gallery of voices [K. Li and Wrench Jr 1982; Suchitha and Bindu 2015; Kaphungkui and Kandali 2019]. However, because these are somewhat independent areas, they do not alone provide all the ingredients required.

Recent works have introduced methods and datasets for character-aware audio-visual subtitling, building on elements from the three areas above [Korbar et al. 2024; Lerner et al. 2022]. The state-of-the-art method of Korbar *et al.* [Korbar et al. 2024], proceeds in two stages: it first builds a gallery of voice embeddings for each character using audio-visual methods, and then generates the character-aware

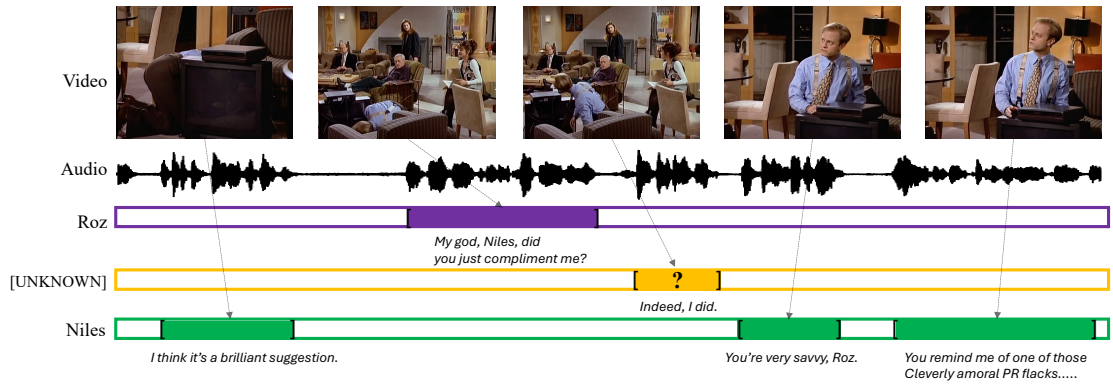


Figure 5.1: An example video clip and output of our method. Dialogues in TV shows typically flow continuously, and speaker identities can often be inferred from the content and context of the conversation. In some cases, it's possible to diarise speakers solely based on textual context. Even though we cannot see the speaker visually – so have no evidence from lip-movement – we can infer that the utterance with a question mark (**?**) belongs to 'Niles' by looking at temporal context of the dialogue.

subtitles using only audio recognition. Despite its accomplishments, however, this method has two significant shortcomings when determining the character speaking during a temporal segment: (1) it has a poor performance for short segments (those lasting less than 2 seconds), often assigning the wrong character; and (2) it has a low yield over all segments, as often it is unable to classify the character.

In this paper we make three contributions. First, we introduce a new method for identifying the speaker for short segments, building on the insight that assignments that cannot be resolved using only local (temporal) information, can often be disambiguated using the *temporal context* of the surrounding dialogue. We investigate two complementary approaches for this task: (i) using speaker recognition, we note that a short utterance in a dialogue may well be spoken by a character with a longer utterance (where the identity is not ambiguous) elsewhere in the scene, and the short segment can then be assigned using *local* voice embeddings, rather than voice embeddings from a gallery where audio conditions may substantially differ; (ii) using a large language model (LLM), the identity of the character speaking the short segment can be resolved based on the *content* of the dialogue, as illustrated by the example in Figure Fig. 5.1. The second contribution is to use a *local visual embedding* around the lip motion synchronised with the speech to determine the speaker. This overcomes a limitation of [Korbar et al. 2024], where a CLIP descriptor of the entire frame is used to predict the speaker identity. The use of a local visual embedding leads to higher yield of assignments for speaker segments.

Taken together these two contributions significantly improve the performance over that of [Korbar et al. 2024]. As our third contribution, we validate our method on a large evaluation dataset covering 12 TV series. This dataset incorporates the existing dataset used by [Korbar et al. 2024] and additional shows from [Lerner et al. 2022], demonstrating the generalisation ability of our method.

5.2 Related Work

Several subtasks within this field have already been explored by researchers. *Speech recognition* [Chan et al. 2016; Gulati et al. 2020; Malik et al. 2021; Radford et al. 2023], or speech-to-text, is primarily concerned with transcribing spoken words into text. However, this subtask typically overlooks the timing of speech and fails to identify the speaker. *Speaker diarisation* [Quan Wang et al. 2018; Diez et al. 2019; J. S. Chung et al. 2020c; T. J. Park et al. 2022] aims to identify speech regions and assign speaker labels to each person in an audio file. This task clusters speech segments by speaker without necessarily matching them to specific known individuals. *Character recognition* [Berg et al. 2004; Ramanathan et al. 2014; Kalogeiton and Zisserman 2020; Y. Hu et al. 2015; Poignant et al. 2017], a well-studied topic in computer vision and speech processing, assigns names to characters appearing in scenes. Character-aware audio-visual subtitling requires the integration of all three tasks, utilising both audio and visual cues from the video. Existing approaches often tackle these problems individually, with methods that are either audio-only or audio-visual. However, there is a growing need for a holistic solution that addresses all aspects simultaneously.

Character recognition in videos. Recognising characters in video [Everingham et al. 2006; Everingham et al. 2009; Nagrani and Zisserman 2017; Haurilet et al. 2016] is a challenging task due to the presence of multiple characters in a single frame, occlusions, and variations in appearance. Several methods have been proposed to incorporate additional modalities, such as audio [A. Brown et al. 2021a; Nagrani and Zisserman 2017], or transcripts [Everingham et al. 2006; Everingham et al. 2009; Haurilet et al. 2016] which are often unavailable. There are a line of works which use speaker diarisation in TV shows and use the result to cluster the speaker identities [Bost et al. 2015; Sharma and Narayanan 2022b]. However, they

simply cluster the speaker identities, not assigning the actual character's name. Our task involves assigning the specific names of speakers in TV shows using a castlist.

Audio-visual speech processing. Numerous studies have examined human conversation from a broad perspective. Given that these interactions primarily occur through speech, a wide range of research focuses on audio-only approaches to various tasks, including speech recognition [Radford et al. 2023; Baevski et al. 2020; Gulati et al. 2020], speaker identification [Desplanques et al. 2020; J. S. Chung et al. 2020b; Koluguri et al. 2022], and speaker diarisation [Bredin 2023; Fujita et al. 2019]. With a rise of the multimodal learning, researchers have started to incorporate visual information such as lip movements in addition to audio information to improve the performance of these tasks. For example, they use lip movements [Afouras et al. 2019a; Shi et al. 2022a] or faces [J. S. Chung et al. 2020c; A. Brown et al. 2021b] in addition to audio to improve the performance of this task.

LLM for video understanding. Large Language Models (LLMs) [Touvron et al. 2023; Achiam et al. 2023; A. Q. Jiang et al. 2023; Anil et al. 2023] have driven the great progress not only in Natural Language processing but also in computer vision [Bai et al. 2023; H. Liu et al. 2023; Alayrac et al. 2022; H. Lu et al. 2024] and audio processing [Yuan Gong et al. 2023; Yuan Gong et al. 2024]. Over the past few years, there has been a plethora of works which leverage LLMs in various video understanding tasks. There are two different approaches to this. The first approach integrates a pretrained LLM with visual or audio backbones as part of the entire model, fine-tuning it to understand multimodal content [Hang Zhang et al. 2023; E. Song et al. 2024; T. Han et al. 2023; Maaz et al. 2024]. The second approach uses LLM separately from video models to improve performance on video understanding tasks [J. Chen et al. 2023; J. Wang et al. 2023].

Human conversation datasets. There has been growing interest in audio-visual datasets with rich transcriptions of spoken conversations, including speech transcripts, timestamps and speaker identities. Several datasets exist with annotations for either one of these aspects. LRS series [Afouras et al. 2019a; Afouras

et al. 2018a] have advanced audio-visual speech recognition technology, but their single-speaker focus limits development of multi-speaker systems for conversational settings. There exist audio-visual speaker diarisation datasets [J. S. Chung et al. 2020c; E. Z. Xu et al. 2022] with multiple speakers but do not have a speech transcripts. The AMI-Corpus [Kraaij et al. 2005] and VoxMM [Kwak et al. 2024] are multimodal datasets which provide audio-visual data with speaker identities and speech transcripts. However, both focuses on different domain than ours such as meeting scenarios, commercial or interviews. *Bazinga!* [Lerner et al. 2022] offers rich transcriptions of TV shows, including word-level timestamps, speech transcripts, and speaker identities. We use this dataset to verify our pipeline's performance.

Relation to the method of Korbar *et al.*[Korbar et al. 2024]. This work also aims to generate character-aware subtitle generation. However, it has several limitations. Firstly, it fails to utilise spatial information from lip-moving areas, which could significantly enhance speaker recognition accuracy. The method utilises CLIP-PAD [Korbar and Zisserman 2022] which recognises characters in scenes without employing a face detection model to identify clips for single-speaker regions. Unfortunately, these clips may contain multiple faces, potentially confusing the model when tasked with identifying the actual speaker. Secondly, it doesn't take advantage of the time-based context when matching speaker names to parts of speech. In TV shows, conversations usually progress in a continuous manner. As a result, it's often possible to figure out who is speaking by considering the overall flow of the dialogue and the context in which things are said.

5.3 Assigning Speakers to Short Audio Segments

The task here is to assign speaker identities to short temporal speech segments. We assume that we have a gallery/library of voice embeddings available for the principal characters.

It is well known that identifying speakers by their voice alone typically fails in verification tasks when the input audios are short [Poddar et al. 2018; Y. He et al. 2023]. This is because state-of-the-art speaker embedding extractors [Desplanques et al. 2020; Koluguri et al. 2022; Torgashov et al. 2023] are normally trained

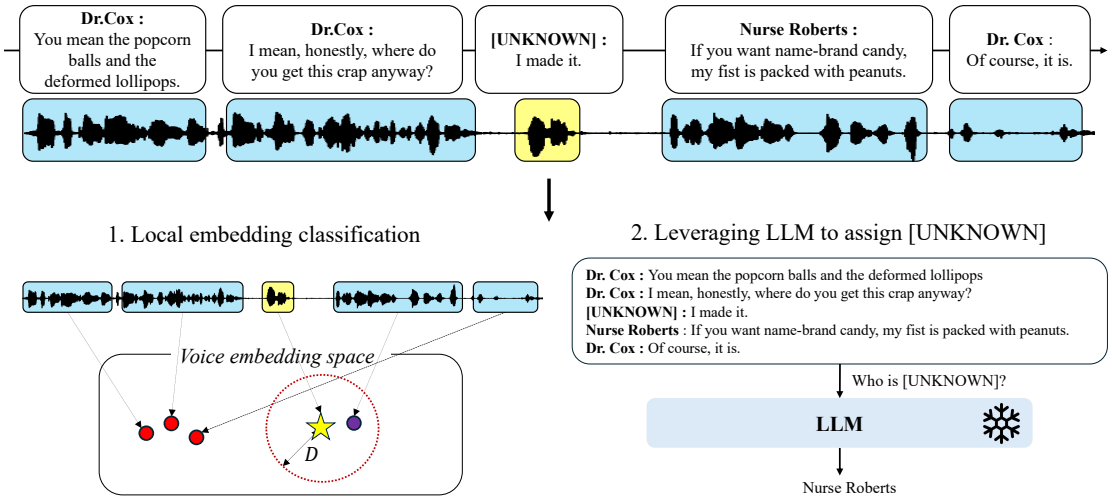


Figure 5.2: **Assigning speakers to short audio segments**. First, we use speaker embeddings from nearby segments where we have high confidence in speaker identification (left). Second, we employ a Large Language Model (LLM) to determine the speaker based on the content of conversation. (right)

with segments of at least 2 seconds of audio waveforms. TV shows contain many short segments (see Fig. 5.5), resulting in false classification when using standard classification methods on the embedding, such as nearest class centroid.

To solve this short segment speaker assignment problem, we use the *temporal context* of the human conversation. There is a high chance that the speaker of the short speech segment we are interested in is involved in the dialogue : which means that the speaker might speak elsewhere and for longer within the scene. The key concept is that speaker identity can be accurately predicted for longer audio segments. These identified segments can then be used to classify speakers in shorter audio segments nearby. We employ this idea into two complementary ways: using local speech embeddings, and using the language (text) of the dialogue. Assuming we know the speakers of long audio segments with a high confidence, we demonstrate how to leverage this information to determine the speakers in shorter audio segments. The method is illustrated in Fig. 5.2.

Local embedding classification. As is known from diarisation, there are advantages in comparing to *local* embeddings when deciding if two speakers are the same or not. Since the two embeddings are computed under the same environment – and so have the same background sounds, the same reverberation, even the same microphone, many of the 'nuissance' variables are removed, simplifying the classification challenge. Thus, to determine the speaker of a short segment, its

embedding is compared to the segment embeddings of the other (known) speakers in the scene, instead of comparing to their class centroid.

In detail, we extract the speaker embeddings within n_{local} preceding and succeeding sentences around the segment of interest (where $n_{local} = 15$). Then the speaker is assigned by computing the distance between the embeddings of the short segment and segments with known speakers using first nearest neighbor classification. If the distance is below a threshold D then the assignment is accepted, otherwise the short segment is classed as `unknown`, and the assignment is determined (if possible) by using the text content, as described next.

Leverage LLM to assign speakers of `unknown`. As illustrated in Fig. 5.1, the speaker identity can be inferred solely by using the *content* of the dialogue (i.e. without actually hearing the voice). Since large language models (LLMs) have a good predictive 'understanding' of dialogues, they can be queried to predict the speaker of the short segment, given the named speakers of other utterances in the dialogue. We apply this LLM classification in the cases that cannot be classified using voice alone, since it is a weaker cue.

Specifically, we ask the LLM model to predict who the speaker is of the short segment, using zero-shot prompting. We provide the n_{llm} (e.g. 15) sentences with the speaker names both before and after this `unknown` sentence. The LLM model is tasked to answer with: either one of the characters that appear within this dialogue with $2n_{llm} + 1$ sentences; or `unknown` if the speaker is from outside the dialogue; or 'Can't tell' if the speaker cannot be inferred only from the provided dialogue. The prompt used also has three examples along with their answers, followed by the actual query and dialogue. The detailed prompt instructions and three examples are provided in the supplementary material.

5.4 Using Local Visual Predictions to Assign Speakers

The task here is to recognise all characters speaking within a video clip. We assume that we have a gallery/library of visual embeddings available for the principal characters. Although speech is sometimes difficult to recognise due to background

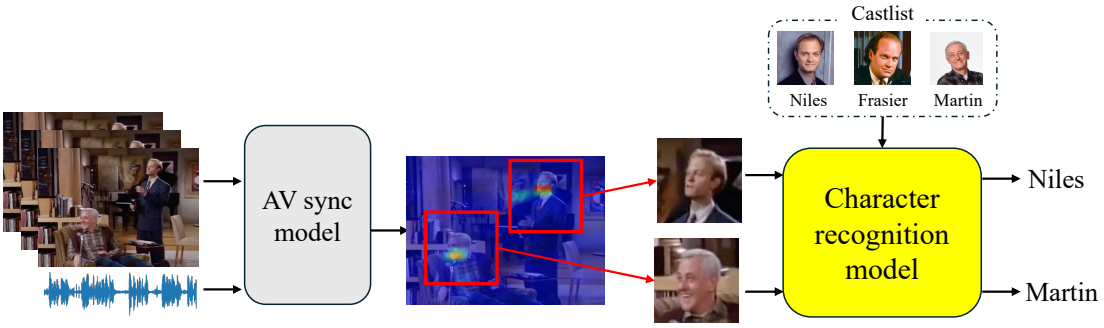


Figure 5.3: The visual prediction process for a speech segment. Visible speakers with lip movements synchronised with the speech audio are recognised by using a visual embedding from the castlist. This assigns an identity to the corresponding speaker.

noise or overlapping voices, the corresponding visual frames often provide a clear view of the speaker. We can use this visual information to help identify speakers. Fig. 5.3 illustrates the method.

To identify all visible speakers in the scene, we employ a multi-step process. First, we run a pretrained audio-visual synchronisation model [Afouras et al. 2020a] that detects lip motions by producing a heat map where areas around moving lips are activated. We then crop spatial regions around the detected peaks with a fixed width and height, and extract visual embeddings from each cropped region using a CLIP-based character recognition model [Korbar and Zisserman 2022]. We compare the distances between these visual embeddings to all actors in the cast list. Finally, we select the cropped regions that identify speakers with high confidence (above a predetermined threshold) and store these predictions for subsequent speaker assignment steps.

This cropping approach essentially extracts a local visual embedding of the face. It overcomes a limitation of the character recognition model [Korbar and Zisserman 2022], which is confused if multiple people appear in the same frame since it uses a global frame embedding. In summary, we use audio-visual cues to assign identities to speaking segments with visible faces, and audio-only embeddings to assign identities when the face is not visible.

5.5 Implementation Details

We follow the approach of [Korbar et al. 2024] for generating character-aware subtitles using video and a cast list for each show. Their two-stage method first builds a gallery of audio exemplars – speech segments with high-confidence character name assignments. These exemplars are then used to assign speaker names to all speech segments using centroid classification. If the minimum distance from the nearest exemplar exceeds a threshold, then no specific character name is assigned, allowing for characters without exemplars who cannot be classified. We detail our implementation of this method in the following subsections, highlighting the improvements we have made over the original method of [Korbar et al. 2024]. Fig. 5.4 shows a schematic overview of our entire pipeline.

5.5.1 Stage 1. Building audio exemplars

The goal of this stage is to extract audio exemplars from the video for which we know the corresponding speaker.

We first run Voice Activity Detection (VAD) based on Automatic Speech Recognition (ASR) model on the audios to generate the speech transcripts with corresponding timestamps at a sentence level. Visible speakers are then determined by using the synchronisation of lip movements and the speech with the self-supervised trained audio-visual model [Afouras et al. 2020b] that produces a heatmap of where the lip-motions are synchronised. We crop the surrounding spatial regions of each peaks in the heatmap and visually recognise characters in the region. Video clips with a single peak are kept as exemplar candidates, but predictions from the clips with more than one peak are also kept for assigning the speaker later. Then, we conduct additional audio filtering for the exemplar candidates to reduce the label noise. We detail the process below.

Stage 1–1. VAD + ASR. The goal of this stage is to generate speech transcripts with corresponding timestamps. We use publicly available pipeline [Bain et al. 2023] to produce speech transcripts with timestamps in a sentence level. We assume each sentence is spoken by a single speaker at this point, but we address the case of overlapping speech in subsequent stages. This step produces subtitles without speaker identities. Unlike [Korbar et al. 2024], we do not use any

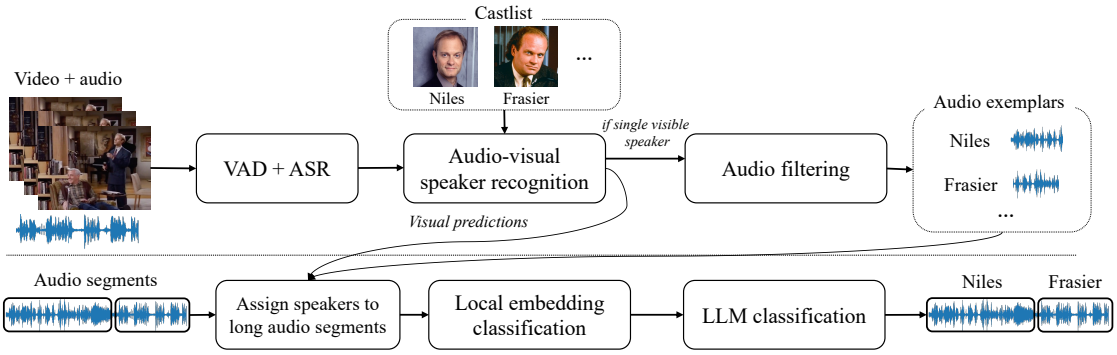


Figure 5.4: A schematic overview of our pipeline. We first extract the audio exemplars from videos (top) and use them label all audio segments (bottom).

pretrained laughter detector. Instead, we run a speech enhancement network to reduce background noise in the following step.

Stage 1–2. Audio-visual speaker recognition. This stage aims to recognise all speakers in the visual scenes and collect video clips with a single visible speaker using a castlist per video. First, we run a pretrained audio-only speech enhancement model [Defossez et al. 2020] to reduce background noise, thereby reducing false positives in the following stage. Then, we visually recognise the visible speakers' identities by using the method explained in Section 5.4. We need a gallery of images to compare the distance between each visible person and characters in the castlist. We collect up to 10 images per each character and form a visual embedding per character.

After running this model, we categorise the video clips into three types: (i) clips without any peaks, (ii) clips with a single peak, and (iii) clips with multiple peaks. The second type, clips with a single peak, are considered as our exemplar candidates in subsequent stages. However, we proceed to run the character recognition model on both the second and third types. We keep the output predictions to use as candidates for classification in Stage 2. Multiple peaks can occur for two reasons: either multiple speakers are talking simultaneously, or the model produces false positives.

Stage 1–3. Audio filtering This stage further reduces label noise by focusing on single-speaker video clips from the previous stage. We extract speaker embeddings from the audio and analyse each embedding's N (e.g. 5) nearest neighbors. We retain the embedding only if all N neighbors belong to the same speaker;

otherwise, we remove the corresponding audio segment from our exemplars.

5.5.2 Stage 2. Assigning speaker identities of each speech segment

Stage 1 aims to collect audio exemplars for which we know the corresponding speaker identities with high certainty. This stage aims to assign speaker identities to all segments. We first classify the long audio segments ($>$ 2 sec). For each segments, we only compare the distance between the exemplars from the visible speakers, which we obtain in Stage 1–2. If no visible speakers are detected, we compare the distance between the exemplars from castlist. Then, we use the local temporal context, using local embedding classification and LLM which are described in Section 5.3. The long segments as well as audio exemplars from the previous stage are used for local embedding classification.

After assigning speakers to each audio segments in a sentence level, we run the public overlapping speech detection model to detect the overlapping speech. If the segment is detected with overlapping speech, we assign the speaker with two nearest speakers along the time axis.

5.5.3 Implementation details

We use WhisperX [Bain et al. 2023] for VAD+ASR model. We further use Silero VAD [Team 2021] for filtering out the false detections from the WhisperX. ECAPA-TDNN [Desplanques et al. 2020] is used for speaker embedding extractor, pre-trained with VoxCeleb [Nagrani et al. 2019]. We use LWTNet [Afouras et al. 2020b] for audio-visual synchronisation model and crop the activate region with $W = H = 350px$ to recognise the characters. We use $n_{local} = 15$ preceding and suceeding audio segments. We use public Llama3-70B 4-bit quantised model [1], finetuned with instruction sets, to assign speakers with $n_{LLM} = 15$ preceding and succeeding sentences. We use public overlapping detection model from `pyannote 2.1` [Bredin and Laurent 2021]. Rest of the parameters are identical to those in [Korbar et al. 2024]. After detecting overlapping speech, we divide the audio segments wherever there is silence longer than 1 second, using word-level timestamps

[1] https://huggingface.co/screevoai/llama3-70b-instruct-4bit

from WhisperX. The same speaker is assigned to these divided audio segments. All hyperparameters are determined by grid search on validation sets.

5.6 Dataset and evaluation metrics

This section explains the dataset we have used to validate our method and evaluation metrics.

5.6.1 Dataset

LLR-TV. [Korbar et al. 2024] has released a TV shows dataset including six episodes each from Frasier, Seinfeld, and Scrubs, along with transcripts, speaker names, and timestamps. The official dataset website [2] has released version 1.1, which includes fixed annotations they have made. We use the current version to verify our method, but we also report the performance from the original paper in Section 5.7. For each series, we use the sixth episode as our validation set to determine the hyperparameters. The rest of the dataset is used as our test set.

***Bazinga!*-gold-TV.** *Bazinga!* [Lerner et al. 2022] dataset provides a rich set of annotations from 16 different TV shows and movies, such as speech transcripts with timestamps, speaker, addressee and entity linking information. The dataset itself is divided into gold and silver based on the level of annotations. We use all TV shows in the gold set to verify our method including *Battlestar Galactica (B.G.)*, *Breaking Bad (B.B.)*, *Buffy the Vampire Slayer (Buffy)*, *Friends*, *Game of Thrones (GoT)*, *Lost*, *The Big Bang Theory (TBBT)*, *The Office (Office)* and *The Walking Dead (W.D.)*. We exclude *StarWars* since our paper focuses on TV shows.

Since our method is audio-visual while the dataset only provides audio, we need to adjust the timestamps in the annotations to match our videos. We use the audio-audio alignment method introduced by [T. Han et al. 2024] to obtain precise temporal alignment by comparing the audio provided in the dataset with the audio from our video source. Similar to LLR-TV, we use the last episode of each series as our validation set, with the remaining episodes serving as our test set.

[2] https://www.robots.ox.ac.uk/~vgg/research/look-listen-recognise/

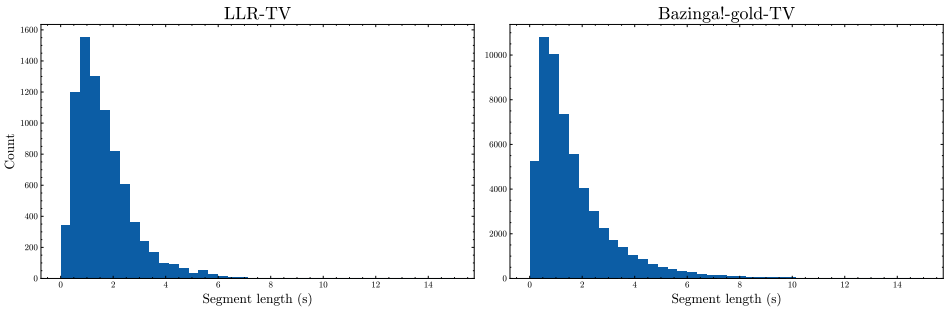


Figure 5.5: Distribution of segment lengths on LLR-TV and *Bazinga!*-gold-TV.

Fig. 5.5 shows the distribution of segment lengths in both datasets. 71.0% of segments in LLR-TV and 71.5% in *Bazinga!*-gold-TV are shorter than 2 seconds. This indicates that recognising speakers in shorter segments is crucial for analysing conversations in TV shows. Note that while LLR-TV is manually corrected, *Bazinga!*-gold-TV provides timestamps obatined through force-alignment. Thus the annotation is relatively noisy (e.g. they do not provide annotations for *Previously...* part.).

The source video used from both dataset were obtained in DVD format.

5.6.2 Evaluation metrics

Diarisation metrics. **DER** [*The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan* 2009] is a standard evaluation metric for speaker diarisation. However, recent studies [Cheng et al. 2022; T. Liu and K. Yu 2022] have highlighted a significant limitation of DER: its time-duration-based computation fails to accurately capture the recognition performance for short-term segments. To address this limitation, Conversational-DER (**CDER**) is introduced, which calculates speaker diarization accuracy at the utterance level and also accounts for short segments. For more details on CDER, please refer to [Cheng et al. 2022].

In this paper, we employ DER with a forgiveness collar of 0.25 seconds, taking into consideration instances of overlapping speech.

Character recognition metrics. Character recognition accuracy (**Acc.**) is calculated for segments that overlap with ground truth segments. A segment is considered correctly classified if the character's name is accurately identified and matches the corresponding overlapping ground truth segment. Precision and recall for character identification are also reported for each show. Both metrics are

Model	Modality	Frasier		Seinfeld		Scrubs	
		DER	**CDER**	**DER**	**CDER**	**DER**	**CDER**
SimpleDiar [*Simple Diarization repository* 2024]	**A**	24.2	58.5	24.5	56.2	**24.4**	52.6
`pyannote` [Bredin 2023]	**A**	24.7	84.1	35.4	88.7	31.1	75.8
LLR∗ [Korbar et al. 2024]	**A** + **V**	24.3	-	29.7	-	36.4	-
LLR† [Korbar et al. 2024]	**A** + **V**	26.4	39.1	28.0	40.7	26.7	40.3
Ours	**A** + **V**	**20.3**	**28.8**	**23.3**	**33.6**	25.7	**37.0**

Table 5.1: Diarisation performance on LLR-TV test set. Lower is better. **LLR∗** is from the original paper before the GT was corrected and **LLR†** is the our reproduced result with annotation corrections from the website. **DER** : Diarisation Error Rate (%), **CDER** : Communication DER (%), **A** : Audio, **V** : Video.

Model	Mod	B.G.		B.B.		Buffy		Friends		GoT		Lost		TBBT		Office		W.D.	
		DER	**CDER**	**DER**	**CDER**	**DER**	**CDER**	**DER**	**CDER**	**DER**	**CDER**	**DER**	**CDER**	**DER**	**CDER**	**DER**	**CDER**	**DER**	**CDER**
SimpleDiar [*Simple Diarization repository* 2024]	**A**	61.3	101.0	68.8	154.5	31.5	62.6	**46.8**	85.6	38.3	85.8	90.9	117.2	**20.6**	38.0	**33.3**	**70.3**	93.4	123.9
`pyannote` [Bredin 2023]	**A**	**58.7**	104.9	70.4	136.7	**31.3**	**59.6**	60.5	128.8	**37.2**	85.1	**88.1**	111.9	30.3	70.0	35.4	112.5	100.7	138.2
LLR† [Korbar et al. 2024]	**A+V**	79.5	101.6	92.9	135.4	55.4	77.5	55.9	72.6	63.7	120.0	111.7	115.4	29.5	39.5	44.2	93.9	108.9	130.6
Ours	**A+V**	62.7	**87.2**	**67.0**	**99.4**	46.2	62.0	47.1	**64.4**	44.2	**82.7**	89.0	**86.6**	27.3	**36.0**	40.2	71.8	**92.5**	**97.1**

Table 5.2: Diarisation performance on *Bazinga!*-gold-TV. **DER** : Diarisation Error Rate (%), **CDER** : Communication DER (%), **Mod** : Modality, **A** : audio, **V** : Video.

calculated for all characters (**Prec.** and **Rec.**) and separately for main characters (**Prec.(M)** and **Rec.(M)**) in each series. A list of main characters for each show is provided in the supplementary material.

5.7 Results

This section presents our overall results on the test sets, comparing them to other baselines. We also provide a detailed analysis of how our method improves the classification of short segments and the effect of using local visual predictions. We conclude by showcasing qualitative examples of our method and comparing them to the baseline.

5.7.1 Overall performance

Diarisation performance. We report the diarisation performance of our method on the LLR-TV test set in Table 5.1. Our method is compared against three competitive baselines, including two audio-only models and one audio-visual diarisation method, LLR. In terms of DER, our method demonstrates superior performance on Frasier and Seinfeld compared to all other models, and achieves comparable results on Scrubs. More notably, when considering the CDER, our method significantly outperforms other baselines with margins of 10.3%, 7.1%, and 3.3% on Frasier, Seinfeld, and Scrubs, respectively. This indicates that our method

	LLR [Korbar et al. 2024]					Ours				
	Acc.	**Prec.**	**Rec.**	**Prec.(M)**	**Rec.(M)**	**Acc.**	**Prec.**	**Rec.**	**Prec.(M)**	**Rec.(M)**
Frasier	87.0	**91.6**	87.0	**92.5**	89.4	**88.9**	89.8	**88.9**	90.3	**92.6**
Seinfeld	84.5	**89.0**	84.6	**92.5**	89.4	**85.8**	87.1	**86.0**	89.5	**90.7**
Scrubs	84.3	**89.2**	84.9	**91.0**	88.1	**84.4**	84.8	**85.1**	85.1	**90.7**

Table 5.3: Character recognition performance on LLR-TV test set. **Prec.** and **Rec.** indicate the precision and recall of overall audio segments respectively, while **Prec.(M)** and **Rec.(M)** are those of main characters in TV shows. **Acc.** is the character recognition accuracy for those which overlap with one of the groundtruth timestamps.

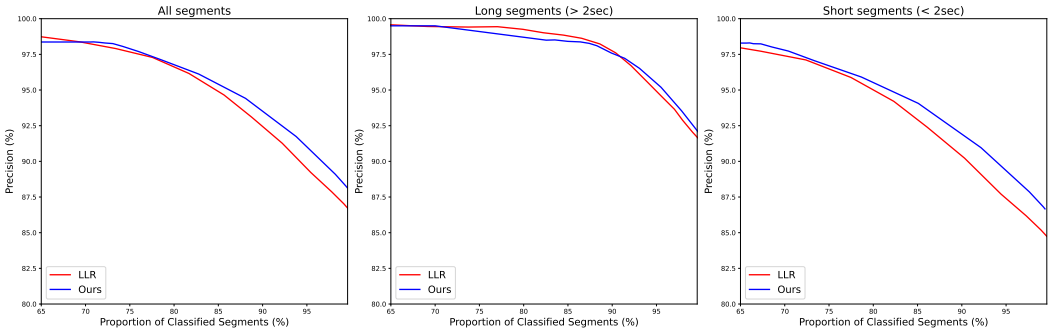


Figure 5.6: Precision-POCS curves for audio segments in LLR-TV test set.

recognises characters more accurately, even in short segments, compared to other methods.

Table 5.2 compares our pipeline against other baselines on the *Bazinga!*-gold-TV, where our method shows better performance in most TV series. It's worth noting that due to label noise in the dataset, the metrics should be taken with a 'grain of salt', a point that is also made in the original paper [Lerner et al. 2022].

Character recognition performance. Table 5.3 shows the character recognition performance on LLR-TV. The reported performance from both LLR and our method is based on the highest accuracy achieved by varying the hyperparameter D (see Section 5.3) on the validation set. Compared to the reproduced LLR, our method demonstrates higher character recognition accuracy. Interestingly, although the LLM is instructed not to predict the speaker when it cannot be inferred from the input dialogue, it mostly selects a speaker from within the dialogue, resulting in higher recall for both all segments and segments from main characters. LLR does not predict speakers for 6.8% of the test set, while our method classifies only 0.77% as `unknown`.

	Centroid	Local	Local + LLM (Ours)			
n_{llm}	-	-	5	10	15	30
LLR-TV val	85.3	87.1	87.5	**88.2**	**88.2**	87.6
LLR-TV test	84.3	86.3	86.6	**86.9**	86.8	86.7

Table 5.4: Character recognition accuracy % on short segments (< 2 sec). **Centroid** : using exemplar centroids only, **Local** : using local embeddings only, **Local + LLM (Ours)** : using both local embeddings and LLM by varying n_{llm}.

Precision-POCS curve. We demonstrate the Precision – Proportion of Classified Segments (POCS) trade-off in Fig. 5.6, showing results for all segments (left), long segments (middle), and short segments (right) by varying the threshold D. We include the curve from the LLR method for comparison. The graphs show that the precision of character recognition decreases as we classify more audio segments. Compared to LLR, our method shows similar performance on long segments but demonstrates superior ability in classifying short segments. This verifies our method's effectiveness in identifying speakers of short utterances.

5.7.2 Effects of local embedding classification and LLM on short segments

Table 5.4 demonstrates the effectiveness of local embedding classification and LLM in classifying short segments. Character recognition accuracies are presented for audio segments shorter than 2 seconds using three methods: (i) nearest centroid classification with exemplars only, (ii) local embedding classification only, and (iii) a combination of local embedding classification and LLM, with varying n_{llm}. The results show that local embedding classification improves performance on short segments compared to using centroids of audio exemplars per character (1.8% and 2.0% improvement on validation and test sets, respectively). This improvement likely occurs because local embeddings tend to share similar audio environments.

Furthermore, leveraging LLM enhances recognition performance by utilising the temporal context of the dialogue. On the LLR-TV test set, $n_{llm} = 10$ shows the best performance, while on the validation set, $n_{llm} = 15$ performs identically to $n_{llm} = 10$. Since only the validation set is used to determine parameters, $n_{llm} = 15$ is applied for all reported results.

	LLR		LLR + vis		LLR + vis + SE (Ours)	
	# of exemplars	% of total	# of exemplars	% of total	# of exemplars	% of total
1-1. VAD + ASR	5554	100	5554	100	5554	100
1-2. Audio-visual speaker recognition	2192	39.5	3332	60.0	3409	**61.4**
1-3. Audio filtering	1213	21.8	1681	30.3	1734	**31.2**

Table 5.5: Effect of using local visual predictions (vis) and speech enhancement (SE) on exemplar yield on the LLR-TV.

5.7.3 Effects of utilising spatial regions and speech enhancement on audio exemplar yield and performance

Table 5.5 demonstrates how cropping lip areas and speech enhancement improve exemplar yield. Unlike LLR, which uses whole frames without speech enhancement, our method identifies characters by focusing on the spatial region of speakers and reducing background noise to decrease false positive peaks, increasing exemplar yield by 8.5% and 0.9% respectively. We also measure how accurate these exemplars are by comparing to the ground truth. Out of 1734 exemplars, the pipeline mispredicts only **6** speakers (0.34%), 5 from Frasier and one from Seinfeld. This high accuracy offers two advantages: (i) more audio segments are correctly classified, and (ii) more precise embeddings are obtained for local embedding classification of short segments. Detailed results are provided in the supplementary material.

5.7.4 Qualitative examples

We present qualitative results from two series, *Scrubs* and *Friends*, in Fig. 5.7. The figure shows speech recognition output and corresponding timestamps produced by our method, along with character recognition results from both LLR and this approach. In both series, LLR fails to predict speakers for short utterances such as *"You know."*, *"Me."* or *"Why not?"*. In contrast, our method utilises the temporal context of the conversation to correctly classify the speaker for these brief segments. It is important to note that the yellow utterances in the figure are initially classified as `unknown`. However, after employing LLM, these are correctly assigned to the appropriate speakers.

Figure 5.7: Qualitative examples from two TV series, *Scrubs* and *Friends*.

5.8 Conclusions

This paper presents an advanced framework for character-aware audio-visual subtitling in TV shows, addressing limitations in existing methods. Key contributions include a novel method for identifying speakers in short segments using temporal context, the use of local visual embeddings around lip-moving areas, and validation on a large dataset covering 12 TV series. Results demonstrate significant improvements in both diarisation performance and character recognition accuracy, particularly for short speech segments.

5.9 Appendix

5.9.1 Character recognition accuracy on *Bazinga!*-gold-TV

Table 5.6 shows the character recognition performance on *Bazinga!*-gold-TV, comparing with that from LLR [Korbar et al. 2024]. The reported performance of LLR corresponds to its highest character recognition accuracy (**Acc.**). This peak accuracy is achieved by adjusting the threshold during the nearest centroid classification process. We showcase that our method achieves higher accuracy than LLR in all TV shows. It also achieves higher recall for both 'all characters' and 'main characters' compared to LLR. LLR exhibits higher precision for 'all characters'

Table 5.6: Character recognition performance on *Bazinga!*-gold-TV test set. **Acc.** is the character recognition accuracy for those which overlap with one of the groundtruth timestamps. **Prec.** and **Rec.** indicate the precision and recall of overall audio segments respectively, while **Prec.(M)** and **Rec.(M)** are those of main characters in TV shows.

	LLR [Korbar et al. 2024]					**Ours**				
	Acc.	**Prec.**	**Rec.**	**Prec.(M)**	**Rec.(M)**	**Acc.**	**Prec.**	**Rec.**	**Prec.(M)**	**Rec.(M)**
B.G.	62.7	67.7	64.1	**72.7**	71.1	**68.7**	**69.1**	**70.1**	69.1	**75.9**
B.B.	60.5	69.4	61.4	**83.6**	64.9	**67.5**	**72.6**	**68.6**	76.3	**69.4**
Buffy	57.6	62.4	58.1	71.7	57.5	**61.6**	**62.6**	**62.1**	**75.0**	**65.0**
Friends	70.9	**82.1**	72.9	**85.8**	79.7	**74.2**	75.6	**76.4**	76.0	**83.2**
GoT	52.6	55.4	55.7	**57.3**	59.7	**61.6**	**62.9**	**65.3**	54.1	**63.7**
Lost	51.4	61.2	53.4	62.6	56.6	**60.4**	**63.9**	**62.6**	**65.4**	**67.3**
TBBT	80.2	**88.8**	80.4	**90.9**	85.4	**81.5**	82.7	**81.8**	83.2	**86.2**
Office	68.7	**77.5**	68.9	**84.8**	74.1	**70.8**	74.4	**71.2**	75.4	**75.0**
W.D.	49.7	**60.3**	51.9	**57.0**	52.9	**54.7**	57.6	**57.1**	53.5	**64.0**

Table 5.7: Exemplar recognition performance on LLR-TV. # **exem** denotes the number exemplars extracted from our method and # **correct** denotes the number of correctly classified exemplars. **Others** are a group of characters who are usually guest stars for one or a few episodes and the number of them is given in parentheses.

Frasier				**Seinfeld**				**Scrubs**			
Char	# exem	# correct	Acc (%)	**Char**	# exem	# correct	Acc (%)	**Char**	# exem	# correct	Acc (%)
Frasier	347	342	98.6	**Jerry**	353	352	99.7	**J.D.**	152	152	100
Niles	58	58	100	**George**	41	41	100	**Dr.Cox**	102	102	100
Roz	28	28	100	**Elaine**	66	66	100	**Turk**	22	22	100
Daphne	30	30	100	**Kramer**	27	27	100	**Dr.Kelso**	65	65	100
Martin	31	31	100					**Elliot**	88	88	100
								Carla	73	73	100
Others (7)	47	47	100	**Others (12)**	122	122	100	**Others (13)**	82	82	100

and 'main characters' in a few TV shows. This difference in performance is due to how each method handles short segments. LLR does not assign speakers to some of short segments, instead marking the character as `unknown`. In contrast, our LLM tends to predict speakers for these `unknown` segments after local embedding classification. It often assigns one of the characters appearing in the dialogue, even when we explicitly instruct the LLM that it doesn't have to make an assignment if it's unsure. Thus, the precision decreases but the recall increases.

5.9.2 Exemplar recognition accuracy

Table 5.7 shows the accuracy of character recognition for the exemplars. We demonstrate that the accuracy of the audio exemplar building stage is nearly perfect. Out of 1,734 exemplars, most characters show 100% accuracy. In terms of the number of exemplars, we extract more audio samples than [Korbar et al. 2024] from the same Seinfeld test set (541 vs. 407). This showcases the effectiveness of our cropping-based approach in terms of both exemplar yield and accuracy.

5.9.3 List of main characters per series

Table 5.8 lists the main characters per series, both in LLR-TV and *Bazinga!*-gold-TV. Note that we report the names of main characters in *Bazinga!*-gold-TV as they are in the dataset annotation.

Table 5.8: List of main characters per show.

Show	Main characters
LLR-TV	
Frasier	Frasier, Martin, Niles, Roz, Daphne
Seinfeld	Jerry, Elaine, George, Kramer
Scrubs	J.D., Dr.Cox, Dr.Kelso, Carla, Turk, Elliot
***Bazinga!*-gold-TV**	
Battlestar Galactica (B.G.)	Admiral William Adama, President Laura Roslin, Captain Kara Thrace, Cpt. Lee Adama, Number six, Dr. Gaius Baltar, Lt. Sharon Valerii
Breaking Bad (B.B.)	Walter White, Skyler White, Jesse Pinkman, Hank Schrader, Marie Schrader
Buffy the Vampire Slayer (Buffy)	Buffy Summers, Willow Rosenberg, Xander Harris, Angel, Rubert Giles
Friends	Rachel Green, Monica Geller, Phoebe Buffay, Joey Tribbiani, Chandler Bing, Dr. Ross Geller
Game of Thrones (GoT)	Danerys Targaryen, Jon Snow, Jorah Mormont, Tyrion Lannister, Catelyn Stark, Sansa Stark, Arya Stark, Cersei Lannister, Eddard Stark, Robert Baratheon
Lost	Dr. Jack Sheperd, Kate Austen, Sayid Jarrah, Hugo Reyes, Sunhwa Kwon, Charlie Pace, Clarie Littleton, Michael Dawson, John Locke, Shannon Rutherford, James Ford
The Big Bang Theory (TBBT)	Sheldon Cooper, Penny, Howard Wolowitz, Raj Koothrappali, Leonard Hofstadter
The Office (Office)	Jim Halpert, Michael Scott, Ryan Howard, Pam Beesly, Dwight Schrute, Stanley Hudson, Phyllis Vance, Angela Martin
Lost	Rick Grimes, Lori Grimes, Carl Grimes, Carol Peletier, Shane Walsh, Andrea Harrison, Dale Horvath, Glenn Rhee

5.9.4 LLM prompt

Table 5.9 shows the LLM prompt we've used to determine the `unknown` character in the dialogue. We adopt the strategy introduced in [R. Y. Park et al. 2024]. Given the query dialogue with the `unknown` character we want to classify, we first ask the LLM to summarize the dialogue. Then, we ask the model who would be the `unknown` in the dialogue based on the generated summary. A list of characters with their corresponding indices is provided in the prompt. We compute the softmaxed logits of the tokens corresponding to each index and choose the largest one to assign the speaker. We use this same prompt for experiments in LLR-TV and *Bazinga!*-gold-TV.

There can be multiple `unknown`s in the query dialogue. We mark the sentence for which we want to identify the speaker by placing asterisks (**) before and after it. This explicitly instructs the LLM to predict the speaker only for that specific sentence.

Table 5.9: LLM prompt to determine `unknown`.

Prompt for determining [UNKNOWN]

system : You are a AI assistant to analyze the transcript of TV shows. Your job is to figure out who are [UNKNOWN]s in a dialogue in TV shows. Tell the truth and answer as precisely as possible.
(provide the dialogue here)
user : Write a summary for the above conversation.
assistant : *(model generates the summary)*
user : Based on the summary, your job is to identify the name of the speaker of '[UNKNOWN]' when the line starts and ends with '**'. You must use the context and the flow of the dialogue, using the speakers' names and what they speak. The list of speakers with their corresponding tokens are provided below. Choose [UNKNOWN] if his or her name is not in the dialogue, or when you are not sure.
(provide the list of speakers here)
Only output one number after ANSWER:
assistant : ANSWER:

Example of the dialogue ($n_{LLM} = 5$)

Dr.Cox : You can use it.
Dr.Cox : God, I hate Halloween.
Carla : Somebody needs to adjust their attitude if they want some candy.
Dr.Cox : You mean, the popcorn balls and the deformed lollipops.
Dr.Cox : I mean, honestly, where do you get this crap anyway?
[UNKNOWN] : I made it.
NurseRoberts : If you want name brand candy, my fish is packed with peanuts.
Dr.Cox : Of course it is.
Carla : Oh, what's the matter?
Carla : Did Raggedy Ann scare you?
Dr.Cox : What are you, a rat?

Example of the list of speakers

1: Dr.Cox, 2: Carla, 3: NurseRoberts, 4: [UNKNOWN]

Part III

Audio-visual dataset curation

Chapter 6

Spot the conversation: speaker diarisation in the wild

The paper has been accepted for publication at the Interspeech, 2020.

Spot the conversation: speaker diarisation in the wild

Joon Son Chung[1,2*] Jaesung Huh[1*] Arsha Nagrani[1,†]
Triantafyllos Afouras[1,‡] Andrew Zisserman[1]
[1]VGG, University of Oxford [2] KAIST, Korea

Abstract

The goal of this paper is speaker diarisation of videos collected 'in the wild'. We make three key contributions. First, we propose an automatic audio-visual diarisation method for YouTube videos. Our method consists of active speaker detection using audio-visual methods and speaker verification using self-enrolled speaker models. Second, we integrate our method into a semi-automatic dataset creation pipeline which significantly reduces the number of hours required to annotate videos with diarisation labels. Finally, we use this pipeline to create a large-scale diarisation dataset called VoxConverse, collected from 'in the wild' videos, which we will release publicly to the research community. Our dataset consists of overlapping speech, a large and diverse speaker pool, and challenging background conditions.

*Equal technical contribution. † Now at Google Research. ‡ Now at Meta AI.

6.1 Introduction

Speaker diarisation is the challenging task of breaking up multi-speaker video into homogeneous single speaker segments, effectively solving *"who spoke when"*. Beyond being an interesting research problem in itself, it is also a valuable pre-processing step for a number of applications, including speech-to-text.

While state-of-the-art diarisation systems perform remarkably well for speech from constrained domains (e.g. conversational telephone speech [Sell and Garcia-Romero 2014; Zhu and Pelecanos 2016; Garcia-Romero et al. 2017; A. Zhang et al. 2019] or meeting speech [Yella and Bourlard 2013]), this success does not transfer to more challenging conditions found in online videos 'in the wild'. The challenges here include the lack of a fixed domain (videos can be from talk shows, news broadcasts, celebrity interviews, home vlogs), a large number of speakers (some of whom are off-screen), short rapid exchanges with cross-talk, and background degradation consisting of channel noise, laughter and applause.

These conditions make manual annotation of online videos a daunting task for human annotators, leading to a dearth of large-scale public diarisation datasets of unconstrained speech. While large-scale evaluations are held regularly by the National Institute of Standards in Technology (NIST-RTE), these are limited to constrained audio-only datasets, which are not freely available to the research community (Table 6.1).

To attempt to remedy some these issues, the DIHARD challenges [Sell et al. 2018; Ryant et al. 2019] were introduced in 2018. These are valuable annual challenges that cover 11 different data domains, including mother-child conversations, meetings and courtroom settings. One of these domains is also web videos, however there is a limited amount of data (only 2 hours). The datasets are also audio-only, and are only available to challenge participants (not released freely to the research community).

A large-scale diarisation dataset of videos 'in the wild' would encourage the development of new audio-visual diarisation techniques that deal with unconstrained conditions. Inspired by the recent success of automatic audio-visual dataset creation pipelines (VoxCeleb [Nagrani et al. 2017; J. S. Chung et al. 2018; Nagrani et al. 2019], VGGSound [H. Chen et al. 2020]), we propose a scalable, audio-visual

method for speaker diarisation in web videos. Our method relies heavily on the recent successes of active speaker detection [J. S. Chung and Zisserman 2016] and face and speaker verification [Parkhi et al. 2015; Schroff et al. 2015; W. Xie et al. 2019]. We then integrate this method into a semi-automatic dataset creation pipeline – consisting of both automatic annotation and manual verification. We use this pipeline to curate VoxConverse, a challenging and diverse speaker diarisation dataset from 'in the wild' videos.

Our automatic diarisation method exploits the following three key ideas; Firstly, the speech for on-screen identities can be accurately segmented automatically using active speaker detection and then identified using face recognition, the core basis for the VoxCeleb pipeline [Nagrani et al. 2019]. Second, there has been great progress in creating audio-visual speech enhancement models [Afouras et al. 2018b; Ephrat et al. 2018; Afouras et al. 2019b], which separate overlapping speech into single speaker streams. Given the amount of cross-talk and background noise in web videos, we use this model to better isolate and identify speaker identities. The above two ideas allow us to accurately identify and isolate speech for on-screen speakers. Finally, to accurately recognise *off-screen* speakers, we utilise state of the art speaker recognition embeddings that verify identities from audio alone (Figure 6.1).

Concretely, we make the following three contributions: (i) We create an automatic audio-visual diarisation method using active speaker detection, face recognition, speech enhancement and audio-only speaker recognition. (ii) We integrate our method into a semi-automatic dataset creation pipeline which consists of human annotation and automatic diarisation. Our pipeline is scalable, and significantly reduces the number of hours required to annotate videos. (iii) We use this pipeline to curate VoxConverse, a challenging 'in the wild' audio-visual diarisation dataset. We compare our audio-visual diarisation method to existing audio-only baselines on our dataset, and show that large performance gains can be obtained from integrating visual information.

6.2 Related works

Speaker diarisation has been an active field of research for many years, but remains one of the most challenging tasks in speech processing. Deep learning techniques have not been applied to speaker diarisation to the same degree that they have for other tasks, partially due to the lack of end-to-end models for diarisation, but also due to the lack of diverse, large-scale datasets like ImageNet [J. Deng et al. 2009] and VoxCeleb [Nagrani et al. 2017].

Much of the progress in the field has been driven by a series of NIST Rich Transcription challenges (NIST-RTE), which focuses almost solely on the meeting domain. The series also proposed the diarisation error rate (DER) as an evaluation metric for speaker diarisation, which is now used as the primary metric across all domains and evaluations. Research into speaker diarisation has largely evolved independently for different domains, with broadcast news [S. Tranter and D. A. Reynolds 2004], telephone speech [Canavan et al. 1997], and meetings [Janin et al. 2003; Carletta et al. 2005] being the most popular domains. For each domain, specific datasets have been introduced and used, all created by manual annotation. We provide a summary in Table 6.1.

The DIHARD series of challenges [Sell et al. 2018; Ryant et al. 2019] were introduced to overcome the domain dependency in the field – the data consists of recordings from different conversational domains, including audiobooks, broadcast interviews, child speech and so on. The evaluation conditions are challenging, and even the best performing systems score relatively high diarisation error rates of around 20% with ground truth voice activity detector (VAD), and 30% with system VAD. The annotation is performed with very fine granularity, which allows evaluation without a forgiveness collar. Barring DIHARD, all other datasets and evaluations include a generous forgiveness collar and exclude overlapping speech from scoring. Inspired by DIHARD, we annotate overlapping speech in VoxConverse and include it in evaluation. For almost all existing datasets, annotation is done manually and solely using the audio. Annotation without visual information is challenging, particularly when the number of speakers is large, since it is easy to be confused between voices without the additional identity redundancy provided by the face. Unlike other works, our dataset creation pipeline is semi-automatic,

scalable and audio-visual.

Name	Acou. Cond.	Free	Ann. Method
2005 NIST RTE	Meetings	✗	Manual
CALLHOME [Canavan et al. 1997]	Telephony	✗	Manual
AMI Meeting Corpus [Carletta et al. 2005]	Meetings	✓	Manual
ICSI Meeting Corpus [Janin et al. 2003]	Meetings	✓	Manual
Fisher† I and II [Cieri et al. 2004]	Telephony	✗	Manual
DIHARD [Sell et al. 2018; Ryant et al. 2019]	Mixed	✗	Manual
VoxConverse	Multi-media	✓	Semi-automatic

Table 6.1: Comparison to existing speaker diarisation datasets. **Acou. Cond.:** Acoustic conditions; **Ann. Method:** Annotation Method; †: Fisher English Training Speech part I and II.

6.3 Dataset description

The development set of VoxConverse consists of 216 multispeaker videos covering 1,218 minutes with 8,268 speaker turns annotated. The test set contains approximately 232 videos covering 2,612 minutes. The statistics of the dataset can be seen in Table 6.2.

Videos included in the dataset are shot in a large number of challenging multi-speaker acoustic environments, including political debates, panel discussions, celebrity interviews, comedy news segments and talk shows. This provides a number of background degradations, including dynamic environmental noise with some speech-like characteristics, such as laughter and applause. Our dataset is audio-visual, and contains face detections and tracks as part of the annotation.

The videos in the datasets consist of quick, short speech segments. On average, 91% of the video time contains speech, and 3–4% of this contains speech where one speaker overlaps with another speaker. The overlap percentage varies between videos; one video for example has an overlap percentage of 29.8%. Videos vary in length from 22 seconds to 20 minutes. Unlike other domains such as telephony, each video has on average between 4 and 6 speakers, with one video in the dataset having 21 speakers.

set	# videos	# mins	# spks	video durations (s)	speech %	overlap %
Dev	216	1,218	1 / 4.5 / 20	22.0 / 338.2 / 1097.4	10.7 / 93.2 / 99.8	0 / 3.8 / 28.7
Test	232	2,612	1 / 6.5 / 21	26.0 / 675.6 / 1200.0	46.9 / 89.6 / 100	0 / 3.1 / 29.8

Table 6.2: VoxConverse dataset statistics. Entries that have 3 values are reported as min/mean/max. # **spks:** Number of unique speakers per video. # **mins:** Total number of minutes in the dataset. **video durations (s):** Length of videos in seconds. **speech %:** Percentage of video time that is speech. **overlap %:** Percentage of speech per video when 2 or more speakers overlap.

6.4 Dataset collection

The dataset collection process consists of two stages – initial annotations are generated automatically using our proposed audio-visual method, and the annotations are then checked and refined by human annotators.

6.4.1 Automatic pipeline

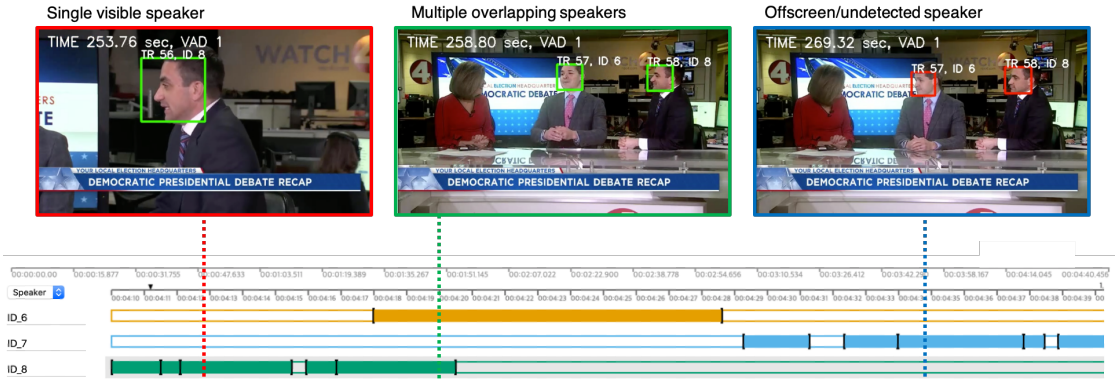


Figure 6.1: Output of our automatic audio-visual diarisation method. Green squares on the images represent face detections with positive ASD output, red squares represent face detections with negative ASD output. The identities are labelled as ID_6, ID_7 and ID_8, and speaker timelines show when each identity is speaking. For clarity, we only show 3 frames from the video. Our method elegantly deals with visual speakers, overlapping speech and undetected/off-screen speakers.

The automatic computer vision pipeline to curate VoxConverse is similar to that used to compile `VoxCeleb1` [Nagrani et al. 2017] and `VoxCeleb2` [J. S. Chung et al. 2018].

Stage 1. Collection of videos. The first stage is to obtain a list of videos. We start from a number of keywords including 'panel debate' and 'discussion' in order to obtain videos where multiple people are talking alternately or at the same time. The list of videos is obtained by searching the keywords on YouTube, and duplicate

URLs that appear in the search results of multiple keywords is removed. Moreover, we remove the videos that are identical or very similar in content based on tf-idf features [Teller 2000] extracted from the YouTube auto-generated subtitles. The list contains a range of videos, ranging from US presidential debates and talk shows to documentaries.

Stage 2. Shot detection. Shot boundaries are then determined to find within-shot frames for which face tracking is to be run. The boundaries are found by comparing intensity and brightness across consecutive frames [Castellano 2020].

Stage 3. Face detection and tracking. A CNN face detector based on the Single Shot Scale-invariant Face Detector (S3FD) [S. Zhang et al. 2017] is used to detect faces on every frame of the video. This detector allows the detection of faces at various scales and poses. Within each shot, face detections are grouped together into face tracks using a position-based tracker, as in [Nagrani et al. 2017; J. S. Chung et al. 2018].

Stage 4. Face-track clustering. A face recognition CNN is used to extract embeddings for every face track. The network used here is based on the ResNet-50 [K. He et al. 2016] trained on the VGGFace2 dataset. The embeddings are extracted 5 times per face track at uniform intervals, and then averaged. The embeddings are clustered using Agglomerative Hierarchical Clustering [Day and Edelsbrunner 1984], but a large penalty is added to the distance matrix between overlapping face track so that they are never clustered together.

Stage 5. Active speaker detection (ASD). The goal of this stage is to determine if the visible face is the speaker. Two systems are used for this purpose. The first method uses a variant [S.-W. Chung et al. 2019] of SyncNet [J. S. Chung and Zisserman 2017], which is a two-stream CNN that determines the active speaker by estimating the correlation between the audio track and the mouth motion of the video. The second method isolates the speech of the target speaker from a mixture of sounds using an audio-visual speech enhancement (AVSE) network [Afouras et al. 2018b] then uses an off-the-shelf voice activity detector, WebRTC [Johnston and Burnett 2012], to determine the speech segment.

Each method has its weaknesses – the SyncNet ASD activates when the phoneme in the background speech matches the viseme shown on the target face since the model does not consider temporal context; the WebRTC voice activity detector is often activated from the residual signal left in the AVSE output which, despite the reduced power, causes false alarms. Therefore, a face track is considered to be speaking only if both of the methods agree, which helps reduce false alarms from laughter and music.

Stage 6. Labelling off-screen speech. A pre-trained speaker recognition model [J. S. Chung et al. 2020b] is used to verify the identity of speech that comes from off-screen speakers. Any parts of the audio with positive voice activity detector (VAD) output, but with no visible active speaker is considered to be an off-screen speech segment. Speaker embeddings are extracted for the whole video, then the off-screen speech segments are compared to all speech segments with visible active speaker using the cosine distance between the embeddings. If the cosine distance is below a threshold, the off-screen segment is assigned to the speaker; if not, the segment is left as unknown for the human annotator to verify. This procedure is closely related to the multi-modal diarisation method of [J. S. Chung et al. 2019].

Discussion. In creating the VoxCeleb datasets, very conservative thresholds were used in both active speaker detection and face verification, since it was necessary to be very certain about the speaker labels without any human intervention. This is, however, at the cost of high false rejection, which meant that a large number of true speech segments were discarded.

In contrast, a speaker diarisation dataset must contain continuous audio recording with different identities speaking in turns. Therefore, we cannot discard parts of video based on low confidence, but the entire video must be labelled in full. The thresholds are optimised to minimise the overall Diarisation Error Rate (DER) (see section 7.5), since high false alarms and high false rejections both lead to increased man-hours during the manual verification and correction stage. Note how our pipeline consists of two ASD methods, as we show later, this redundancy is beneficial for performance.

6.4.2 Manual verification

The output of the automatic pipeline has been checked and corrected by the authors of this paper using a customised version of the VGG Image Annotator [Abhishek Dutta and Zisserman 2019; A. Dutta et al. 2016]. This was done so that the authors can identify the failure modes and make guidelines for the external annotators when the process is scaled up. The tool allows the user to watch and verify the annotation at various speeds, and with aid of video.

During the annotation process, a number of failure modes were identified. The most common is non-visible speech segment assigned to the wrong speaker, but false alarm of the VAD and missed overlapped speech are also relatively common.

Guidelines. Speech segments are split when pauses are greater than 0.25 seconds. Unlike some previous datasets in diarisation, laughter is not assigned to identities, as it is difficult to assign an accurate label to audience laughter. Anything that can be transcribed, including short utterances such as 'yes' and 'right', are considered to be speech. Known speakers are named in the annotation process to facilitate easier cross-checking. The annotators are asked to be as careful as possible that the marked boundaries are within 0.1 seconds of the true boundary.

Quality check. In order to verify the quality of manually checked annotations, a subset of the data has been labelled independently by two different annotators. This subset contains 1 hour of material from 15 YouTube videos. The diarisation error rate between the two annotations is approximately 1%, using the labels from one annotator as the reference and the other as the prediction. This error can be mostly attributed to disagreements on the source of off-screen speech segments.

Discussion. Diarisation labels for 'in the wild' conversations are difficult to obtain. It is almost impossible to manually annotate the segments in our dataset without the video. Even with the video, it can take 10 times the video duration to annotate segments to satisfactory quality if *starting from scratch*, particularly for many speakers. In contrast, the verification of our audio-visual method output takes around twice the video duration, and is possible with less experienced annotators.

The time taken to annotate correlates strongly to the quality of the output from the automatic method. The first few videos in the development set were annotated with initial hyperparameters that gave relatively poor performance. The diarisation labels were then manually fixed, and the parameters were re-tuned on this data to minimise the diarisation error rate. More videos were then generated using the new set of parameters and this process was repeated a few times. While it is possible that some types of errors are more time-consuming for humans to fix compared to others, we have observed that the annotation became faster after each iteration.

6.5 Experiments

We compare our audio-visual method to an audio-only DIHARD 2019 baseline, and also compare performance to two ablations.

DIHARD 2019 baseline. The second DIHARD [Sell et al. 2018; Ryant et al. 2019] challenge provides a baseline system based on the JHU submission of the first DIHARD challenge. We use this public code[1] as an audio-only baseline.

The overall procedure is as follows. Speech segments are obtained using VAD, and divided into short overlapping segments (1.5s with 0.75s overlap). Speaker embeddings are extracted using the x-vector [Snyder et al. 2018] system, and the similarities between the embeddings are scored with a pre-trained probabilistic linear discriminant analysis (PLDA) [Ioffe 2006; Kenny et al. 2013] model also provided in the code. Segments are then grouped using agglomerative hierarchical clustering (AHC) based on PLDA scores. We report the best performance by tuning the threshold of the AHC on the development set.

Two variants are compared, with and without the speech enhancement module [L. Sun et al. 2018] which has been made publicly available [2]. The system uses a Long short-term memory (LSTM) based speech denoising model trained on simulated training data. This model shows state-of-the-art performance on speech enhancement, and has shown its effectiveness for diarisation in the first DIHARD challenge.

[1] https://github.com/iiscleap/DIHARD_2019_baseline_alltracks
[2] https://github.com/staplesinLA/denoising_DIHARD18

Table 6.3: Results on the dev set using baseline methods and our proposed audio-visual method. All values are in %. **MS:** missed speech; **FA:** false alarm; **SC:** speaker confusion; **DER:** diarsation error rate (where $DER = MS + FA + SC$). For each metric, the lower the better. † Audio-only baselines.

Name	**MS**	**FA**	**SC**	**DER**
DIHARD 2019 baseline [Sell et al. 2018] †	11.1	1.4	11.3	23.8
DIHARD 2019 baseline w/ SE [Sell et al. 2018; L. Sun et al. 2018] †	9.3	1.3	9.7	20.2
Ours (SyncNet ASD only)	2.2	4.1	4.0	10.4
Ours (AVSE ASD only)	2.0	5.9	4.6	12.4
Ours (proposed)	2.4	2.3	3.0	7.7

Ablations. A crucial design choice that we made is that we used two active speaker detection methods, and a segment was only marked positive when both methods gave a positive output. We consider two ablations of our method – one using only SyncNet-based ASD, and the other using only AVSE-based ASD.

Evaluation protocol. Methods are evaluated on the VoxConverse development set. We use the diarisation error rate (DER), defined as the sum of missed speech (MS), false alarm speech (FA), and speaker misclassification error (speaker confusion, SC). A forgiveness collar of 0.25 seconds is applied in order to compensate for small inconsistencies in annotation.

Training. All thresholds are tuned on the VoxConverse development set. The AHC threshold for speaker clustering is the only hyperparameter to be tuned in the audio-only baseline. The audio-visual method requires three key thresholds – cosine distance for face clustering, SyncNet confidence for active speaker detection, and cosine distance for speaker identification. The first of these affect performance the most, since any error in the identity clustering directly causes speaker confusion.

Results. Table 7.3 shows the results of all the evaluations. Our audio-visual method obtains a DER much lower than the audio-only state-of-the-art baselines, showing the efficacy of using visual information for diarisation on this dataset. The ablation analysis for the ASD methods proves the effectiveness of using two active speaker detectors – the combined method has a significant decrease in false alarm rate for only a small increase in missed speech.

With regards to the difficulty of VoxConverse, we note that the DIHARD 2019

baseline obtains a DER of about 20% on our dataset (Table 7.3), and hence there is a lot of room for improvement. While this is lower than the 26% that the same model achieves on the extremely challenging DIHARD development set (with ground truth VAD), we hypothesize that this difference may be attributed to the use of a 0.25-second forgiveness collar in our evaluation protocol.

6.6 Conclusion

We have developed a high performance audio-visual algorithm for automated diarisation, and used it to generate a new speaker diarisation dataset, VoxConverse from 'in the wild' videos. The pipeline is fully scalable and effective across a range of domains. VoxConverse currently contains 70 hours of annotated video, but we are in the process of scaling up. The data was used in the second VoxCeleb Speaker Recognition Challenge in October 2020 and, after that, was released publicly to the research community free of charge.

Chapter 7

Playing a Part: Speaker Verification at the Movies

The paper has been accepted for publication at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2021.

Playing a Part: Speaker Verification at the Movies

Andrew Brown[1,†]* Jaesung Huh[1]* Arsha Nagrani[1,‡]*
Joon Son Chung[1,2] Andrew Zisserman[1]
[1] VGG, University of Oxford [2] KAIST, Korea
[3] Google Research

Abstract

The goal of this work is to investigate the performance of popular speaker recognition models on speech segments from movies, where often actors intentionally disguise their voice to play a character. We make the following three contributions: (i) We collect a novel, challenging speaker recognition dataset called VoxMovies with speech for 856 identities from almost 4000 movie clips. VoxMovies contains utterances with varying emotion, accents and background noise, and therefore comprises an entirely different domain to the interview-style, emotionally calm utterances in current speaker recognition datasets such as VoxCeleb; (ii) We provide a number of domain adaptation evaluation sets, and benchmark the performance of state-of-the-art speaker recognition models on these evaluation pairs. We demonstrate that both speaker verification and identification performance drops steeply on this new data, showing the challenge in transferring models across domains; and finally (iii) We show that simple domain adaptation paradigms improve performance, but there is still large room for improvement.

*Equal technical contribution. † Now at Meta AI. ‡ Now at Google Research.

7.1 Introduction

After hearing the actor Steve Martin's smooth American accent on the Ellen show, is it possible to recognise that he is the voice behind 'Inspector Jacques Clouseau', the comical French character in the movie 'Pink Panther', without access to his visual appearance? Or recognise Anne Hathaway's voice in the movie 'Les Misérables', where she plays 'Fantine', singing through tears about her sadness and desperation? In this paper we investigate the challenging task of speaker recognition for actors from two different domains – the first being when they are speaking *'naturally'* in interviews, and second while *playing a part* in a movie, where they may be intentionally modifying their voice in order to play the role of a character or show emotion.

While recent years have shown great successes in speaker recognition [Snyder et al. 2018; Garcia-Romero et al. 2020; Wan et al. 2018; J. S. Chung et al. 2020b], these successes have been reliant on the collection of large, labelled datasets such as VoxCeleb [Nagrani et al. 2017; J. S. Chung et al. 2018; Nagrani et al. 2019] and others [McLaren et al. 2016; Fan et al. 2020]. The VoxCeleb datasets, while valuable, have been collected *entirely from interviews* of celebrities in YouTube videos and are limited in terms of linguistic content (celebrities mostly speak about their professions [Nagrani et al. 2020a]), emotion, and background noise. In contrast, movies contain speech covering emotions such as anger, sadness, assertiveness, and fright, and varied background conditions – imagine the shouting in a violent scene from an action movie, or a romantic scene of reconciliation in a romcom. As we show in this paper, models trained on VoxCeleb, when applied to a novel domain such as speech in movies, suffer from significant degradation in performance. In order to accurately measure this, there is a compelling need for real-world datasets and evaluation sets across these domains. Collecting and annotating datasets for every new domain encountered in the real-world, however can be an extremely expensive and time-consuming process. We introduce a scalable method to automatically generate data in a new domain (movies), and investigate the performance of state-of-the-art speaker recognition models on this data, where actors are intentionally disguising their voice. Being able to detect human identity under such conditions of spoofing is valuable for security and authentication [Cai et al. 2017; Z. Chen et al. 2017; Das et al. 2019], and as shown by psychology studies [Re-

ich and Duke 1979; Hirson and Duckworth 1993], is a challenging task even for humans.

In order to encourage research in domain adaptation for speaker recognition, we make the following three contributions: (i) We collect a novel speaker recognition dataset called VoxMovies, from 3,792 popular movie clips uploaded to YouTube. Our dataset consists of almost 9,000 utterances from 856 identities that appear in the VoxCeleb datasets, and contains challenging emotional, linguistic and channel variation (Fig. 7.1); (ii) We provide a number of domain adaptation evaluation sets, and benchmark the performance of state-of-the-art speaker recognition models on these evaluation pairs. We demonstrate that performance drops steeply on this new data for both speaker verification *and* identification, showing the challenge in transferring models across domains (from interviews to movies). We also investigate performance on positive pairs sampled across different movies, and reveal further performance drops; and finally (iii) We demonstrate that domain adaptation approaches added on top of already trained models improve performance, but there is still a severe degradation.

VoxMovies has been used to create a challenging test set for the VoxCeleb Speaker Recognition Challenge [Nagrani et al. 2020b] (VoxSRC2020)[1]. Data: https://www.robots.ox.ac.uk/~vgg/data/voxmovies/.

7.2 Related Work

Due to the reliance of machine learning on large, labelled datasets, domain adaptation has become popular in fields such as computer vision [Tzeng et al. 2017; Hoffman et al. 2018], text classification [P. Liu et al. 2017; Yuan Zhang et al. 2017], speech enhancement [Liao et al. 2018; Meng et al. 2018] and speaker verification [Garcia-Romero and McCree 2014; Garcia-Romero et al. 2014a; Garcia-Romero et al. 2014b]. Recent interest in domain adaptation for speaker recognition has largely focused on boosting the performance on datasets such as NIST-SRE16 or Speakers in the Wild (SITW) [McLaren et al. 2015], which does not have a large training set. In this case, most methods train on VoxCeleb [Nagrani et al. 2017; J. S. Chung et al. 2018] and evaluate on these evaluation sets.

[1] http://www.robots.ox.ac.uk/ vgg/data/voxceleb/competition2020.html

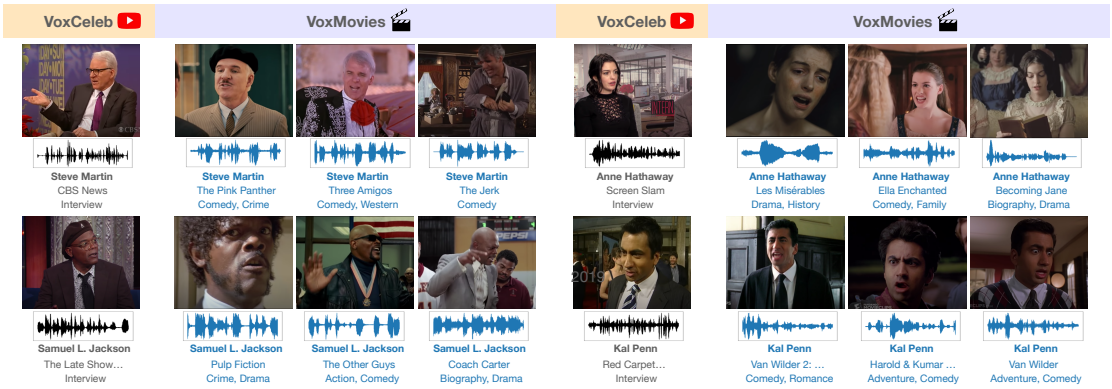


Figure 7.1: **Domain Gap between VoxCeleb and VoxMovies:** Unlike VoxCeleb (left in each panel), which consists of utterances from interviews, VoxMovies is sourced from movies of different genres (right in each panel). While VoxCeleb features utterances largely with calm, unvaried emotions, VoxMovies has challenging emotion and background noise. We show only a single frame for each utterance, below which is the name and genre of the film/video. VoxMovies contains 24 utterances from 5 movies for Anne Hathaway, including singing in the musical, 'Les Misérables', reading in an English accent in 'Becoming Jane', and arguing heatedly in an American accent in 'Ella Enchanted'. Samuel L. Jackson has 59 utterances from 14 movies, including his famous, assertive speech in 'Pulp Fiction' and bragging as an arrogant policeman in 'The Other Guys'.

Recent methods focus on adversarial training techniques, with [Qing Wang et al. 2018] introducing domain adversarial training that exploits domain prediction, and [Luu et al. 2020] proposing a channel adversarial training method by adding a channel discriminator to mitigate the domain mismatch problem by using the video labels in VoxCeleb, with both works [Luu et al. 2020; Qing Wang et al. 2018] using gradient reversal layers. [Rohdin et al. 2019] proposes an end-to-end domain adversarial training method adopting the architecture of Wasserstein Generative Adversarial Network (GAN) [Arjovsky et al. 2017] with adversarial domain loss and cross-entropy loss. It significantly improves the performance on language adaptation. [Bhattacharya et al. 2019a] also introduces a domain adaptation technique to new language or recording conditions by using a Domain Adversarial Neural Speaker Embeddings (DANSE) model which contains a 1-dimensional self-attentive residual block. [Bhattacharya et al. 2019b] explores various GANs to make the domain discriminator unable to distinguish whether embeddings are from the source or target domain. [Kang et al. 2020] exploits Model-Agnostic Meta-Learning (MAML), projecting speaker representations to a generalized embedding space and achieves better results on CNCeleb dataset [Fan et al. 2020]. Other research methods train channel-invariant, noise-robust speaker recognition models which can help improve the overall performance of the model in diverse do-

mains. [Jianfeng Zhou et al. 2019] shows a multi-task learning method that trains both a speaker recognition network and a discriminator that distinguishes types of noise in speaker representations. [J. S. Chung et al. 2020a] also tackles a similar problem by using an environment network and a KL-divergence based confusion loss to learn speaker-discriminative, environment-invariant representations.

Unlike existing works, we provide novel domain data for the *same identities as in VoxCeleb*, allowing us to investigate both cross-domain speaker verification, where pairs have one segment from either domain, and cross-domain identification, where speaker identification models are tested on a new, previously unseen domain. Both have not been previously possible. We focus on the domain adaptation from interviews to movies for male and female actors (explored for faces in [Nagrani and Zisserman 2017]). We propose and benchmark on such evaluations conditions, and additionally on several within-domain verification tasks.

7.3 Cross-Domain Data

Our goal in this work is to investigate the effects of cross-domain speaker verification in movies. The domain change we focus on here is from YouTube interviews (domain *D-I*), to speech in different genres of movies (domain *D-M*). The data for the two different domains are sourced as follows:

D-I: **Interviews from VoxCeleb [Nagrani et al. 2017; J. S. Chung et al. 2018]** These datasets are sourced solely from interviews uploaded to YouTube. These are mainly in studio, outdoor or red-carpet locations. In turn, and due to the often *professional* context, voices are mainly calm and rarely show any strong emotion. These utterances are degraded with real world noise that would be expected from these environments, such as background chatter or laughter.

D-M: **Movies from CMD [Bain et al. 2020] (VoxMovies)** For the second domain, we curate a dataset of speech from movies, called VoxMovies, which consists of 8,905 utterances for 856 different identities, sourced from 3,792 video clips from 1,452 movies. These movies cover a range of genres (see Figure 7.2). The utterances in VoxMovies are sourced from the Condensed Movies dataset (CMD) [Bain et al. 2020], which covers the *key scenes* from movies. The distinctive change of domain can be seen in the following characteristics of VoxMovies:

(1) Emotion: In line with different movie genres, the utterances cover emotions such as anger, sadness, assertiveness, and fright. Furthermore, the videos in CMD represent scenes that are integral to the story-line and the different character developments, such as a fight between two main characters, or when they make up later in the film. Hence the utterances in the dataset often capture the most emotional parts of each movie.

(2) Background noise: Each key scene in the CMD dataset covers many different settings, from a loud basketball stadium in *Coach Carter*, to an 18th century gathering in *Becoming Jane* (see Figure 7.1). This represents a far more varied set of degradation for each of the utterances. Also, there is often background music.

Importantly, in VoxMovies this variety of emotion and background noise is seen both within and across different identities. Firstly this is because on average each identity has utterances from 2.7 different movies (see Table 7.1), and these movies are likely to be of different genres. Secondly, the videos in this dataset show the important character arcs of these identities within each movie, where they show different emotions at different points in the story-line. Examples and further details can be found in Figure 7.1.

Note that the utterances in VoxMovies are all from identities that are represented in VoxCeleb1 and VoxCeleb2. We create an evaluation set, VoxMovies-(Test), featuring identities from VoxCeleb1. We also provide a small amount of VoxMovies data for training domain adaptation methods, VoxMovies-(Train), using a subset of the identities in the VoxCeleb2 dev set. There is no identity overlap between these partitions. Statistics are shown in Table 7.1.

7.4 Dataset Collection Pipeline

Our dataset collection pipeline is similar to the one used to collect the VoxCeleb datasets, albeit applied to YouTube clips of movie scenes from the Condensed Movies Dataset (CMD) [Bain et al. 2020], described below.

Condensed Movies Dataset. CMD [Bain et al. 2020] consists of key scenes from over 3K movies, totalling 1,270 hours. Provided alongside the dataset are cast lists for each of the featured movies, face-tracks for each of the clips, and face embeddings for discriminating identity for each of these face-tracks. The cast lists

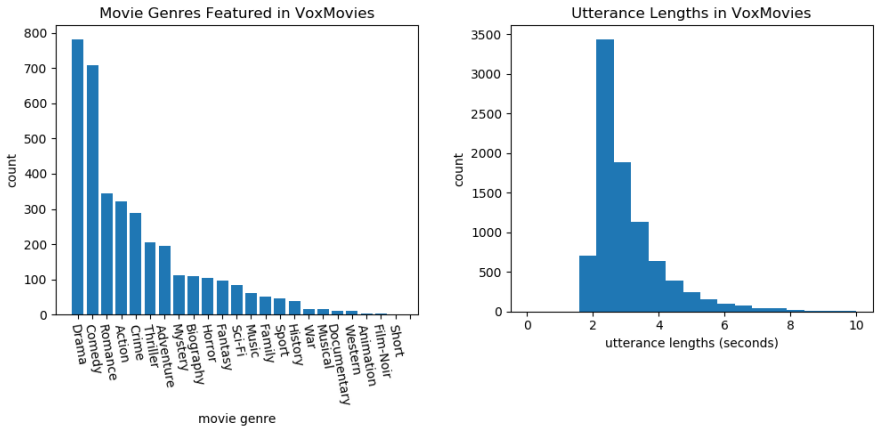


Figure 7.2: **Statistics of the VoxMovies dataset.** (a) The different movie genres that the utterances in VoxMovies are sourced from, and (b) the distribution of utterance lengths. The minimum length is 2 seconds by design choice.

Partition	ID Source	IDs	# Utter.	Clips / ID	Movies / ID
VoxMovies-(Test)	VoxCeleb1	485	4,943	5.1	2.7
VoxMovies-(Train)	VoxCeleb2	371	3,962	5.0	2.5

Table 7.1: **Dataset Statistics for VoxMovies.** The utterances are sourced from the Condensed Movies Dataset [Bain et al. 2020], which contains clips of key scenes from movies. Identities (IDs) overlap with the VoxCeleb1 (test) and VoxCeleb2 (dev) datasets.

give the names of people who are likely to appear in the clip. Our three stage method for collecting VoxMovies is as follows:

Stage 1. Sourcing candidate names. We compute the intersection between the VoxCeleb1 and VoxCeleb2 identities and the CMD cast lists.

Stage 2. Face verification. In this stage, face-tracks from the CMD dataset are classified as to whether they depict any of the candidate names from the previous stage using a three step process. (1) Example face images for each of the candidate names for the train and test set are sourced from the VGGFace2 [Cao et al. 2018] and VGGFace1 [Parkhi et al. 2015] datasets (these face datasets contain the same identities as the VoxCeleb2 and VoxCeleb1 datasets respectively). (2) 256D Embeddings are taken from the final hidden layer of a SE-Net50 architecture CNN for each of the face images, trained for discriminating face-identity using the VGGFace2 dataset [Cao et al. 2018]. For each identity, these embeddings are average pooled across all example face-images and L2 normalised, leaving one embedding per identity. (3) Verification is performed by computing the cosine

Eval. set	Positives Source	Negatives Source	# Utter.	# Positive Pairs	# Negative Pairs
E-1	*D-M (same)*	*D-M*	20,572	10,286	10,286
E-2	*D-I*, *D-M*	*D-I*, *D-M*	46,578	23,289	23,289
E-3	*D-I*, *D-M*	*D-I*	46,804	23,402	23,402
E-4	*D-I*, *D-M*	*D-M*	46,866	23,433	23,433
E-5	*D-M (diff)*	*D-M*	41,090	20,545	20,545

Table 7.2: **Statistics for the different evaluation sets:** Our evaluation sets are sourced from two different domains, interview material (*D-I*) and movie material (*D-M*). Key: **Utter.**: Utterances., *D-M (same)*: Segments in a pair are sourced from the same movie, *D-M (diff)*: Segments sourced from different movies.

similarity between the identity embeddings and the face-track embeddings from CMD. Any face-track with a similarity to an identity embedding above a high threshold is assigned that name.

Stage 3. Active speaker verification. Here we identify which face-tracks from the previous stage are speaking. This is performed using an audio-visual synchronisation network [S.-W. Chung et al. 2019], which predicts the correlation between the audio track and the motion of the mouth, and outputs a synchronisation confidence score for each 5 frame window. A window with confidence above a high threshold is classified as speaking. Face-tracks with a minimum of 2 seconds of consecutive speech are kept, and the audio track is clipped to the speaking period. See Figure 7.2 for a histogram of segment lengths.

Discussion. A manual check of VoxMovies-(Test) reveals that the automated process has a precision of $> 99\%$. The $< 1\%$ false positive face verifications, or false positive active speaker verifications are manually removed. The high thresholds for stages 2 and 3 are chosen as 0.55 and 0.13, respectively. These values (both cosine similarity) achieved high precision on a manually labelled validation set. As shown in Table 7.1, VoxMovies-(Test) has slightly more utterances than VoxMovies-(Train). This is due to the fact that a larger proportion of the identities in VoxCeleb1 are well-known actors than in VoxCeleb2, and hence they appeared in more of the movies in CMD. Impressively, on average, each identity has utterances from 2.7 different movies, with Robert DeNiro appearing in 25 movies.

7.5 Experiments

7.5.1 Evaluation Tasks

Our goal is to determine the performance of state-of-the-art models trained on the VoxCeleb2 dev set for speaker recognition performance on data from the movie domain (*D-M*).

Verification: We use the VoxCeleb1 and VoxMovies-(Test) datasets for evaluation, as these sets have no overlap with the identities in the VoxCeleb2 dev set (which is used to train all baselines). Given the two domains (*D-I* and *D-M*), we note that there are three different ways that pairs can be sourced for evaluation: pairs can be sourced entirely from *D-I*, entirely from *D-M*, or from both, where one utterance is from *D-I* and one is from *D-M*. For positive pairs sourced from *D-M*, we can add the further constraint that both utterances must come from the same movie (*D-M (same)*) or from different movies (*D-M (diff)*). We use these options to create 5 challenging evaluation sets E1-E5 (Table 7.2), of increasing difficulty. More details are provided in Section 7.6.

Identification: We also demonstrate domain mismatch from VoxMovies and VoxCeleb, with speaker identification. Here we add a single linear layer on baseline models trained on VoxCeleb2 dev set for verification. The task is to fine-tune this layer with the cross-entropy loss using utterances from 485 speakers in the VoxCeleb1 dataset (using only the training data from the identification split, see Table 5 in [Nagrani et al. 2017]). We then test performance using the VoxCeleb1 identification test data (in-domain test set) and the utterances from VoxMovies (out of domain test set).

Evaluation Metrics: For verification, we report equal error rate (**EER-** %) and minimum detection cost **(minDCF)** with $C_{miss} = 1$, $C_{fa} = 1$ and $P_{target} = 0.05$. The formula of minimum detection cost function is the same as the one used in NIST SRE [*NIST 2018 Speaker Recognition Evaluation Plan* 2018] and the VoxSRC2020 evaluations. For identification, we report top 1 and top 5 % accuracy.

Eval set	I-vec. [Dehak et al. 2010]		X-vec. [Snyder et al. 2018]		Thin-R34 [Cai et al. 2018]		Thick-R34 [Heo et al. 2020]	
	EER	DCF	EER	DCF	EER	DCF	EER	DCF
VoxCb1†	5.53	0.336	3.30	0.220	2.05	0.166	1.05	0.084
VoxCb1-H†	9.13	0.467	5.82	0.352	4.37	0.283	2.39	0.154
VoxCb1-E†	5.55	0.338	3.35	0.221	2.27	0.164	1.22	0.086
VoxSRC20*	11.07	0.566	8.22	0.450	6.35	0.374	3.79	0.213
E-1	16.6	0.727	12.92	0.665	9.72	0.562	6.09	0.365
E-2	17.91	0.822	14.75	0.712	10.58	0.610	7.40	0.423
E-3	18.91	0.917	13.58	0.806	10.58	0.666	7.50	0.484
E-4	19.83	0.872	21.56	0.845	12.52	0.737	9.23	0.579
E-5	21.5	0.913	17.97	0.861	14.11	0.760	10.47	0.574

Table 7.3: **Baseline results on various VoxMovies test evaluation pairs.** We show the performance of 4 popular state-of-the-art speaker recognition models. † Cleaned versions of these evaluation pairs from the VoxCeleb1 dataset. *VoxSRC2020 validation set[1].

7.5.2 Baseline models

We compare four baseline models to investigate the performance on our dataset. All models are trained on the VoxCeleb2 dev set.

1. I-vector [Dehak et al. 2010]: Following the Kaldi [Povey et al. 2011] VoxCeleb recipe v1, we extract 400D features, followed by Probabilistic LDA (PLDA) scoring.

2. X-vector [Snyder et al. 2018]: Following the Kaldi [Povey et al. 2011] VoxCeleb recipe v2, we train a x-vector model with a PLDA back-end to extract 512D features.

3. Thin ResNet-34 [Cai et al. 2018]: Consists of 34 residual blocks (one-fourth of the channel dimensions of original ResNet-34 [K. He et al. 2016]), with self-attentive pooling. [J. S. Chung et al. 2020b] trains this network with an angular variant of the prototypical loss. We use the pre-trained model which is publicly available [J. S. Chung et al. 2020b].

4. Thick ResNet-34 [Heo et al. 2020]: This has double the number of channels in Thin ResNet-34 and uses attentive statistical pooling to model higher order statistics such as standard deviation. The model is trained with both angular prototypical loss and vanilla softmax loss to improve performance. We use the pre-trained model which is publicly available [J. S. Chung et al. 2020b]. This model currently represents the state-of-the-art on the VoxCeleb1 test sets. 512D features are extracted for both thick and thin ResNet-34 architectures.

Evaluation set	Top1 accuracy	Top5 accuracy
VoxCeleb1-test†	89.47%	97.38%
VoxMovies-(Test)	52.23%	73.31%

Table 7.4: Identification results for 485 identities on the VoxMovies-(Test) set (out of domain) and on the VoxCeleb1 test set (same domain). Note how performance drops steeply on the out of domain test set. † VoxCeleb1 test set for identification.

Eval set	Baseline	FT	S-norm	FT + S-norm
E-1	6.09	5.76	5.89	**5.66**
E-2	7.40	7.10	7.18	**7.03**
E-3	**7.50**	8.38	8.16	8.48
E-4	9.23	7.37	8.03	**7.19**
E-5	10.47	9.55	10.15	**9.35**
E-3a	**1.15**	1.53	1.29	1.58
E-3b	0.87	0.97	**0.68**	0.98
E-3c	**7.72**	9.90	9.64	10.23

Table 7.5: Domain transfer results for Thick ResNet-34. EER(%) is reported. **FT:** Fine-tuning on the VoxMovies-(Train) set. **S-norm:** Score-normalisation.

7.5.3 Domain Transfer

In this section, we implement two common domain transfer methods using the *small* amount of data provided in the VoxMovies-(Train) set.

Fine-tuning on a small amount of target domain data. We fine-tune the pretrained Thick-ResNet34 with data from the VoxMovies-(Train) set and the VoxCeleb2 dev set (overlapping speakers only). To decrease the domain gap between the datasets, we always pick one utterance from VoxMovies and another from VoxCeleb2 to form positive pairs in each mini-batch. This forces the model to decrease the distance between embeddings from the same speaker's utterances, hence reducing the domain gap during training. The network is trained with angular prototypical loss [J. S. Chung et al. 2020b], for 500 epochs using Adam with learning rate of 1e-5. Only the last fully-connected layer is fine-tuned while weights of other layers are fixed.

Score Normalisation. [Matejka et al. 2017] introduces various score normalisation techniques for test conditions with diverse domains. A *cohort* is used to estimate the amount of shift and scale for normalisation, allowing robust threshold setting. We experiment with the Z-norm, T-norm and S-norm, and find the best performance to be using the S-norm (Sec.2 in [Matejka et al. 2017]). We use the VoxMovies-(Train) set as the cohort (speakers in the VoxMovies-(Train) and

(Test) sets are disjoint, which fits the assumption of cohort selection).

7.6 Results

Verification using Baseline Models. The results for the baseline models on the different verification tasks are given in Table 7.3. The change of domain in the VoxMovies evaluation sets offers a significant challenge – the Thick ResNet-34 which achieves an impressive 1.05 EER on the VoxCeleb1 test set can only achieve 6.09 EER on the least challenging set, E-1. When comparing eval sets that share a positives or negatives source (see Table 7.2), several conclusions are made: (1) Verifying the same speaker with cross domain utterances (*D-I*, *D-M* - E-4) is harder than with utterances from the same movie (*D-M (same)* - E-1). Interestingly, this shows that the change in an actor's voice from a calm interview setting in *D-I* to strong emotions, accents and different background noise in *D-M*, is more challenging for speaker verification than the differing emotions in an actor's voice within the same movie at different points in the story-line. Furthermore, the same speaker from different movies (*D-M (diff)* - E-5) is harder still, showing that an actor's voice changes most between different movies. (2) Negative pair verification is hardest when the negatives are both taken from the unseen domain, *D-M* (E-4). This is more difficult than when they are taken from *D-I* (E-3) or *D-I*, *D-M* (E-2), which are of roughly equal difficulty to the Thick ResNet-34. This is largely because the baseline models were trained on *D-I*, so any source of negatives or positives exclusively from that domain will be less challenging.

Identification. We use the Thin ResNet-34 model for identification. Table 7.4 shows identification accuracy on both domains. As expected, the identification accuracy drops significantly from VoxCeleb1-test to VoxMovies-(Test) by 37.24% (top1% acc.). Anne Hathaway is one of the hardest to identify (top1 acc. 29.17% - down from 90% in VoxCeleb1-test), whereas Samuel L. Jackson is easier (acc. 100% drops to 72.88% in movies). This may be due to Hathaway's multiple accents in her movies (Fig. 7.1).

Domain Transfer. We report the results in Table 7.5. Fine-tuning with VoxMovies-(Train) reduces the EER on most of the evaluation sets. The largest improvement is seen in evaluation sets that showed the worst performance in baseline experiments, namely by 2.04% in E-4 and 1.12% in E-5. Both E-4 and E-5 source

negatives from *D-M* , showing that fine-tuning and score normalisation work well when transferring existing models from *D-I* to *D-M*. E-3 on the other hand, which sources negatives from *D-I* shows performance degradation. We conclude that fine-tuning on *D-M* degrades the performance on negative pairs only from *D-I*. To verify this conclusion, we introduce three new evaluation sets, E-3a (positives: *D-I*, negatives: *D-I*), E-3b (positives: *D-I*, negatives: *D-I*, *D-M*), E-3c (positives: *D-M*, negatives: *D-I*) in Table 7.5. The fine-tuned model becomes worse at verification in *D-I* (as shown by the degradation of E-3a). Performance on negatives from *D-I* contribute most to this (as shown by the degradation of E-3c, relative to E-3b).

7.7 Conclusion

In this paper, we provide a novel speaker recognition dataset from movies called VoxMovies which contains almost 9,000 utterances with diverse emotion, accents and background conditions. We demonstrate that state-of-the-art models trained on interview data from VoxCeleb degrade significantly on cross-domain evaluation sets from VoxMovies and while simple domain adaptation techniques boost performance, there is still large room for improvement. We hence encourage the research community to develop new methods and systems for this challenging new domain.

Chapter 8

Epic-Sounds: A Large-scale Dataset of Actions that Sound

The paper has been accepted for publication at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2023.

Epic-Sounds: A Large-scale Dataset Of Actions That Sound

Jaesung Huh[1*] Jacob Chalk[2*] Evangelos Kazakos[2]
Dima Damen[2] Andrew Zisserman[1]
[1]VGG, University of Oxford [2]University of Bristol

Abstract

We introduce Epic-Sounds, a large-scale dataset of audio annotations capturing temporal extents and class labels within the audio stream of the egocentric videos. We propose an annotation pipeline where annotators temporally label distinguishable audio segments and describe the action that could have caused this sound. We identify actions that can be discriminated purely from audio, through grouping these free-form descriptions of audio into classes. For actions that involve objects colliding, we collect human annotations of the materials of these objects (e.g. a glass object being placed on a wooden surface), which we verify from visual labels, discarding ambiguities. Overall, Epic-Sounds includes 78.4k categorised segments of audible events and actions, distributed across 44 classes as well as 39.2k non-categorised segments. We train and evaluate two state-of-the-art audio recognition models on our dataset, highlighting the importance of audio-only labels and the limitations of current models to recognise *actions that sound*.

*Equal technical contribution.

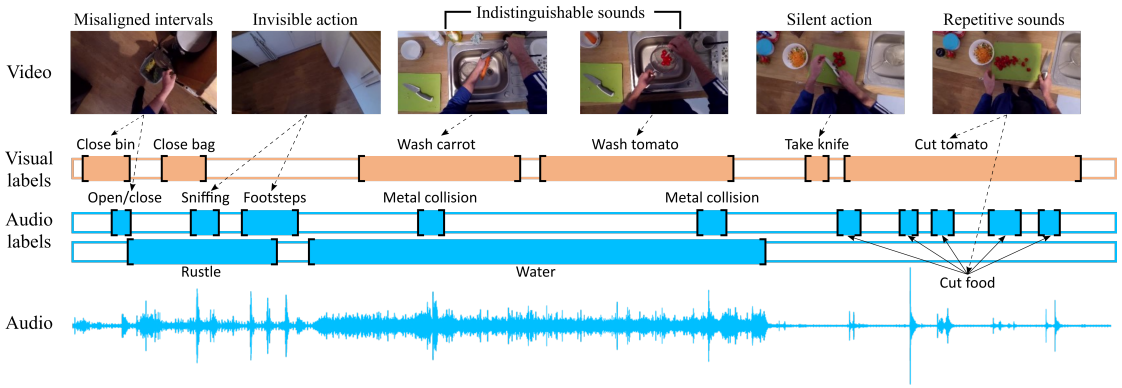


Figure 8.1: Sample video with corresponding audio from EPIC-KITCHENS-100 [Damen et al. 2022]. We compare the already published **visual labels** with our collected Epic-Sounds **audio labels**. We demonstrate the differences between the modality annotations, both in temporal extent and class labels, highlighting: **Misaligned intervals:** temporal boundaries are distinct; **Invisible action:** action not seen in the video, but which produces distinct sounds (0-to-1 matching); **Indistinguishable sounds:** sounds from two distinct visual actions, but are audibly inseparable; **Silent action:** visual action that does not have audible sounds (1-to-0); and visual actions containing multiple **repetitive sounds** (1-to-N).

8.1 Introduction

Humans perceive objects and actions through multiple senses, especially vision and audition [Smith and Gasser 2005]. Inspired by this, a plethora of works aim to solve various video understanding tasks, such as action recognition [Gao et al. 2020; Nagrani et al. 2020c; Nagrani et al. 2021] and detection [Tian et al. 2018; Bagchi et al. 2021], by fusing the two modalities. These attempts are especially common for egocentric video datasets due to the camera's close proximity to the ongoing actions resulting in clearer inputs, both visually and audibly. Research has shown improved performance by using audio and video jointly in egocentric data [Kazakos et al. 2019; Kazakos et al. 2021a; Ramazanova et al. 2023; Xiong et al. 2022].

In general, these works make two key incorrect assumptions: First, that the visual and auditory events temporally coincide; Second, that a single set of classes can be used for both modalities, typically derived from vision. In practice, visual and auditory events exhibit varied levels of both temporal and semantic congruence, thus violating these assumptions (See Fig. 8.1). In the case of actions such as 'close bin', the onset of the visual event can be defined as the time that the person grasps the handle, whereas the onset of the audio event is delayed to the moment when the lid of the bin slams. Some actions are audibly indistinguishable, e.g. 'wash

Name	Source	# hrs	# seg.	# cls	Modality	T	D
DESED [Turpault et al. 2019]	real + synth.	43h	8k	10	**A**	✓	N/A
URBAN-SED [Salamon et al. 2017]	synth.	30h	50k	10	**A**	✓	N/A
TUT 2016 [Mesaros et al. 2016]	real	2h	6.3k	18	**A**	✓	N/A
AudioSet [Gemmeke et al. 2017]	YouTube	5833h	1.8M	632	**A** + **V**	✗	✗
VGG-Sound [H. Chen et al. 2020]	YouTube	550h	200k	309	**A** + **V**	✗	✗
SSW60 [Van Horn et al. 2022]	real	25.7h	9.2k	60	**A** + **V**	✗	✗
LLP [Tian et al. 2020]	YouTube	33h	19.4k	25	**A** + **V**	✓	✗
Epic-Sounds	home kitchens	100h	78.4k	44	**A** + **V**	✓	✓

Table 8.1: Comparison to existing datasets. **A**: Audio. **V**: Video. **T**: Temporal annotations. We showcase that Epic-Sounds is the only dataset with distinct classes for audio and video modalities (**D**). We only report categorised segments of Epic-Sounds here.

carrot' vs 'wash tomato', as it is impossible to determine which vegetable is being washed through sound alone. Consequently, using the visual temporal labels as targets for training an audio classifier is often a flawed endeavour – the resulting audio classifier will not be able to discriminate all of the visual events; and many audio labels that could provide supervision for training are missed. Based on these observations, we crowdsource temporal and semantic labels for the audio of EPIC-KITCHENS-100 that are distinct from the visual ones.

However, as evidence suggests [VanDerveer 1979], humans perform poorly at recognising objects and events using audio alone, making their annotation using only audio challenging. Due to the lack of sufficient information in audio for inferring fine-grained properties of events, humans tend to use vague terms for describing them; *e.g.* when the interaction from the collision of two objects is indistinguishable from audio, annotators often describe the associated event as 'clang' or 'bang'. To alleviate this, we further augment these semantics with the *materials* of the objects that interact. We verify these from the video, discarding incorrect audio-only material annotations.

In summary, we introduce Epic-Sounds, a large-scale dataset of daily-life sounds, derived from the audio of EPIC-KITCHENS-100. Epic-Sounds contains 78,366 categorised sound events spanning over 44 categories, as well as 39,187 non-categorised sound events, totalling 117,553 sound events across 100 hours of footage collected in 700 videos from 45 home kitchens. The sound classes are based on descriptions from only listening to audio, thus suitable for problems in acoustics such as audio/sound recognition and sound event detection. Epic-Sounds is available from: https://epic-kitchens.github.io/epic-sounds.

8.2 Related Work

Sound event detection datasets. Sound Event Detection (SED) is the task of detecting the onset and offset of audio events as well as recognising the event within the detected boundaries. SED datasets [Turpault et al. 2019; Salamon et al. 2017; Mesaros et al. 2016] are similar to Epic-Sounds as these include annotations of temporal boundaries of events, whereas sound recognition datasets [Piczak 2015; Moreaux et al. 2019; Fonseca et al. 2022] do not. Nevertheless, they differ from Epic-Sounds in several aspects. First, they are of smaller scale making the training of modern architectures impractical. Second, [Turpault et al. 2019] and [Salamon et al. 2017] contain synthetic audio, and therefore models trained on these datasets generalise poorly to real recordings. Third, [Turpault et al. 2019; Salamon et al. 2017; Mesaros et al. 2016] contain sounds associated with generic scenes and events, whereas Epic-Sounds focuses on fine-grained sounds generated from diverse audible events in 45 home kitchens.

Audio-visual datasets. We compare Epic-Sounds to publicly available sound recognition or detection datasets in Table 8.1. AudioSet [Gemmeke et al. 2017] is the largest audio-visual dataset of audio events with 2.1M clips and 527 annotated classes, while VGG-Sound [H. Chen et al. 2020] contains over 200K video clips and 300 audio classes. They are both collected from YouTube and each audio clip is 10s long. Both do not have temporal annotations for events, and importantly, a single set of annotations is collected for both modalities. The LLP dataset [Tian et al. 2020] is the closest to ours, in that both visual and auditory events are annotated independently, providing separate temporal segments. However, unlike ours, both modalities still share the same label set. Also, LLP is of smaller scale and contains diverse events while Epic-Sounds focuses on sounds resulting from actions.

Fine-grained audio-visual datasets. The PACS dataset [S. Yu et al. 2022] focuses on understanding the physical common sense attributes of objects shown in the video, which is similar to our 'material' based annotation procedure. However, these attributes are distinguished by 13.4K question-answer pairs; displaying the video with and without audio, and then querying a variety of physical properties. SSW60 [Van Horn et al. 2022] consists of 31K images, 3.8K audio and 5.4K videos of 60 species of birds, proposed to facilitate works on fine-grained categorization

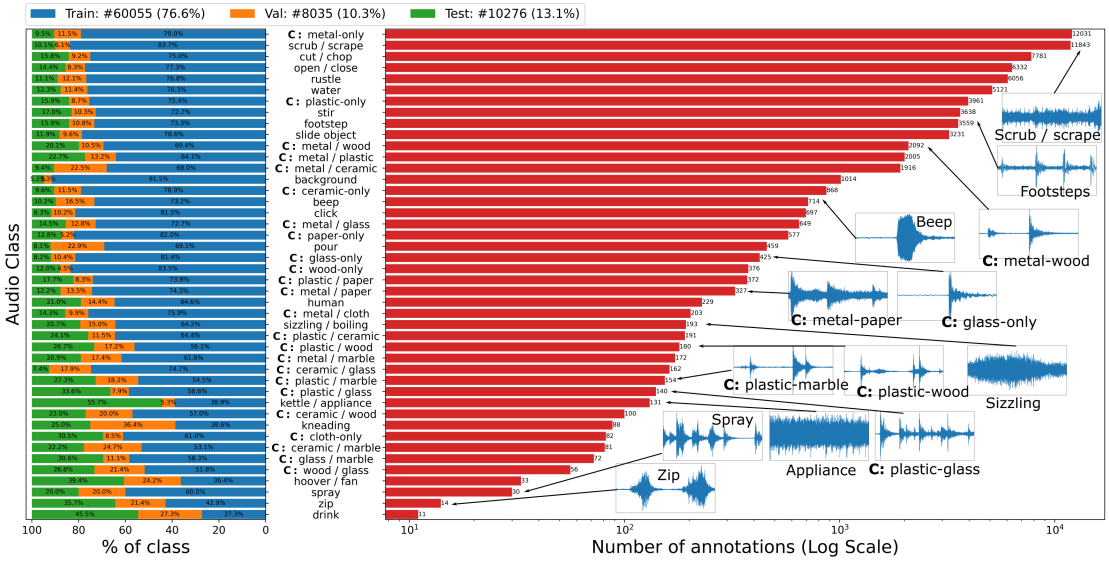


Figure 8.2: Left: The percentage distribution of each audio class across the Epic-Sounds dataset splits. Right: Class frequencies showcasing the long-tail distribution. **C:** represents a collision-based sound between objects of the same or two distinct material types.

using audio-visual fusion. Both datasets do not contain temporal annotations of sounds.

8.3 Epic-Sounds: dataset statistics

EPIC-KITCHENS-100. EPIC-KITCHENS-100 [Damen et al. 2022] is a large-scale egocentric audio-visual dataset which contains 100 hours of videos containing unscripted daily activities and object interactions in people's kitchens. It consists of 700 videos and 89,977 segments describing visual actions that occur. Actions consist of verb and noun labels, where there are 97 verb classes and 300 noun classes. The average action length is 2.6s. Since these actions are based only on video, we emphasise that we do not refer to any of these labels during the annotation process.

Epic-Sounds. The dataset consists of 78,366 categorised temporal annotations with an average length of 4.9s, distributed across 44 classes. We match the train / val / test splits from EPIC-KITCHENS-100, giving the per-class proportion across splits in Fig. 8.2 (left). We divide the test split into two roughly even subsets: one for audio-based interaction recognition, and one for audio-based interaction detection. We release start/end times for the recognition subset, and keep those for the detection hidden for the relevant challenge.

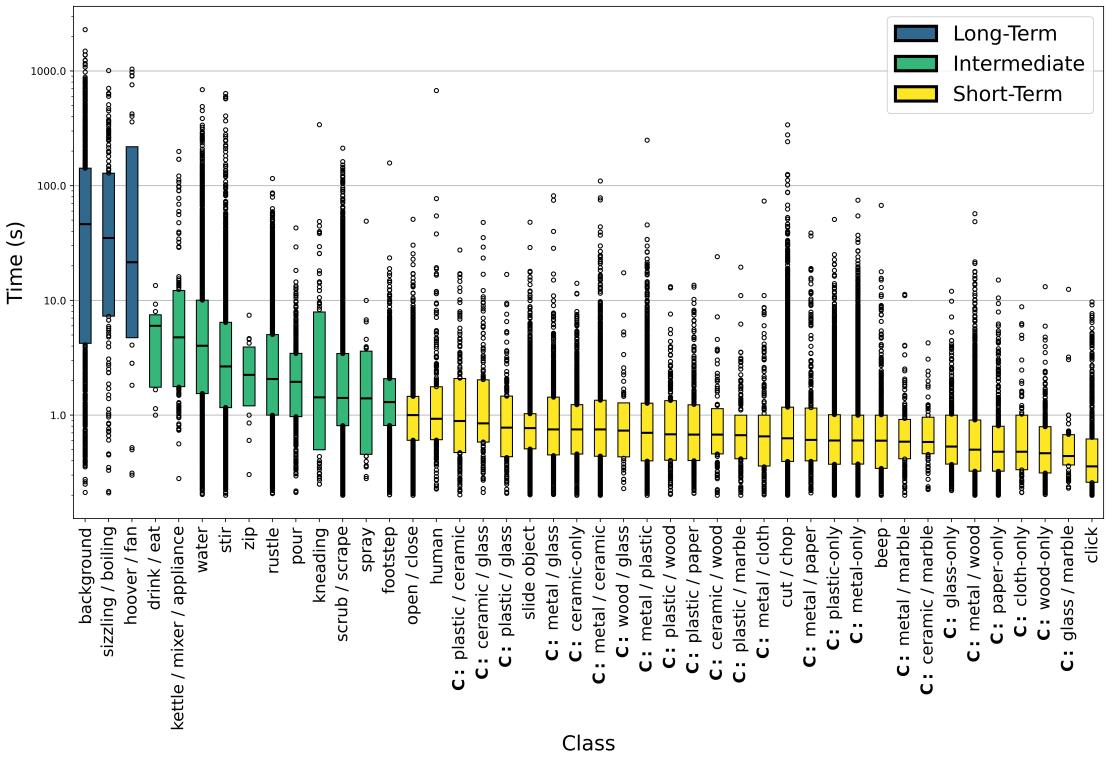


Figure 8.3: Box plot for the lengths of the annotations over classes, ordered by the median of their lengths. The majority of the classes, 30 (68%) are short-term, 11 (25%) are intermediate classes and only 3 (7%) are considered long-term (median $>$ 10s). **C:** collision-based sounds between objects of the same or two distinct material types.

Class frequency is also shown in Fig. 8.2 (right), highlighting that Epic-Sounds is naturally long-tailed. We also visualise the waveforms for a sampled subset of the classes. Here, there are both classes which produce waveforms consistent with short-term, percussive sounds such as all the collision-based classes, as well as long-term sounds e.g. sizzling. We also visualise the length of the annotations distributed across the classes in Fig. 8.3. Here, we sort the classes by the median of their lengths, $\tilde{t}$, and distinguish three categories: long-term ($\tilde{t} \geq 10s$); intermediate ($1s < \tilde{t} < 10s$); and short-term ($\tilde{t} \leq 1s$) classes. Long-term classes relate to lengthier activities, such as cooking and hoovering. In the intermediate classes, there are sounds such as scrub / scrape, or rustle, and then near instantaneous/percussive sounds in the short-term category, including all collision-based classes.

8.4 Data collection Pipeline

The data collection process is conducted through the collection of temporal segments of distinct sounds, described by free-form vocabulary, followed by clustering generic sound categories into distinct classes. This section details this process, as well as post-processing steps taken to refine the results.

8.4.1 Data collection of labelled temporal segments

The objective is to annotate all the distinctive audio events that occur across all the videos in EPIC-KITCHENS-100. The annotation consists of the temporal interval of the event, together with a free-form text description. As the video length in this dataset varies greatly, from 30 seconds to 1.5 hours, we trim the videos into a series of manageable lengths for annotations of 3-4 minutes. We deem our decision to only provide the audio stream as a key step so the annotators focus on the temporal bounds of the acoustic event alone, rather than being biased by visual and contextual information present in the video stream (consider the 'misaligned intervals' example shown in Fig. 8.1, where visual and auditory temporal segments do not align for the same event). However, the annotators are provided with the plotted audio waveform to act as a visual guide to assist in targeting specific audio signatures and streamline the annotation process.

Instructions to the annotators. We worked with 20 annotators hired from an annotation company. We use a customised version of the VIA tool [Abhishek Dutta and Zisserman 2019] to gather the annotations. Annotators are asked to listen to the audio and detect any distinctive audio event. They then are instructed to mark the start and end time of each distinctive sound they hear. Each segment is then given a semantic label which best describes the annotator's perception of the action associated with the audio event. We impose no restriction on the vocabulary used, so the annotators may describe this however they wish. As a guide, we provide a list of sound labels that commonly occur in daily life, which the annotator may refer to, though they are not required to explicitly choose from this list. We term a segment-label pair as an 'audio annotation'. A second annotator performs quality assessment to the audio annotations produced by the first annotator, particularly

focusing on any missed audio events.

For each unique label description, the VIA tool creates a separate time-line, effectively grouping sequences of the same event. Note that sound events can overlap in time. If two segments are less than 0.3s apart, we instruct the annotators to merge the two segments as we deem them to belong to the same event. Additionally, annotators are asked to identify consistent background sounds (or noise) that occur throughout a large portion of the audio (e.g. radio, fan or washing machine). The annotators were asked to tag these as 'background'. The procedure described thus far resulted in the annotation of 556 distinct sound descriptions.

Humans tend to use abstract words to describe sounds, such as 'clang' or 'clatter', especially for those generated from the collisions between objects. To address this, we use a customised LISA [*VGG List Annotator (LISA)* 2022] annotation interface for annotating the material of the objects that collide based on audio. We instruct annotators to select from a pre-specified list which *materials* are involved in the collision. This list, along with examples of objects for each material, are provided in Table 8.2. These cover all the materials popular in kitchens.

Annotators are encouraged to select one or more materials, or mark the material as indistinguishable by choosing the 'Can't tell' option. We drop the instances in the latter case – as we believe these are unhelpful for sound or event understanding tasks. However, some material sounds might be deceiving. For example, one might perceive the material collision to be between a glass and a wooden object, but in fact it's food poured into a ceramic container. We thus ask annotators to then visually verify their material annotations using the corresponding video. Importantly, annotators have to listen and choose the perceived material first, and cannot change these after watching the video. Instead, they select the actual materials involved when viewing the video. We only retain visually-verified collision sounds – i.e. materials correctly perceived from the audio only, then verified from the visual observation. We choose all collision material labels for which at least 40 examples are present. As a result, abstract labels related to collision (e.g. 'clang/-clatter', 'put objects on surface') are clustered into 24 sound categories describing the materials involved, such as **C**: metal-only, or **C**: plastic-wood. We use the letter **C** to indicate these are collision-based classes.

Material	Example objects	# of times selected
Metal	metal or stainless steel	15523
Plastic	plastic bowl, plastic container	5464
Ceramic	ceramic cup, plate	2634
Wood	wooden spatula, wooden table	2408
Paper	kitchen roll, cardboard boxes	1253
Glass	wine glasses, glass cup	1248
Stone / Marble	kitchen worktops, marble tables	377
Cloth	towels, teatowels, clothes	257
Others	materials not listed above (e.g. food)	3596
Can't tell	cannot determine the material	10030

Table 8.2: Material options for collision sounds. We note # **of time** each material was **selected** in collision sounds, and discard the sounds annotated with 'Others' or 'Can't tell'.

8.4.2 Post-processing Annotations

From labels to classes. We post-process the audio labels to fix spelling errors and group semantic equivalences. For example, sounds like 'buzzer', 'beep' and 'alarm' are grouped into one *beep* class. Similarly, sounds described by the verbs 'wipe', 'scour', 'scrape' and 'scrub' are also grouped into a single class. We also manually review tail instances to determine whether these form novel classes or should be merged with others. In cases where the description was not meaningful, the categorised annotation is dropped. For example, the sound 'spray' was considered a meaningful tail instance of an action that sounds. In contrast, the label 'dog barking' was discarded as it is not relevant to our context. This produces the 44 audio classes, as shown in Fig. 8.2.

Error checking audio classes. Due to differences in sound perception between annotators, some errors exist amongst the classes. For example, where one annotator hears a drawer being pulled and hence labels 'open / close', another may hear 'drag object' for a similar audio. To resolve such errors, we manually review each of the labels in the test and validation set. Specifically, the following procedures are conducted to correct samples in the validation and test set. (1) We ask annotators to manually review all the val / test samples, providing them only sounds for non-collision classes and sounds and corresponding video clips for collision classes. (2) We collect the samples in which the first and second annotations are inconsistent, and ask a new set of annotators to manually choose whether the 1st or 2nd annotations are correct. The annotators could choose 'can't tell' or 'neither of the two'. (3) We manually verify those decisions, and removed the samples from the

Split	Model		Top-1	Top-5	mCA	mAP	mAUC
Val	Chance	-	7.71	30.95	2.29	0.023	0.500
	SSAST [Yuan Gong et al. 2022]	L	28.74	64.87	7.14	0.079	0.755
	ASF [Kazakos et al. 2021b]	L	45.53	79.33	13.48	0.172	0.789
	SSAST [Yuan Gong et al. 2022]	F	53.47	**84.56**	**20.22**	0.235	**0.879**
	ASF [Kazakos et al. 2021b]	F	**53.75**	84.54	20.11	**0.254**	0.873
Recognition Test	Chance	-	7.85	31.91	2.39	0.024	0.500
	SSAST [Yuan Gong et al. 2022]	L	29.93	66.60	7.17	0.082	0.725
	ASF [Kazakos et al. 2021b]	L	45.00	78.98	15.00	0.183	0.788
	SSAST [Yuan Gong et al. 2022]	F	53.71	84.54	**22.28**	0.223	0.820
	ASF [Kazakos et al. 2021b]	F	**54.45**	**85.17**	20.41	**0.254**	**0.852**
Entire Test	Chance	-	7.22	30.11	2.27	0.023	0.500
	SSAST [Yuan Gong et al. 2022]	L	27.50	65.55	6.68	0.080	0.741
	ASF [Kazakos et al. 2021b]	L	44.55	78.44	14.49	0.145	0.772
	SSAST [Yuan Gong et al. 2022]	F	53.75	83.76	**20.76**	**0.237**	**0.860**
	ASF [Kazakos et al. 2021b]	F	**54.86**	**84.26**	20.30	0.232	0.823

Table 8.3: Results of the Baseline Models on the Epic-Sounds validation, recognition test and entire test splits. L: Linear-Probe; F: Fine-Tuning.

val/test sets. For the training set, we utilise the overlaps between audio segments and visual segments to select the samples for reviewing. We deem the use of the visual labels acceptable for error correction, as the annotation process is complete. Thus, utilising the visual labels for post-processing no longer compromises the issues stated in Fig. 8.1.

We review all audio classes for which there exists a *mapping* to visual classes in EPIC-KITCHENS-100. We identify two types of mapping, trivial; the audio class itself already exists as a visual class e.g. 'scrub', and relational; the audio class does not exist as a visual class itself but can be semantically mapped to one or more of the visual classes, such as the audio class 'click' relating to the verb 'turn on/off' or the noun 'light switch'. We consider all annotations *not* labelled as the audio class of interest, but which overlap with action clips containing its visual mappings. We then manually assess each overlapping annotation, through listening to the audio and determining the label's correctness. We run this error checking cycle multiple times to ensure all incorrectly classified instances are accounted for.

Non-categorised audio events. As a result of post-processing, there are audio events that we recognise the sound exists but no semantic label matching the 44 classes could be given. These are samples we either could not assign class labels, or collision sounds for which they could not be visually verified. We release these temporal boundaries of these 39,187 samples as *non-categorised*.

8.5 Experiments and Results

This section describes how two state-of-the-art sound recognition models perform on classifying Epic-Sounds. We assess models through performance metrics and class confusion matrices.

Baselines. We train and evaluate the Auditory SlowFast (ASF) [Kazakos et al. 2021b] and Self-Supervised Audio Spectrogram Transformer (SSAST) [Yuan Gong et al. 2022] audio encoder networks, with both a linear probe, i.e. by freezing the model weights and only training the last classification layer, and by fine-tuning. We also compare to a chance baseline. ASF is pretrained on VGG-Sound, and SSAST is pretrained on AudioSet and LibriSpeech [Panayotov et al. 2015].

Audio processing. We follow the audio processing of [Kazakos et al. 2021b] for extracting the input spectrograms for both models, noting that this outperformed the default audio processing of SSAST (200×128 spectrograms for 2s of audio, or 400×128 for 4s of audio sampled at 16kHz). Namely, audio is resampled at 24kHz for both models. We randomly sample 2s of audio to create log-mel-spectrograms with 128 Mel bands. If the audio annotation is shorter than 2s we pad the produced spectrogram with its last column. We use a window and hop size of 10ms and 5ms respectively, resulting to a spectrogram of size 400×128.

Training & Validation Configuration. We train both models for 30 epochs, setting the initial learning rate to 1e$-$3 for ASF which decays to 10% on epoch 25 and 1e$-$4 for SSAST, which is warmed up from 1e$-$6 for 2 epochs and decays to 5% then 1% on epochs 10 and 20. Both models are trained with cross-entropy loss, optimising ASF using SGD with Nesterov momentum equal to 0.9, and SSAST using AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$. Both models use a weight decay of 0.0001 and a batch size of 128. We use a base 384×384 ViT with patch size 16 as the backbone for SSAST and the 8×8 ResNet50 variant of ASF. For data augmentation, SpecAugment [D. S. Park et al. 2019] is used, again following [Kazakos et al. 2021b], using two frequency masks with $F = 27$, two time masks with $T = 25$ and time warp with $W = 5$. We use test augmentations simliar to [Kazakos et al. 2021b], dividing the audio into 5 equally sized sub-clips and then averaging their individual predictions from the networks. For the linear probe results, we freeze the backbone of both SSAST and ASF and train only the last linear layer with

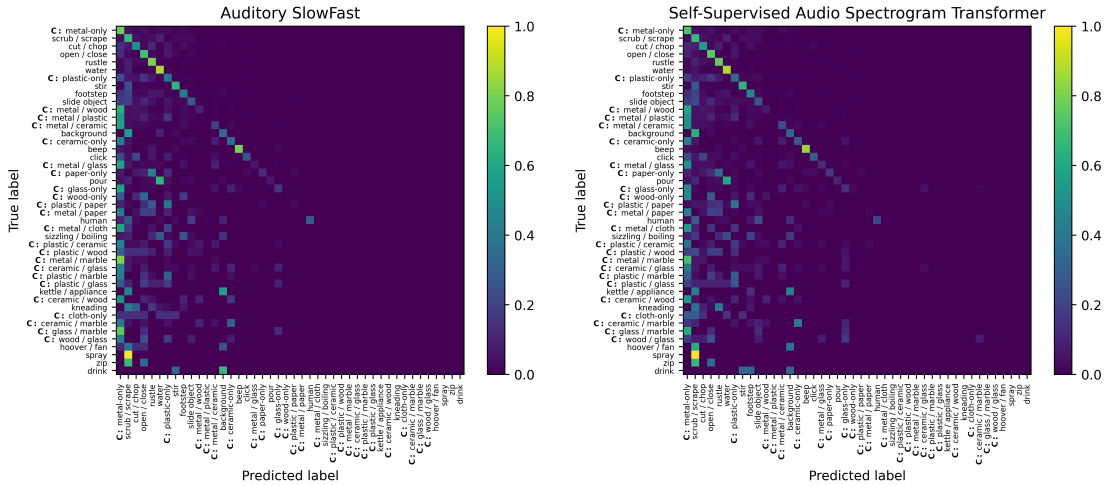


Figure 8.4: Confusion Matrices on Val for ASF (left) and SSAST (right).

the same training hyperparmeters and pretrained backbones as before.

Evaluation Metrics. We report the top-1 and top-5 accuracy, as well as mean average precision (mAP), mean area under ROC curve (mAUC), and mean per class accuracy (mCA), for both the validation and test sets.

Results. We report quantitative results for both models in Table 8.3. Overall, ASF outperforms SSAST by 0.28%, 0.74% and 1.11% for top-1 accuracy on the validation, recognition test and entire test set respectively. ASF exhibits better mAP for the validation set and recognition test set, whereas SSAST performs better on the entire test set, suggesting these models share a similar level of robustness to the long-tailed data. The performance of the linear probe drops significantly compared to fine-tuning results for ASF and almost halves for SSAST. In the latter case, we note that self-supervision alone does not learn class-discriminative features.

Fig. 8.4 shows the validation confusion matrices for finetuned ASF and SSAST. We see that both models are able to detect a subset of distinctive, unique sounds such as rustle, water and beep. Concerning the collision-based classes, both models tend to classify uni-material collisions more successfully than bi-material collisions, but generally produce a false positive prediction of the metal-only collision class, suggesting that the models may struggle to detect how material properties alter the sound produced from a collision.

Reflections. When comparing audio to video labels, we reflect on our motivation

in Fig. 8.1. The top-3 audio classes that have 1-to-0 overlap with visual classes are: wood / glass collision (51.78%), metal / marble collision (51.67%), glass / marble collision (51.39%). In this instance, the classes relate to sounds produced by visual actions such as placing objects which are occasionally deemed trivial, or happen off-screen, resulting in missed visual annotations while still producing distinctive auditory signals. The top-3 video classes that have no overlap with audio (0-to-1) are: take basket (71.43%), pour basil (71.43%) and brush oil (70.0%) – these are silent actions. The top-3 many-to-1 classes that contain repeated audio sounds (on average) are: cut / chop (1.77-to-1), beep (1.31-to-1), metal-only collision (1.18-to-1), these relate to actions that have a 'stop-start' pattern e.g. pauses between chops, button presses on an appliance, or between repetitively moving items in a cutlery draw or sink.

8.6 Conclusion

In this paper, we present a large-scale dataset, Epic-Sounds, which consists of 78.4k categorised segments and 39.2k non-categorised segments, totalling 117.6k segments spanning 100 hours of audio, capturing diverse actions that sound in home kitchens. Sound categories are annotated based on audio human descriptions. We also provide benchmark classification performance using the state-of-the-art sound recognition networks. The audio annotations in this dataset enable a veridical evaluation of audio classifiers, and can replace the current evaluations based on visual annotations. We anticipate that multi-modal approaches will benefit from these audio labels. The dataset can also be used for audio event detection, though we have not evaluated that in this work.

Chapter 9

Discussion

This chapter starts with an overview of the key achievements and impact of the research presented in this thesis (Section 9.1). Subsequently, we explore potential directions for future investigation and research (Section 9.2).

9.1 Achievements and Impact

Action recognition from egocentric untrimmed videos. In Chapter 2, we propose the Multimodal Temporal Context Network (MTCN) which utilises the neighbouring actions to improve the action recognition in the interval we are interested in. With minimal additional information which is the temporal boundary of neighbouring actions, MTCN outperformed the state-of-the-art methods at the time the paper was published.

This work introduces a novel masked language model for the first time which uses only the labels of action sequences and achieves further improvement by combining the predictions from an audio-visual transformer. Since 2021 when the paper is published, it has gained 50 citations until September 2024 and inspired a series of follow-up work studying efficient multimodal fusion techniques [Y.-B. Lin et al. 2022; Xinghang Hu et al. 2022; Hanxin Wang et al. 2023] and egocentric action recognition [Plizzari et al. 2022; Radevski et al. 2023]. The model achieved 4th place in EPIC-KITCHENS-100 action recognition challenge in 2021 [Damen et al. 2021].

In Chapter 3, we introduce A Time Interval Machine (TIM) which uses the ac-

tion's time interval as a query for audio-visual action recognition. Audio-visual transformer learns to recognise the action from its interval and the unaltered surrounding context. We jointly train the model on modality specific time intervals and label sets, allowing the TIM to recognise multiple events across both visual and auditory modalities.

TIM outperformed the audio-visual action recognition networks on EPIC-KITCHENS-100 that leverage large language models [Y. Zhao et al. 2023; Y. Zhao and Krähenbühl 2023] or additional private datasets [Yan et al. 2022; Xiong et al. 2022]. TIM achieved 2nd place in EPIC-KITCHENS-100 action recognition challenge in 2024 pretraining only on public data. It also showed promising directions for sound detection and action detection, achieving 2nd and 3rd place in the sound detection and action detection tracks of the EPIC-KITCHENS-100 challenge, respectively.

Character audio-visual subtitling. In Chapter 4, we first propose a new task, character-aware audio-visual subtitling, which aims to generate a full transcript of the dialogue, with precise speech timestamps, and the character speaking identified. Our method only uses off-the-shelf deep neural network models which allows people to use it easily without the need for training. In Chapter 5, we advance the previous work by leveraging the temporal context of dialogue within a scene. Inspired by the success of Large Langauge Model (LLM), we show for the first time that asking the LLM to predict the speaker of the utterance can improve the character recognition performance on short segments, which the state-of-the-art speaker recognition models suffers from.

To the best of our knowledge, these are the pioneering works which introduce the pipeline for character-aware audio-visual subtitling task to perform speech recognition, speaker diarisation and character recognition at the same time. Our method introduces speaker diarisation method *without* using face recognition or tracking models, which is a common practice in other literature [M.-K. He et al. 2022; E. Z. Xu et al. 2022]. We show the promising steps towards Subtitles for Deaf and Hard-of-hearing (SDH) [Szarkowska 2020] and can be extended to full-length movies in the future. It also demonstrates the generalisation ability of our method to various genres of TV shows including comedies, dramas, sci-fi or horror.

Audio-visual datasets. In Chapter 6, we introduce VoxConverse, a large-scale speaker diarisation dataset from Youtube videos. We also introduce a novel audio-visual diarisation pipeline which consists of active speaker detection, speech enhancement, and speaker recognition models.

The paper is published in Interspeech 2020 and has received 165 citations until September 2024. VoxConverse is still the largest public speaker diarisation dataset which advances the development of new speaker diarisation methods [Kwon et al. 2021; Hongji Wang et al. 2023; Bredin 2023]. It also promotes the other fields of speech processing such as speaker verification [J.-w. Jung et al. 2023] and active speaker detection [Y. J. Kim et al. 2021; C. Jung et al. 2024]. The data creation pipeline inspired the development of other valuable diarisation datasets [Grauman et al. 2022; Kwak et al. 2024], further advancing research in audio-visual diarisation.

In Chapter 7, we introduce VoxMovies, a large-scale speaker recognition dataset from movies. This is the first and only dataset that contains utterances from movie characters with a diverse range of emotions and background noises. We also show that simple domain adaptation methods can help improve the performance of speaker models trained on different domains on this dataset.

The paper was published in International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2021 and has received 34 citations as of September 2024. Researchers have primarily worked with the VoxCeleb dataset [Nagrani et al. 2017], which has become the de facto speaker recognition evaluation set. However, due to its limited domain in celebrity interviews and news clips, state-of-the-art models [Y. Zheng et al. 2023; Torgashov et al. 2023] achieve $< 0.5\%$ Equal Error Rate (EER) on its evaluation set but do not generalise well to other domains. VoxMovies has been instrumental in evaluating models' generalisation abilities across diverse emotions and speaking styles [Xuechen Liu et al. 2021; Xuechen Liu et al. 2022; Hong et al. 2023].

Both VoxMovies and VoxConverse were used as a key component of the VoxCeleb Speaker Recognition Challenges (VoxSRC) [Huh et al. 2024]. The series of VoxSRC challenges aimed to advance the research on speaker recognition and diarisation by providing a common test sets to the community. VoxConverse was used as a

validation set of the speaker diarisation track from year 2020 to 2023 and 91 teams participated in this track over this period. VoxMovies was used to create more challenging validation and test sets in speaker recognition tracks in the VoxSRC 2020 challenge. Existing speaker recognition datasets primarily consisted of recordings from celebrity interviews or news clips, where speakers typically speak calmly in controlled environments. VoxMovies serves as an out-of-domain dataset and covers a wide range of emotions and background noises which are common in movies.

Chapter 8 introduces Epic-Sounds, a large-scale sound recognition dataset from the egocentric videos in EPIC-KITCHENS-100. This work challenges two false assumptions: (i) the visual and auditory events temporally coincide, and (ii) a single set of classes can be used for both modalities. Prior work in audio understanding assumes that visual labels offer good training for the auditory stream in egocentric videos [Kazakos et al. 2019; Kazakos et al. 2021a]. These assumptions are shown to be incorrect, and highly problematic, through a thorough, extensive and innovative data annotation pipeline.

So far, Epic-Sounds has had an impact in many different research areas such as (i) the development of new sound recognition models for everyday events [Bhosale et al. 2023; Piergiovanni et al. 2024] and (ii) a new line of research in audio-visual learning dealing with two modalities with different temporal and semantic labels, such as action recognition (See Chapter 3) and sounding action discovery / retrieval [C. Chen et al. 2024]. The dataset has also been used as a sound interaction recognition challenge [1] from year 2023 to 2024 with 15 participants from different sites. It has great potential to stimulate research in multimodal learning methods that deal with different annotations from each modality.

9.2 Future Works

Long-term video understanding. Although MTCN (Chapter 2) and TIM (Chapter 3) aim to leverage the temporal context for action recognition, there is still room for improvement towards long-term video understanding. MTCN requires additional information, time intervals of neighbouring actions, which are not always available. Also, TIM shows the best performance on 30 seconds of video

[1] https://codalab.lisn.upsaclay.fr/competitions/9729

segments, and it is still challenging to generalise to the longer video (*e.g.* 1 hour). There are a few challenges towards longer video understanding: (i) the computational constraints of processing extended video sequences, as it becomes difficult to construct models that can consume the full context of long videos within GPU memory limitations, and (ii) the challenge of maintaining long-range dependencies and capturing relevant information across distant time steps in lengthy videos. One approach to address this is to introduce a memory vector to store the information of the video from the past and update it when the model processes the later part of the video. There are a few works that utilise this concept in NLP [Z. Dai et al. n.d.] and computer vision [C.-Y. Wu et al. 2022] but there still remains a challenge in determining if these models can also work in multimodal setting. Another approach we can take is using Video-based LLMs (VideoLLMs) [Luo et al. 2023; Weng et al. 2024], borrowing the power of these models trained with enormous amounts of video and language data from the internet. [Weng et al. 2024] shows promising results on zero-shot video question-answering datasets, but it is unknown whether the model can show similar performance when it also consumes an additional modality.

Character audio-visual subtitling. Chapter 4 and Chapter 5 introduce a novel pipeline for character-aware audio-visual subtitling. There are still several limitations in our method that can be improved in the future. First, our method still requires additional metadata of cast list for each episode, which might not be available for some TV shows. However, a human can recognise the speaker identity even without the metadata by watching the video. There exist some prior works [Sue E Tranter 2006; Bredin et al. 2014] which use the textual information in speech transcripts to assign the speaker of each utterance. We can adopt this idea and integrate it into our method. Second, our method consists of multiple stages, each with an individual model, which might cause error propagation. Our pipeline consists of voice activity detection [Bain et al. 2023] followed by audio-visual speaker detection [Afouras et al. 2020b] and audio-only speaker recognition [Desplanques et al. 2020]. We observed that if the first stage of voice activity detection makes a false prediction, the rest of the pipeline is affected. An end-to-end model which consumes the raw video and outputs the transcripts including the speaker identities directly can be a future direction to avoid this issue.

Finally, we have not proven that our method can generalise on videos other than TV shows. There are a plethora of public video datasets from different domains such as movies [Bain et al. 2019] or news clips [J. S. Chung et al. 2020c]. It is possible to adapt our method to these domains and make the character-aware audio-visual subtitling more practical. Especially in movies, I have already developed VoxMovies dataset which contains utterances from movie characters. We could finetune the speaker embedding extractors and audio-visual speaker detection models on this dataset to improve the performance of our method.

Towards internet scale datasets. Over the past few years, there has been a significant shift towards deep learning models trained on internet-scale datasets, resulting in high-quality representations. State-of-the-art models including CLIP [Radford et al. 2021] for vision-language understanding, Whisper [Radford et al. 2023] for speech recognition, and large language models [Touvron et al. 2023; Achiam et al. 2023; Anil et al. 2023] typically leverage internet-scale data. While we introduce three new datasets in this thesis, they are all less than 1,000 hours in duration, which limits the models' ability to learn robust representations. There is still a lack of established dataset curation methods for collecting and annotating data at such massive scales, an area that remains underexplored. The development of efficient and effective techniques for curating internet-scale training data would be beneficial for the research community, enabling scientists to push the boundaries of their models.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. (2023). "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774*.

Triantafyllos Afouras, Yuki M Asano, Francois Fagan, Andrea Vedaldi, and Florian Metze (2022). "Self-supervised object detection from audio-visual correspondence". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman (2019a). "Deep Audio-Visual Speech Recognition". In: *IEEE PAMI*.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman (2018a). "LRS3-TED: a large-scale dataset for visual speech recognition". In: *arXiv preprint arXiv:1809.00496*.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman (2018b). "The Conversation: Deep Audio-Visual Speech Enhancement". In: *INTERSPEECH*.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman (2019b). "My lips are concealed: Audio-visual speech enhancement through obstructions". In: *Interspeech*.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman (2020a). "Now you're speaking my language: Visual language identification". In: *INTERSPEECH*.

Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman (2020b). "Self-Supervised Learning of Audio-Visual Objects from Video". In: *Proc. ECCV*.

Wataru Akahori, Tatsunori Hirai, and Shigeo Morishima (2017). "Dynamic subtitle placement considering the region of interest and speaker location". In: *International Conference on Computer Vision Theory and Applications*. SciTePress.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. (2022). "Flamingo: a visual language model for few-shot learning". In: *Advances in neural information processing systems*.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. (2023). "Gemini: a family of highly capable multimodal models". In: *arXiv preprint arXiv:2312.11805*.

Relja Arandjelovic and Andrew Zisserman (2018). "Objects that sound". In: *Proceedings of the European conference on computer vision (ECCV)*.

Martin Arjovsky, Soumith Chintala, and Léon Bottou (2017). "Wasserstein gan". In: *arXiv preprint arXiv:1701.07875*.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid (2021). "ViViT: A Video Vision Transformer". In: *Proceedings of International Conference on Computer Vision (ICCV)*.

Yusuf Aytar, Carl Vondrick, and Antonio Torralba (2016). "Soundnet: Learning sound representations from unlabeled video". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton (2016). "Layer normalization". In: *arXiv preprint arXiv:1607.06450*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli (2020). "wav2vec 2.0: A framework for self-supervised learning of speech representations". In: *NeurIPS*.

Anurag Bagchi, Jazib Mahmood, Dolton Fernandes, and Ravi Kiran Sarvadevabhatla (2021). "Hear me out: Fusional approaches for audio augmented temporal action localization". In: *arXiv preprint arXiv:2106.14118*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou (2023). "Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities". In: *arXiv preprint arXiv:2308.12966*.

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman (2023). "WhisperX: Time-Accurate Speech Transcription of Long-Form Audio". In: *INTERSPEECH*.

Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman (2020). "Condensed Movies: Story Based Retrieval with Contextual Embeddings". In: *Proc. ACCV*.

Max Bain, Arsha Nagrani, Daniel Schofield, and Andrew Zisserman (2019). "Count, Crop and Recognise: Fine-Grained Recognition in the Wild". In: *Workshop on Computer Vision for Wildlife Conservation, ICCV*.

Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori (2018). "Object Level Visual Reasoning in Videos". In: *Proceedings of European Conference on Computer Vision (ECCV)*.

Sara Beery, Guanhang Wu, Vivek Rathod, Ronny Votel, and Jonathan Huang (2020). "Context r-cnn: Long term temporal context for per-camera object detection". In: *Proc. CVPR*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer (2015). "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin (2003). "A neural probabilistic language model". In: *Journal of Machine Learning Research (JMLR)*.

T. Berg, A. Berg, J. Edwards, M. Mair, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth (2004). "Names and Faces in the News". In: *Proc. CVPR*.

Gedas Bertasius and Lorenzo Torresani (2020). "Classifying, segmenting, and tracking object instances in video with mask propagation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani (2021). "Is Space-Time Attention All You Need for Video Understanding?" In: *Proceedings of International Conference on Machine Learning (ICML)*.

Gautam Bhattacharya, Jahangir Alam, and Patrick Kenny (2019a). "Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training". In: *Proc. ICASSP*.

Gautam Bhattacharya, Joao Monteiro, Jahangir Alam, and Patrick Kenny (2019b). "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification". In: *Proc. ICASSP*.

Swapnil Bhosale, Sauradip Nag, Diptesh Kanojia, Jiankang Deng, and Xiatian Zhu (2023). "DiffSED: Sound Event Detection with Denoising Diffusion". In: *arXiv preprint arXiv:2308.07293*.

Steven Bird (2006). "NLTK: the natural language toolkit". In: *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*.

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis (2017). *Soft-NMS – Improving Object Detection With One Line of Code*. arXiv: 1704.04503 [cs.CV].

Xavier Bost, Georges Linarès, and Serigne Gueye (2015). "Audiovisual speaker diarization of TV series". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Hervé Bredin (2023). "pyannote. audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe". In: *Proc. Interspeech*. ISCA.

Hervé Bredin and Antoine Laurent (2021). "End-to-end speaker segmentation for overlap-aware resegmentation". In: *Proc. Interspeech*.

Hervé Bredin, Antoine Laurent, Achintya Sarkar, Viet-Bac Le, Sophie Rosset, and Claude Barras (2014). "Person instance graphs for named speaker identification in tv broadcast". In: *Odyssey 2014*.

Andrew Brown, Ernesto Coto, and Andrew Zisserman (2021a). "Automated Video Labelling: Identifying Faces by Corroborative Evidence". In: *International Conference on Multimedia Information Processing and Retrieval*.

Andrew Brown, Vicky Kalogeiton, and Andrew Zisserman (2021b). "Face, Body, Voice: Video Person-Clustering with Multiple Modalities". In: *ICCV 2021 Workshop on AI for Creative Video Editing and Understanding*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Adrian Bulat, Juan-Manuel Perez-Rua, Swathikiran Sudhakaran, Brais Martínez, and Georgios Tzimiropoulos (2021). "Space-time Mixing Attention for Video Transformer". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Weicheng Cai, Danwei Cai, Wenbo Liu, Gang Li, and Ming Li (2017). "Countermeasures for Automatic Speaker Verification Replay Spoofing Attack: On Data Augmentation, Feature Representation, Classification and Fusion." In: *INTERSPEECH*.

Weicheng Cai, Jinkun Chen, and Ming Li (2018). "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system". In: *Speaker Odyssey*.

Alexandra Canavan, David Graff, and George Zipperlen (1997). "Callhome american english speech". In: *Linguistic Data Consortium*.

Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman (2018). "VGGFace2: A dataset for recognising faces across pose and age". In: *Proc. Int. Conf. Autom. Face and Gesture Recog.*

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko (2020). "End-to-End Object Detection with Transformers". In: *Proc. ECCV*.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. (2005). "The AMI meeting corpus: A pre-announcement". In: *International workshop on machine learning for multimodal interaction*. Springer.

Joao Carreira and Andrew Zisserman (2017). "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alejandro Cartas, Petia Radeva, and Mariella Dimiccoli (2021). "Modeling Long-Term Interactions to Enhance Action Recognition". In: *Proceedings of International Conference on Pattern Recognition (ICPR)*.

Brandon Castellano (2020). *PySceneDetect*. https://github.com/Breakthrough/PySceneDetect.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals (2016). "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Changan Chen, Kumar Ashutosh, Rohit Girdhar, David Harwath, and Kristen Grauman (2024). "Soundingactions: Learning how actions sound from narrated egocentric videos". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman (2021). "Localizing Visual Sounds the Hard Way". In: *Proc. CVPR*.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman (2020). "VGGSound: A large-scale audio-visual dataset". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Jun Chen, Deyao Zhu, Kilichbek Haydarov, Xiang Li, and Mohamed Elhoseiny (2023). "Video ChatCaptioner: Towards the Enriched Spatiotemporal Descriptions". In: *arXiv preprint arXiv:2304.04227*.

Lele Chen, Sudhanshu Srivastava, Zhiyao Duan, and Chenliang Xu (2017). "Deep cross-modal audio-visual generation". In: *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*.

Zhuxin Chen, Zhifeng Xie, Weibin Zhang, and Xiangmin Xu (2017). "ResNet and Model Fusion for Automatic Spoofing Detection." In: *INTERSPEECH*.

Gaofeng Cheng, Yifan Chen, Runyan Yang, Qingxuan Li, Zehui Yang, Lingxuan Ye, Pengyuan Zhang, Qingqing Zhang, Lei Xie, Yanmin Qian, et al. (2022). "The conversational short-phrase speaker diarization (cssd) task: Dataset, evaluation metric and baselines". In: *2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE.

Joon Son Chung, Jaesung Huh, and Seongkyu Mun (2020a). "Delving into VoxCeleb: environment invariant speaker recognition". In: *Speaker Odyssey Workshop*.

Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han (2020b). "In defence of metric learning for speaker recognition". In: *Proc. Interspeech*.

Joon Son Chung, Jaesung Huh, Arsha Nagrani, Triantafyllos Afouras, and Andrew Zisserman (2020c). "Spot the conversation: speaker diarisation in the wild". In: *INTERSPEECH*.

Joon Son Chung, Amir Jamaludin, and Andrew Zisserman (2017). "You said that?" In: *Proc. BMVC*.

Joon Son Chung, Bong-Jin Lee, and Icksang Han (2019). "Who said that?: Audio-visual speaker diarisation of real-world meetings". In: *Proc. Interspeech*.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman (2018). "VoxCeleb2: Deep Speaker Recognition". In: *INTERSPEECH*.

Joon Son Chung and Andrew Zisserman (2016). "Out of time: automated lip sync in the wild". In: *Workshop on Multi-view Lip-reading, ACCV*.

Joon Son Chung and Andrew Zisserman (2017). "Lip Reading in Profile". In: *Proc. BMVC*.

Soo-Whan Chung, Joon Son Chung, and Hong-Goo Kang (2019). "Perfect match: Improved cross-modal embeddings for audio-visual synchronisation". In: *Proc. ICASSP*. IEEE.

Christopher Cieri, David Miller, and Kevin Walker (2004). "Fisher English training speech parts 1 and 2". In: *Philadelphia: Linguistic Data Consortium*.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le (2020). "Randaugment: Practical automated data augmentation with a reduced search space". In:

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov (n.d.). "Transformer-xl: Attentive language models beyond a fixed-length context. arXiv 2019". In: *arXiv preprint arXiv:1901.02860* ().

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray (2018). "Scaling Egocentric Vision: The EPIC-KITCHENS Dataset". In: *European Conference on Computer Vision (ECCV)*.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. (2022). "Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100". In: *International Journal of Computer Vision*.

Dima Damen, Adriano Fragomeni, Jonathan Munro, Toby Perrett, Daniel Whettam, and Michael Wray (2021). *EPIC-KITCHENS 2021 Challenge Report*. https://epic-kitchens.github.io/Reports/EPIC-KITCHENS-Challenges-2021-Report.pdf.

Rohan Kumar Das, Jichen Yang, and Haizhou Li (2019). "Long Range Acoustic Features for Spoofed Speech Detection". In: *INTERSPEECH*.

Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham J Mysore, Fredo Durand, and William T Freeman (2014). "The visual microphone: Passive recovery of sound from video". In.

William HE Day and Herbert Edelsbrunner (1984). "Efficient algorithms for agglomerative hierarchical clustering methods". In: *Journal of classification*.

Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi (2020). "Real Time Speech Enhancement in the Waveform Domain". In: *Interspeech*.

Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet (2010). "Front-end factor analysis for speaker verification". In: *IEEE Transactions on Audio, Speech, and Language Processing*.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). "ImageNet: A Large-Scale Hierarchical Image Database". In: *Proc. CVPR*.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck (2020). "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification". In: *Proc. Interspeech*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "Bert: Pre-training of deep bidirectional transformers for language understanding". In:

Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL).

Mireia Diez, Lukáš Burget, Federico Landini, and Jan Černockỳ (2019). "Analysis of speaker diarization based on Bayesian HMM with eigenvoice priors". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing.*

Yifan Ding, Yong Xu, Shi-Xiong Zhang, Yahuan Cong, and Liqiang Wang (2020). "Self-supervised learning for audio-visual speaker diarization". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *Proceedings of International Conference on Learning Representations (ICLR).*

Yerai Doval and Carlos Gómez-Rodríguez (2019). "Comparing neural-and N-gram-based language models for word segmentation". In: *Journal of the Association for Information Science and Technology.*

Yuexi Du, Ziyang Chen, Justin Salamon, Bryan Russell, and Andrew Owens (2023). "Conditional generation of audio from video via foley analogies". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

A. Dutta, A. Gupta, and A. Zisserman (2016). *VGG Image Annotator (VIA).* http://www.robots.ox.ac.uk/ vgg/software/via/.

Abhishek Dutta and Andrew Zisserman (2019). "The VIA Annotation Software for Images, Audio and Video". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19.

Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein (2018). "Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation". In: *ACM Transactions on Graphics (TOG).*

Mark Everingham, Josef Sivic, and Andrew Zisserman (2006). ""Hello! My name is... Buffy" – Automatic Naming of Characters in TV Video". In: *Proc. BMVC.*

Mark Everingham, Josef Sivic, and Andrew Zisserman (2009). "Taking the Bite out of Automatic Naming of Characters in TV Video". In: *Image and Vision Computing.*

Yue Fan, JW Kang, LT Li, KC Li, HL Chen, ST Cheng, PY Zhang, ZY Zhou, YQ Cai, and Dong Wang (2020). "CN-CELEB: a challenging Chinese speaker recognition dataset". In: *Proc. ICASSP.*

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He (2019). "Slowfast networks for video recognition". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Fan Feng, Yue Ming, Nannan Hu, Hui Yu, and Yuanan Liu (2023). "CSS-Net: A Consistent Segment Selection Network for Audio-visual Event Localization". In: *IEEE Transactions on Multimedia*.

Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra (2022). "FSD50K: an open dataset of human-labeled sound events". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe (2019). "End-to-end neural speaker diarization with permutation-free objectives". In: *Proc. Interspeech*.

Antonino Furnari and Giovanni Maria Farinella (2019). "What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention". In: *Proceedings of the IEEE/CVF International conference on computer vision*.

Valentin Gabeur, Paul Hongsuck Seo, Arsha Nagrani, Chen Sun, Karteek Alahari, and Cordelia Schmid (2022). "AVATAR: Unconstrained Audiovisual Speech Recognition". In: *INTERSPEECH*.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid (2020). "Multi-modal transformer for video retrieval". In: *Proceedings of European Conference on Computer Vision (ECCV)*.

Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba (2020). "Foley music: Learning to generate music from videos". In: *Proc. ECCV*.

Ruohan Gao, Rogerio Feris, and Kristen Grauman (2018). "Learning to separate object sounds by watching unlabeled video". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ruohan Gao and Kristen Grauman (2019). "Co-separating sounds of visual objects". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Ruohan Gao and Kristen Grauman (2021). "Visualvoice: Audio-visual speech separation with cross-modal consistency". In: *Proc. CVPR*. IEEE.

Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani (2020). "Listen to look: Action recognition by previewing audio". In: *Proc. CVPR*.

Daniel Garcia-Romero and Alan McCree (2014). "Supervised domain adaptation for i-vector based speaker recognition". In: *Proc. ICASSP*.

Daniel Garcia-Romero, Alan McCree, Stephen Shum, Niko Brummer, and Carlos Vaquero (2014a). "Unsupervised domain adaptation for i-vector speaker

recognition". In: *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*.

Daniel Garcia-Romero, Alan McCree, David Snyder, and Gregory Sell (2020). "Jhu-HLTCOE System for the Voxsrc Speaker Recognition Challenge". In: *Proc. ICASSP*.

Daniel Garcia-Romero, David Snyder, Gregory Sell, Daniel Povey, and Alan McCree (2017). "Speaker diarization using deep neural network embeddings". In: *Proc. ICASSP*. IEEE.

Daniel Garcia-Romero, Xiaohui Zhang, Alan McCree, and Daniel Povey (2014b). "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks". In: *SLT*.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter (2017). "Audio set: An ontology and human-labeled dataset for audio events". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Jon Gillick, Wesley Deng, Kimiko Ryokai, and David Bamman (2021). "Robust Laughter Detection in Noisy Environments." In: *Proc. Interspeech*.

Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman (2019). "Video Action Transformer Network". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

Rohit Girdhar and Kristen Grauman (2021). "Anticipative video transformer". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra (2022). "Omnivore: A Single Model for Many Visual Modalities". In: *CVPR*.

Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass (2022). "SSAST: Self-Supervised Audio Spectrogram Transformer". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yuan Gong, Alexander H Liu, Hongyin Luo, Leonid Karlinsky, and James Glass (2023). "Joint Audio and Speech Understanding". In: *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

Yuan Gong, Hongyin Luo, Alexander H Liu, Leonid Karlinsky, and James Glass (2024). "Listen, Think, and Understand". In: *Proc. ICLR*.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and

Roland Memisevic (2017). *The "something something" video database for learning and evaluating visual common sense*. arXiv: 1706.04261 [cs.CV].

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. (2022). "Ego4d: Around the world in 3,000 hours of egocentric video". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber (2006). "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks". In: *Proceedings of International Conference on Machine Learning (ICML)*.

Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik (2018). *AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions*. arXiv: 1705.08421 [cs.CV].

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. (2020). "Conformer: Convolution-augmented transformer for speech recognition". In: *arXiv preprint arXiv:2005.08100*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio (2017). "On integrating a language model into neural machine translation". In: *Computer Speech & Language*.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman (2023). "AutoAD: Movie Description in Context". In: *Proc. CVPR*.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman (2024). "AutoAD III: The Prequel – Back to the Pixels". In: *Proc. CVPR*.

Awni Y. Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng (2014). "Deep Speech: Scaling up end-to-end speech recognition". In: *CoRR*.

David Harwath, Antonio Torralba, and James Glass (2016). "Unsupervised Learning of Spoken Language with Visual Context". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Monica-Laura Haurilet, Makarand Tapaswi, Ziad Al-Halah, and Rainer Stiefelhagen (2016). "Naming TV characters by watching and analyzing dialogs". In: *Proc. WACV*. IEEE.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mao-Kui He, Jun Du, and Chin-Hui Lee (2022). "End-to-end audio-visual neural speaker diarization". In: *Proc. Interspeech*.

Yayun He, Zuheng Kang, Jianzong Wang, Junqing Peng, and Jing Xiao (2023). "Voiceextender: Short-Utterance Text-Independent Speaker Verification With Guided Diffusion Model". In: *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE.

Dan Hendrycks and Kevin Gimpel (2016). "Gaussian error linear units (gelus)". In: *arXiv preprint arXiv:1606.08415*.

Hee Soo Heo, Bong-Jin Lee, Jaesung Huh, and Joon Son Chung (2020). "Clova Baseline System for the VoxCeleb Speaker Recognition Challenge 2020". In: *arXiv preprint arXiv:2009.14153*.

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. (2017). "CNN architectures for large-scale audio classification". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Roei Herzig, Ofir Abramovich, Elad Ben-Avraham, Assaf Arbelle, Leonid Karlinsky, Ariel Shamir, Trevor Darrell, and Amir Globerson (2022). "PromptonomyViT: Multi-Task Prompt Learning Improves Video Transformers using Synthetic Scene Data". In: *arXiv preprint arXiv:2212.04821*.

ALLEN Hirson and MARTIN Duckworth (1993). "Glottal fry and voice disguise: a case study in forensic phonetics". In: *Journal of biomedical engineering*.

Daniel Tang Kuok Ho and Rangis Intai (2017). "Effectiveness of audio-visual aids in teaching lower secondary science in a rural secondary school". In: *Asia Pacific Journal of Educators and Education*.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell (2018). "Cycada: Cycle-consistent adversarial domain adaptation". In: *Proc. ICML*.

Qian-Bei Hong, Chung-Hsien Wu, and Hsin-Min Wang (2023). "Decomposition and reorganization of phonetic information for speaker embedding learning". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Shota Horiguchi, Yusuke Fujita, Shinji Watanabe, Yawen Xue, and Kenji Nagamatsu (2020). "End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors". In: *Proc. Interspeech*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed (2021). "Hubert: Self-supervised speech representation learning by masked prediction of hidden units". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Xinghang Hu, Yanli Ji, and Gedamu Alemu Kumie (2022). "Multi-level multi-modal feature fusion for action recognition in videos". In: *Proceedings of the 3rd International Workshop on Human-Centric Multimedia Analysis*.

Xixi Hu, Ziyang Chen, and Andrew Owens (2022). "Mix and localize: Localizing sound sources in mixtures". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yongtao Hu, Jimmy SJ Ren, Jingwen Dai, Chang Yuan, Li Xu, and Wenping Wang (2015). "Deep multimodal speaker naming". In: *Proceedings of the 23rd ACM international conference on Multimedia*.

Yifei Huang, Minjie Cai, Zhenqiang Li, Feng Lu, and Yoichi Sato (2020). "Mutual Context Network for Jointly Estimating Egocentric Gaze and Action". In: *IEEE Transactions on Image Processing (TIP)*.

Jaesung Huh, Jacob Chalk, Evangelos Kazakos, Dima Damen, and Andrew Zisserman (2023). "EPIC-SOUNDS: A Large-Scale Dataset of Actions that Sound". In: *IEEE International Conference on Acoustics, Speech, & Signal Processing (ICASSP)*.

Jaesung Huh, Joon Son Chung, Arsha Nagrani, Andrew Brown, Jee-weon Jung, Daniel Garcia-Romero, and Andrew Zisserman (2024). "The VoxCeleb Speaker Recognition Challenge: A Retrospective". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Vladimir Iashin and Esa Rahtu (2020). "Multi-modal dense video captioning". In: *Proceedings of Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

International Movie Database (n.d.). https://www.imdb.com.

Sergey Ioffe (2006). "Probabilistic linear discriminant analysis". In: *Proc. ECCV*. Springer.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira (2021). "Perceiver: General Perception with Iterative Attention". In: *CoRR*.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. (2003). "The ICSI meeting corpus". In: *Proc. ICASSP*. IEEE.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim (2022). "Visual Prompt Tuning". In: *European Conference on Computer Vision (ECCV)*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. (2023). "Mistral 7B". In: *arXiv preprint arXiv:2310.06825*.

Alan B Johnston and Daniel C Burnett (2012). *WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web*. Digital Codex LLC.

Chaeyoung Jung, Suyeon Lee, Kihyun Nam, Kyeongha Rho, You Jin Kim, Youngjoon Jang, and Joon Son Chung (2024). "TalkNCE: Improving Active Speaker Detection with Talk-Aware Contrastive Learning". In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Jee-weon Jung, Hee-Soo Heo, Bong-Jin Lee, Jaesung Huh, Andrew Brown, Youngki Kwon, Shinji Watanabe, and Joon Son Chung (2023). "In search of strong embedding extractors for speaker diarisation". In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Vicky Kalogeiton and Andrew Zisserman (2020). "Constrained Video Face Clustering using 1NN Relations". In: *Proc. BMVC*.

Jiawen Kang, Ruiqi Liu, Lantian Li, Yunqi Cai, Dong Wang, and Thomas Fang Zheng (2020). "Domain-Invariant Speaker Vector Projection by Model-Agnostic Meta-Learning". In: *arXiv preprint arXiv:2005.11900*.

NK Kaphungkui and Aditya Bihar Kandali (2019). "Text dependent speaker recognition with back propagation neural network". In: *International Journal of Engineering and Advanced Technology (IJEAT)*.

Georgios Kapidis, Ronald Poppe, Elsbeth van Dam, Lucas Noldus, and Remco Veltkamp (2019). "Multitask Learning to Improve Egocentric Action Recognition". In: *Proceedings of International Conference on Computer Vision Workshops (ICCVW)*.

Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev,

Mustafa Suleyman, and Andrew Zisserman (2017). "The Kinetics Human Action Video Dataset". In: *CoRR*.

Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, and Dima Damen (2021a). "With a Little Help from my Temporal Context: Multimodal Egocentric Action Recognition". In: *British Machine Vision Conference (BMVC)*.

Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen (2019). "EPIC-Fusion: Audio-Visual Temporal Binding for Egocentric Action Recognition". In: *Proc. ICCV*.

Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen (2021b). "Slow-Fast Auditory Streams For Audio Recognition". In: *International Conference on Acoustics, Speech, and Signal Processing*.

Patrick Kenny, Themos Stafylakis, Pierre Ouellet, Md Jahangir Alam, and Pierre Dumouchel (2013). "PLDA for speaker verification with utterances of arbitrary duration". In: *Proc. ICASSP*. IEEE.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush (2016). "Character-aware neural language models". In: *Proceedings of AAAI Conference on Artificial Intelligence*.

You Jin Kim, Hee-Soo Heo, Soyeon Choe, Soo-Whan Chung, Yoohwan Kwon, Bong-Jin Lee, Youngki Kwon, and Joon Son Chung (2021). "Look who's talking: Active speaker detection in the wild". In: *arXiv preprint arXiv:2108.07640*.

Keisuke Kinoshita, Marc Delcroix, and Naohiro Tawara (2021). "Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Nithin Rao Koluguri, Taejin Park, and Boris Ginsburg (2022). "Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context". In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong (2021). "Movinets: Mobile video networks for efficient video recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Bruno Korbar, Jaesung Huh, and Andrew Zisserman (2024). "Look, Listen and Recognise: character-aware audio-visual subtitling". In: *International Conference on Acoustics, Speech, and Signal Processing*.

Bruno Korbar, Du Tran, and Lorenzo Torresani (2018). “Cooperative learning of audio and video models from self-supervised synchronization”. In: *Advances in Neural Information Processing Systems*.

Bruno Korbar and Andrew Zisserman (2022). “Personalised CLIP or: how to find your vacation videos”. In: *Proc. BMVC*.

Wessel Kraaij, Thomas Hain, Mike Lincoln, and Wilfried Post (2005). “The AMI meeting corpus”. In: *Proc. International Conference on Methods and Techniques in Behavioral Research*.

Patricia K Kuhl and Andrew N Meltzoff (1982). “The bimodal perception of speech in infancy”. In: *Science*.

Doyeop Kwak, Jaemin Jung, Kihyun Nam, Youngjoon Jang, Jee-won Jung, Shinji Watanebe, and Joon Son Chung (2024). “VoxMM: Rich transcription of conversations in the wild”. In: *International Conference on Acoustics, Speech, and Signal Processing*.

Youngki Kwon, Hee Soo Heo, Jaesung Huh, Bong-Jin Lee, and Joon Son Chung (2021). “Look who’s not talking”. In: *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE.

Sangho Lee, Youngjae Yu, Gunhee Kim, Thomas Breuel, Jan Kautz, and Yale Song (2021). “Parameter Efficient Multimodal Transformers for Video Representation Learning”. In: *Proceedings of International Conference on Learning Representations (ICLR)*.

Paul Lerner, Juliette Bergoënd, Camille Guinaudeau, Hervé Bredin, Benjamin Maurice, Sharleyne Lefevre, Martin Bouteiller, Aman Berhe, Léo Galmant, Ruiqing Yin, et al. (2022). “Bazinga! A Dataset for Multi-Party Dialogues Structuring”. In: *LREC*.

Guang Li, Linchao Zhu, Ping Liu, and Yi Yang (2019). “Entangled transformer for image captioning”. In: *Proceedings of International Conference on Computer Vision (ICCV)*.

Jiaman Li, Yihang Yin, Hang Chu, Yi Zhou, Tingwu Wang, Sanja Fidler, and Hao Li (2020). “Learning to Generate Diverse Dance Motions with Transformer”. In: *CoRR*.

KP Li and EH Wrench Jr (1982). “Text-independent speaker recognition with short utterances”. In: *The Journal of the Acoustical Society of America*.

Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa (2021). “AI Choreographer: Music Conditioned 3D Dance Generation With AIST++”. In: *Proceedings of International Conference on Computer Vision (ICCV)*.

Yanghao Li, Tushar Nagarajan, Bo Xiong, and Kristen Grauman (2021). "Ego-Exo: Transferring Visual Representations From Third-Person to First-Person Videos". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yin Li, Miao Liu, and James M. Rehg (2018). "In the Eye of Beholder: Joint Learning of Gaze and Actions in First Person Video". In: *Proceedings of European Conference on Computer Vision (ECCV)*.

Yingwei Li, Yi Li, and Nuno Vasconcelos (2018). "RESOUND: Towards Action Recognition without Representation Bias". In: *Proceedings of European Conference on Computer Vision (ECCV)*.

Jiachen Lian, Alexei Baevski, Wei-Ning Hsu, and Michael Auli (2023). "Av-data2vec: Self-supervised learning of audio-visual speech representations with contextualized target representations". In: *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE.

Chien-Feng Liao, Yu Tsao, Hung-Yi Lee, and Hsin-Min Wang (2018). "Noise adaptive speech enhancement using domain adversarial training". In: *arXiv preprint arXiv:1807.07501*.

Ji Lin, Chuang Gan, and Song Han (2019). "TSM: Temporal Shift Module for Efficient Video Understanding". In: *Proceedings of International Conference on Computer Vision (ICCV)*.

Mengxi Lin, Nakamasa Inoue, and Koichi Shinoda (2017). "CTC network with statistical language modeling for action sequence recognition in videos". In: *Proceedings of the on Thematic Workshops of ACM Multimedia*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár (2020). "Focal Loss for Dense Object Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yan-Bo Lin, Jie Lei, Mohit Bansal, and Gedas Bertasius (2022). "Eclipse: Efficient long-range video retrieval using sight and sound". In: *European Conference on Computer Vision*. Springer.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee (2023). "Visual Instruction Tuning". In: *NeurIPS*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang (2017). "Adversarial multi-task learning for text classification". In: *arXiv preprint arXiv:1704.05742*.

Tao Liu and Kai Yu (2022). "BER: Balanced Error Rate For Speaker Diarization". In: *arXiv preprint arXiv:2211.04304*.

Xuechen Liu, Md Sahidullah, and Tomi Kinnunen (2021). "Optimized power normalized cepstral coefficients towards robust deep speaker verification". In: *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE.

Xuechen Liu, Md Sahidullah, and Tomi Kinnunen (2022). "Learnable nonlinear compression for robust speaker verification". In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai (2021). "Efficient Training of Visual Transformers with Small Datasets". In: *Advances in Neural Information Processing Systems*.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu (2022). "Video swin transformer". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf (2020). "Object-Centric Learning with Slot Attention". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Ilya Loshchilov and Frank Hutter (2017). "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101*.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan (2024). *DeepSeek-VL: Towards Real-World Vision-Language Understanding*. arXiv: 2403.05525 [cs.AI].

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee (2019). "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Minlong Lu, Danping Liao, and Ze-Nian Li (2019). "Learning Spatiotemporal Attention for Egocentric Action Recognition". In: *Proceedings of International Conference on Computer Vision Workshops (ICCVW)*.

Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Da Li, Pengcheng Lu, Tao Wang, Linmei Hu, Minghui Qiu, and Zhongyu Wei (2023). "Valley: Video assistant with large language model enhanced ability". In: *arXiv preprint arXiv:2306.07207*.

Chau Luu, Peter Bell, and Steve Renals (2020). "Channel adversarial training for speaker verification and diarization". In: *Proc. ICASSP*.

Pingchuan Ma, Alexandros Haliassos, Adriana Fernandez-Lopez, Honglie Chen, Stavros Petridis, and Maja Pantic (2023). "Auto-avsr: Audio-visual speech

recognition with automatic labels". In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan (2024). "Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*.

J Naga Madhuri (2013). "Use of Audio Visual Aids in Teaching and Speaking". In: *Research Journal of English Language and Literature*.

Sagnik Majumder and Kristen Grauman (2022). "Active audio-visual separation of dynamic sound sources". In: *Proc. ECCV*. Springer.

Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom (2021). "Automatic speech recognition: a survey". In: *Multimedia Tools and Applications*.

Pavel Matejka, Ondrej Novotnỳ, Oldrich Plchot, and Lukas Burget (2017). "Analysis of Score Normalization in Multilingual Speaker Recognition." In.

Nalliveettil George Mathew and Ali Odeh Hammoud Alidmat (2013). "A study on the usefulness of audio-visual aids in EFL classroom: Implications for effective instruction." In: *International Journal of Higher Education*.

Harry McGurk and John MacDonald (1976). "Hearing lips and seeing voices". In: *Nature*.

Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson (2016). "The speakers in the wild (SITW) speaker recognition database." In: *Interspeech*.

Mitchell McLaren, Aaron Lawson, Luciana Ferrer, Diego Castan, and Martin Graciarena (2015). "The speakers in the wild speaker recognition challenge plan". In: *Interspeech 2016 Special Session, San Francisco*.

Zhong Meng, Jinyu Li, Yifan Gong, et al. (2018). "Adversarial feature-mapping for speech enhancement". In: *INTERSPEECH*.

Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen (2016). "TUT database for acoustic scene classification and sound event detection". In: *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur (2010). "Recurrent neural network based language model". In: *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

Kyle Min and Jason J. Corso (2021). "Integrating Human Gaze Into Attention for Egocentric Activity Recognition". In: *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*.

Bogdan Mocanu, Ruxandra Tapu, and Titus Zaharia (2019). "Enhancing the accessibility of hearing impaired to video content through fully automatic dynamic captioning". In: *2019 E-Health and Bioengineering Conference (EHB)*.

Marc Moreaux, Michael Garcia Ortiz, Isabelle Ferrané, and Frédéric Lerasle (2019). "Benchmark for kitchen20, a daily life dataset for audio-based human action recognition". In: *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE.

Pedro Morgado, Nuno Vasconcelos, and Ishan Misra (2021). "Audio-Visual Instance Discrimination with Cross-Modal Agreement". In: *Proc. CVPR*.

Liad Mudrik, Nathan Faivre, and Christof Koch (2014). "Information integration without awareness". In: *Trends in cognitive sciences*.

Jonathan Munro and Dima Damen (2020). "Multi-Modal Domain Adaptation for Fine-Grained Action Recognition". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

Arsha Nagrani, Joon Son Chung, Samuel Albanie, and Andrew Zisserman (2020a). "Disentangled Speech Embeddings using Cross-Modal Self-Supervision". In: *International Conference on Acoustics, Speech, and Signal Processing*.

Arsha Nagrani, Joon Son Chung, Jaesung Huh, Andrew Brown, Ernesto Coto, Weidi Xie, Mitchell McLaren, Douglas A Reynolds, and Andrew Zisserman (2020b). "Voxsrc 2020: The second voxceleb speaker recognition challenge". In: *arXiv preprint arXiv:2012.06867*.

Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman (2019). "Voxceleb: Large-scale speaker verification in the wild". In: *Computer Speech and Language*.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman (2017). "VoxCeleb: a large-scale speaker identification dataset". In: *INTERSPEECH*.

Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman (2020c). "Speech2action: Cross-modal supervision for action recognition". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun (2021). "Attention bottlenecks for multimodal fusion". In: *Advances in neural information processing systems*.

Arsha Nagrani and Andrew Zisserman (2017). "From Benedict Cumberbatch to Sherlock Holmes: Character Identification in TV series without a Script". In: *Proc. BMVC*.

Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann (2021). "Video transformer network". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Netflix player control tests (n.d.). https://about.netflix.com/en/news/player-control-tests, 2023.

Yan Bin Ng and Basura Fernando (2019). "Human Action Sequence Classification". In: *CoRR*.

NIST 2018 Speaker Recognition Evaluation Plan (2018). https://www.nist.gov/system/files/documents/2018/08/17/sre18_eval_plan_2018-05-31_v6.pdf, See Section 3.1.

The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan (2009). https://web.archive.org/web/20100606092041if_/http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf, See Section 6.

IA Ojelade, BG Aregbesola, Adams Ekele, and Tope Gloria Olatunde-Aiyedun (2020). "Effects of audio-visual instructional materials on teaching science concepts in secondary schools in Bwari Area Council Abuja, Nigeria". In: *Ojelade, IA, Aregbesola, BG, Ekele, A., & Aiyedun, TG (September 2020). Effects of Audio-Visual Instructional Materials on Teaching Science Concepts in Secondary Schools in Bwari Area Council Abuja, Nigeria. The Environmental Studies Journal (TESJ)*.

Andrew Owens and Alexei A Efros (2018). "Audio-visual scene analysis with self-supervised multisensory features". In: *Proceedings of the European conference on computer vision (ECCV)*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur (2015). "Librispeech: An ASR corpus based on public domain audio books". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le (2019). "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition". In: *Proceedings of Interspeech*.

Robin Y Park, Rhydian Windsor, Amir Jamaludin, and Andrew Zisserman (2024). "Automated Spinal MRI Labelling from Reports Using a Large Language Model".

In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer.

Tae Jin Park, Naoyuki Kanda, Dimitrios Dimitriadis, Kyu J Han, Shinji Watanabe, and Shrikanth Narayanan (2022). "A review of speaker diarization: Recent advances with deep learning". In: *Computer Speech & Language*.

Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman (2015). "Deep Face Recognition". In: *Proc. BMVC*.

Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Mateusz Malinowski, Yi Yang, Carl Doersch, et al. (2024). "Perception test: A diagnostic benchmark for multimodal video models". In: *Advances in Neural Information Processing Systems*.

Mandela Patrick, Dylan Campbell, Yuki M. Asano, Ishan Misra Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F. Henriques (2021). "Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). "Deep contextualized word representations". In: *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Karol J. Piczak (Oct. 13, 2015). "ESC: Dataset for Environmental Sound Classification". In: *Proceedings of the 23rd Annual ACM Conference on Multimedia*. Brisbane, Australia: ACM Press. URL: http://dl.acm.org/citation.cfm?doid=2733373.2806390.

AJ Piergiovanni, Isaac Noble, Dahun Kim, Michael S Ryoo, Victor Gomes, and Anelia Angelova (2024). "Mirasol3B: A Multimodal Autoregressive model for time-aligned and contextual modalities". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Chiara Plizzari, Mirco Planamente, Gabriele Goletto, Marco Cannici, Emanuele Gusso, Matteo Matteucci, and Barbara Caputo (2022). "E2 (go) motion: Motion augmented event stream for egocentric action recognition". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Arnab Poddar, Md Sahidullah, and Goutam Saha (2018). "Speaker verification with short utterances: a review of challenges, trends and opportunities". In: *IET Biometrics*.

Johann Poignant, Hervé Bredin, and Claude Barras (2017). "Multimodal person discovery in broadcast tv: lessons learned from mediaeval 2015". In: *Multimedia Tools and Applications*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. (2011). "The Kaldi speech recognition toolkit". In: *IEEE workshop on automatic speech recognition and understanding*.

Gorjan Radevski, Dusan Grujicic, Matthew Blaschko, Marie-Francine Moens, and Tinne Tuytelaars (2023). "Multimodal distillation for egocentric action recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. (2021). "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever (2023). "Robust speech recognition via large-scale weak supervision". In: *Proc. ICML*.

Akam Rahimi, Triantafyllos Afouras, and Andrew Zisserman (2022). "Reading to Listen at the Cocktail Party: Multi-Modal Speech Separation". In: *Proc. CVPR*.

Vignesh Ramanathan, Armand Joulin, Percy Liang, and Li Fei-Fei (2014). "Linking people in videos with "their" names using coreference resolution". In: *Proc. ECCV*. Springer.

Merey Ramazanova, Victor Escorcia, Fabian Caba, Chen Zhao, and Bernard Ghanem (2023). "Owl (observe, watch, listen): Audiovisual temporal context for localizing actions in egocentric videos". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Alan R Reich and James E Duke (1979). "Effects of selected vocal disguises upon speaker identification by listening". In: *The Journal of the ASA*.

Alexander Richard and Juergen Gall (2016). "Temporal action detection using a statistical language model". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

Johan Rohdin, Themos Stafylakis, Anna Silnova, Hossein Zeinali, Lukáš Burget, and Oldřich Plchot (2019). "Speaker verification using end-to-end adversarial language adaptation". In: *Proc. ICASSP*.

Lawrence D Rosenblum, Mark A Schmuckler, and Jennifer A Johnson (1997). "The McGurk effect in infants". In: *Perception & psychophysics*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. (2015). "Imagenet large scale visual recognition challenge". In: *International journal of computer vision*.

Neville Ryant, Kenneth Church, Christopher Cieri, Alejandrina Cristia, Jun Du, Sriram Ganapathy, and Mark Liberman (2019). "The second DIHARD diarization challenge: Dataset, task, and baselines". In: *Interspeech*.

Justin Salamon, Duncan MacConnell, Mark Cartwright, Peter Li, and Juan Pablo Bello (2017). "Scaper: A library for soundscape synthesis and augmentation". In: *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE.

F. Schroff, D. Kalenichenko, and J. Philbin (2015). "FaceNet: A unified embedding for face recognition and clustering". In: *Proc. CVPR*.

Scrubs fandom (2024). https://scrubs.fandom.com/wiki/Category:Transcripts.

Seinfeld scripts dot com (2024). https://www.seinfeldscripts.com/seinfeld-scripts.html.

Gregory Sell and Daniel Garcia-Romero (2014). "Speaker diarization with PLDA i-vector scoring and unsupervised calibration". In: *IEEE Spoken Language Technology Workshop*. IEEE.

Gregory Sell, David Snyder, Alan McCree, Daniel Garcia-Romero, Jesús Villalba, Matthew Maciejewski, Vimal Manohar, Najim Dehak, Daniel Povey, Shinji Watanabe, et al. (2018). "Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge." In: *Interspeech*.

Fadime Sener, Dipika Singhania, and Angela Yao (2020). "Temporal Aggregate Representations for Long-Range Video Understanding". In: *Proceedings of European Conference on Computer Vision (ECCV)*.

Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag (2021). "Better aggregation in test-time augmentation". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Rahul Sharma and Shrikanth Narayanan (2022a). "Audio visual character profiles for detecting background characters in entertainment media". In: *arXiv preprint arXiv:2203.11368*.

Rahul Sharma and Shrikanth Narayanan (2022b). "Using active speaker faces for diarization in tv shows". In: *arXiv preprint arXiv:2203.15961*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani (2018). "Self-Attention with Relative Position Representations". In: *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed (2022a). "Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction". In: *arXiv preprint arXiv:2201.02184*.

Bowen Shi, Wei-Ning Hsu, and Abdelrahman Mohamed (2022b). "Robust self-supervised audio-visual speech recognition". In: *arXiv preprint arXiv:2201.01763*.

Joonbo Shin, Yoonhyung Lee, and Kyomin Jung (2019). "Effective sentence scoring method using bert for speech recognition". In: *Proceedings of Asian Conference on Machine Learning (ACML)*.

Karen Simonyan and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *ICLR*.

Simple Diarization repository (2024). https://github.com/JaesungHuh/SimpleDiarization.

Linda Smith and Michael Gasser (2005). "The development of embodied cognition: Six lessons from babies". In: *Artificial life*.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur (2018). "X-vectors: Robust dnn embeddings for speaker recognition". In: *Proc. ICASSP*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Xun Guo, Tian Ye, Yan Lu, Jenq-Neng Hwang, et al. (2024). "MovieChat: From Dense Token to Sparse Memory for Long Video Understanding". In: *Proc. CVPR*.

Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao (2015). "Sun rgb-d: A rgb-d scene understanding benchmark suite". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Charles Spence (2007). "Audiovisual multisensory integration". In: *Acoustical science and technology*.

Barry E Stein and Terrence R Stanford (2008). "Multisensory integration: current issues from the perspective of the single neuron". In: *Nature reviews neuroscience*.

Jonathan C Stroud, Zhichao Lu, Chen Sun, Jia Deng, Rahul Sukthankar, Cordelia Schmid, and David A Ross (2020). "Learning video representations from textual web supervision". In: *arXiv preprint arXiv:2007.14937*.

Kun Su, Xiulong Liu, and Eli Shlizerman (2021). "How does it sound? generation of rhythmic soundtracks for human movement videos". In: *Conf. Neural Inf. Process. Syst.*

TR Suchitha and AT Bindu (2015). "Feature Extraction using MFCC and Classification using GMM". In: *International Journal for Scientific Research & Development (IJSRD).*

Swathikiran Sudhakaran, Adrian Bulat, Juan-Manuel Perez-Rua, Alex Falcon, Sergio Escalera, Oswald Lanz, Brais Martinez, and Georgios Tzimiropoulos (2021). "SAIC_Cambridge-HuPBA-FBK Submission to the EPIC-Kitchens-100 Action Recognition Challenge 2021". In: *arXiv preprint arXiv:2110.02902.*

Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz (2019). "LSTA: Long Short-Term Attention for Egocentric Action Recognition". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR).*

Swathikiran Sudhakaran and Oswald Lanz (2018). "Attention is All We Need: Nailing Down Object-centric Attention for Egocentric Activity Recognition". In: *Proceedings of British Machine Vision Conference (BMVC).*

Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid (2019a). "Learning Video Representations using Contrastive Bidirectional Transformer". In: *CoRR.*

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid (2019b). "Videobert: A joint model for video and language representation learning". In: *Proceedings of International Conference on Computer Vision (ICCV).*

Lei Sun, Jun Du, Chao Jiang, Xueyang Zhang, Shan He, Bing Yin, and Chin-Hui Lee (2018). "Speaker Diarization with Enhancing Speech for the First DIHARD Challenge." In: *Interspeech.*

Agnieszka Szarkowska (2020). "Subtitling for the Deaf and the Hard of Hearing". In: *The Palgrave Handbook of Audiovisual Translation and Media Accessibility.*

Tsung-Ming Tai, Oswald Lanz, Giuseppe Fiameni, Yi-Kwan Wong, Sze-Sen Poon, Cheng-Kuang Lee, Ka-Chun Cheung, and Simon See (2022). "NVIDIA-UNIBZ Submission for EPIC-KITCHENS-100 Action Anticipation Challenge 2022". In: *arXiv preprint arXiv:2206.10869.*

Taoran Tang, Jia Jia, and Hanyang Mao (2018). "Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis". In: *Proceedings of the 26th ACM international conference on Multimedia.*

Silero Team (2021). *Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier.* https://github.com/snakers4/silero-vad.

Tuomas Teinonen, Richard N Aslin, Paavo Alku, and Gergely Csibra (2008). "Visual speech contributes to phonetic learning in 6-month-old infants". In: *Cognition*.

Virginia Teller (2000). "Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition". In: *Computational Linguistics*.

The Frasier Archives (2024). https://www.kacl780.net/.

Yapeng Tian, Dingzeyu Li, and Chenliang Xu (2020). "Unified Multisensory Perception: Weakly-Supervised Audio-Visual Video Parsing". In: *ECCV*.

Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu (2018). "Audio-visual event localization in unconstrained videos". In: *Proceedings of the European conference on computer vision (ECCV)*.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang (2022). "VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training". In: *Advances in Neural Information Processing Systems*.

Nikita Torgashov, Rostislav Makarov, Ivan Yakovlev, Pavel Malov, Andrei Balykin, and Anton Okhotnikov (2023). "The ID R&D VoxCeleb Speaker Recognition Challenge 2023 System Description". In: *arXiv preprint arXiv:2308.08294*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. (2023). "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971*.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri (2018). "A Closer Look at Spatiotemporal Convolutions for Action Recognition". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

SE Tranter and Douglas A Reynolds (2004). "Speaker diarisation for broadcast news". In: *Speaker Odyssey*.

Sue E Tranter (2006). "Who really spoke when? Finding speaker turns and identities in broadcast news audio". In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. IEEE.

Nicolas Turpault, Romain Serizel, Ankit Parag Shah, and Justin Salamon (2019). "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis". In: *Workshop on Detection and Classification of Acoustic Scenes and Events*.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell (2017). "Adversarial discriminative domain adaptation". In: *Proc. CVPR*.

Efthymios Tzinis, Scott Wisdom, Aren Jansen, Shawn Hershey, Tal Remez, Daniel PW Ellis, and John R Hershey (2021). "Into the wild with audioscope: Unsupervised audio-visual separation of on-screen sounds". In: *Proc. ICLR*.

Grant Van Horn, Rui Qian, Kimberly Wilber, Hartwig Adam, Oisin Mac Aodha, and Serge Belongie (2022). "Exploring Fine-grained Audiovisual Categorization with the SSW60 Dataset". In: *ECCV*.

Nancy Jean VanDerveer (1979). *Ecological acoustics: Human perception of environmental sounds*. Cornell University.

VGG List Annotator (LISA) (2022). https://www.robots.ox.ac.uk/ vgg/software/lisa/.

Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno (2018). "Generalized end-to-end loss for speaker verification". In: *Proc. ICASSP*.

Hanxin Wang, Shuchang Zhou, Qingbo Wu, Hongliang Li, Fanman Meng, Linfeng Xu, and Heqian Qiu (2023). "Confusion Mixup Regularized Multimodal Fusion Network for Continual Egocentric Activity Recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Hongji Wang, Chengdong Liang, Shuai Wang, Zhengyang Chen, Binbin Zhang, Xu Xiang, Yanlei Deng, and Yanmin Qian (2023). "Wespeaker: A research and production oriented speaker embedding learning toolkit". In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Junke Wang, Dongdong Chen, Chong Luo, Xiyang Dai, Lu Yuan, Zuxuan Wu, and Yu-Gang Jiang (2023). "Chatvideo: A tracklet-centric multimodal and versatile video understanding system". In: *arXiv preprint arXiv:2304.14407*.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool (2016). "Temporal segment networks: Towards good practices for deep action recognition". In: *Proceedings of European Conference on Computer Vision (ECCV)*.

Luyu Wang, Pauline Luc, Yan Wu, Adria Recasens, Lucas Smaira, Andrew Brock, Andrew Jaegle, Jean-Baptiste Alayrac, Sander Dieleman, Joao Carreira, et al. (2022). "Towards learning universal audio representations". In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Qing Wang, Wei Rao, Sining Sun, Leib Xie, Eng Siong Chng, and Haizhou Li (2018). "Unsupervised domain adaptation via domain adversarial training for speaker recognition". In: *Proc. ICASSP*.

Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopz Moreno (2018). "Speaker diarization with LSTM". In: *Proc. ICASSP*.

Weiyao Wang, Du Tran, and Matt Feiszli (2020). "What makes training multi-modal classification networks hard?" In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xiaolong Wang, Ali Farhadi, and Abhinav Gupta (2016). "Actions ~ Transformations". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao (2022). *InternVideo: General Video Foundation Models via Generative and Discriminative Learning*. arXiv: 2212.03191 [cs.CV].

TMF Wazeema and MIF Kareema (2017). "Implication of multimedia audio-visual aids in the English language classroom". In.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang (2024). "Longvlm: Efficient long video understanding via large language models". In: *arXiv preprint arXiv:2404.03384*.

Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen (2019). "Fine-Grained Action Retrieval Through Multiple Parts-of-Speech Embeddings". In: *Proceedings of International Conference on Computer Vision (ICCV)*.

Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick (2019). "Long-Term Feature Banks for Detailed Video Understanding". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer (2022). "Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer (2020). "Audiovisual SlowFast Networks for Video Recognition". In: *CoRR*.

Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman (2019). "Utterance-level Aggregation For Speaker Recognition In The Wild". In: *International Conference on Acoustics, Speech, and Signal Processing*.

Xuehan Xiong, Anurag Arnab, Arsha Nagrani, and Cordelia Schmid (2022). "M&M Mix: A Multimodal Multiview Transformer Ensemble". In: *arXiv preprint arXiv:2206.09852*.

Eric Zhongcong Xu, Zeyang Song, Satoshi Tsutsui, Chao Feng, Mang Ye, and Mike Zheng Shou (2022). "AVA-AVD: Audio-Visual Speaker Diarization in the Wild". In: MM '22. Lisboa, Portugal.

Mengmeng Xu, Chen Zhao, David S. Rojas, Ali Thabet, and Bernard Ghanem (2020). *G-TAD: Sub-Graph Localization for Temporal Action Detection*. arXiv: 1911.11462 [cs.CV].

Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid (2022). "Multiview transformers for video recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le (2019). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang (2019). "Cross-modal self-attention network for referring image segmentation". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sree Harsha Yella and Hervé Bourlard (2013). "Improved overlap speech diarization of meeting recordings using long-term conversational features". In: *Proc. ICASSP*. IEEE.

Samuel Yu, Peter Wu, Paul Pu Liang, Ruslan Salakhutdinov, and Louis-Philippe Morency (2022). "PACS: A Dataset for Physical Audiovisual CommonSense Reasoning". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Aonan Zhang, Quan Wang, Zhenyao Zhu, John Paisley, and Chong Wang (2019). "Fully supervised speaker diarization". In: *Proc. ICASSP*.

Chen-Lin Zhang, Jianxin Wu, and Yin Li (2022). "ActionFormer: Localizing Moments of Actions with Transformers". In: *European Conference on Computer Vision*. LNCS.

Chuhan Zhang, Ankush Gupta, and Andrew Zisserman (2021). "Temporal Query Networks for Fine-grained Video Understanding". In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hang Zhang, Xin Li, and Lidong Bing (2023). "Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding". In: *arXiv preprint arXiv:2306.02858*. URL: https://arxiv.org/abs/2306.02858.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz (2018). "mixup: Beyond Empirical Risk Minimization". In: *Proceedings of International Conference on Learning Representations (ICLR)*.

Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li (2017). "S3fd: Single shot scale-invariant face detector". In: *Proc. ICCV*.

Yuan Zhang, Regina Barzilay, and Tommi Jaakkola (2017). "Aspect-augmented adversarial networks for domain adaptation". In: *Transactions of the Association for Computational Linguistics*.

Yunhua Zhang, Hazel Doughty, Ling Shao, and Cees GM Snoek (2022). "Audio-adaptive activity recognition across video domains". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba (2018). "The sound of pixels". In: *Proceedings of the European conference on computer vision (ECCV)*.

Yue Zhao and Philipp Krähenbühl (2023). "Training a large video model on a single machine in a day". In: *arXiv preprint arXiv:2309.16669*.

Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar (2023). "Learning Video Representations from Large Language Models". In: *CVPR*.

Yu Zheng, Yajun Zhang, Chuanying Niu, Yibin Zhan, Yanhua Long, and Dongxing Xu (2023). "Unisound system for voxceleb speaker recognition challenge 2023". In: *arXiv preprint arXiv:2308.12526*.

Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren (2020). "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression". In: *The AAAI Conference on Artificial Intelligence (AAAI)*.

Zeyun Zhong, David Schneider, Michael Voit, Rainer Stiefelhagen, and Jürgen Beyerer (2023). "Anticipative feature fusion transformer for multi-modal action anticipation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba (2018). "Temporal relational reasoning in videos". In: *Proceedings of European Conference on Computer Vision (ECCV)*.

Jianfeng Zhou, Tao Jiang, Lin Li, Qingyang Hong, Zhe Wang, and Bingyin Xia (2019). "Training multi-task adversarial network for extracting noise-robust speaker embedding". In: *Proc. ICASSP*.

Jinxing Zhou, Dan Guo, and Meng Wang (2022). "Contrastive Positive Sample Propagation along the Audio-Visual Event Line". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jinxing Zhou, Liang Zheng, Yiran Zhong, Shijie Hao, and Meng Wang (2021). "Positive Sample Propagation along the Audio-Visual Event Line". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Weizhong Zhu and Jason Pelecanos (2016). "Online speaker diarization using adapted i-vector transforms". In: *Proc. ICASSP*. IEEE.

Appendix A

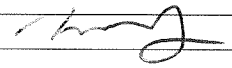
Statement of Authorship

A statement of authorship is provided for each multi-authored paper included in this thesis. The statements describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication, there exists a complete statement that is filled out and signed by the candidate and supervisor.

Statement of Authorship for the paper "With a Little Help from my Temporal Context: Multimodal Egocentric Action Recognition" in Chapter 2.

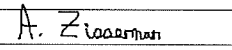
Paper title	With a Little Help from my Temporal Context: Multimodal Egocentric Action Recognition
Authors	Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, Dima Damen
Publication status	Published
Publication details	British Machine Vision Conference (BMVC), 2021.

Student Confirmation

Student name	Jaesung Huh	
Contribution to the paper	• conception of research ideas • implementation of action language model and running all experiments related to this • writing and presentation of the paper	
Signature and Date		Oct. 11th 2024

Supervisor Confirmation

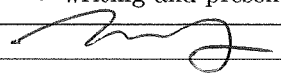
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date		Oct. 11th 2024

Statement of Authorship for the paper "TIM: A Time Interval Machine for Audio-Visual Action Recognition" in Chapter 3.

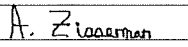
Paper title	TIM: A Time Interval Machine for Audio-Visual Action Recognition
Authors	Jacob Chalk, Jaesung Huh, Evangelos Kazakos, Andrew Zisserman, Dima Damen
Publication status	Published
Publication details	Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

Student Confirmation

Student name	Jaesung Huh	
Contribution to the paper	Co first-author contribution: • conception of research ideas • design and implementation of the models • running of large-scale experiments • writing and presentation of the paper	
Signature and Date		Oct. 11th 2024

Supervisor Confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date		Oct. 11th 2024

Statement of Authorship for the paper "Look, Listen and Recognise: character-aware audio-visual subtitling" in Chapter 4.

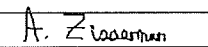
Paper title	Look, Listen and Recognise: character-aware audio-visual subtitling
Authors	Bruno Korbar, Jaesung Huh, Andrew Zisserman
Publication status	Published
Publication details	International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2024.

Student Confirmation

Student name	Jaesung Huh	
Contribution to the paper	Co first-author contribution: • conception of research ideas • curation of LLR dataset • implementation of the audio pipeline • writing and presentation of the paper	
Signature and Date		Oct. 11th 2024

Supervisor Confirmation

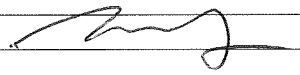
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date		Oct. 11th 2024

Statement of Authorship for the paper "Character-aware audio-visual subtitling in context" in Chapter 5.

Paper title	Character-aware audio-visual subtitling in context
Authors	Jaesung Huh, Andrew Zisserman
Publication status	Accepted and will be presented in December, 2024
Publication details	Asian Conference on Computer Vision (ACCV), 2024.

Student Confirmation

Student name	Jaesung Huh	
Contribution to the paper	First-author contribution: • conception of research ideas • running of large-scale experiments • implementation of the whole pipeline • writing and presentation of the paper	
Signature and Date		Oct. 11th 2024

Supervisor Confirmation

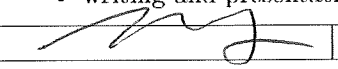
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date		Oct. 11th 2024

Statement of Authorship for the paper "Spot the conversation: speaker diarisation in the wild" in Chapter 6.

Paper title	Spot the conversation: speaker diarisation in the wild
Authors	Joon Son Chung, Jaesung Huh, Arsha Nagrani, Triantafyllos Afouras, Andrew Zisserman
Publication status	Published
Publication details	Interspeech, 2020.

Student Confirmation

Student name	Jaesung Huh	
Contribution to the paper	Co first-author contribution: • conception of research ideas • curation of the VoxConverse dataset • running baseline experiments • writing and presentation of the paper	
Signature and Date		Oct. 11th 2024

Supervisor Confirmation

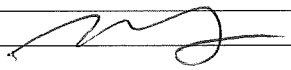
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date		Oct. 11th 2024

Statement of Authorship for the paper "Playing a Part: Speaker Verification at the Movies" in Chapter 7.

Paper title	Playing a Part: Speaker Verification at the Movies
Authors	Andrew Brown, Jaesung Huh, Arsha Nagrani, Joon Son Chung, Andrew Zisserman
Publication status	Published
Publication details	International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2021.

Student Confirmation

Student name	Jaesung Huh	
Contribution to the paper	Co first-author contribution: • conception of research ideas • running all experiments • writing and presentation of the paper	
Signature and Date		Oct. 11th 2024

Supervisor Confirmation

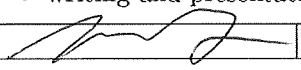
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date	A. Zisserman	Oct. 11th 2024

Statement of Authorship for the paper "Epic-Sounds: A Large-scale Dataset of Actions That Sound" in Chapter 8.

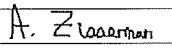
Paper title	Epic-Sounds: A Large-scale Dataset of Actions That Sound
Authors	Jaesung Huh, Jacob Chalk, Evangelos Kazakos, Dima Damen, Andrew Zisserman
Publication status	Published
Publication details	International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2023.

Student Confirmation

Student name	Jaesung Huh	
Contribution to the paper	Co first-author contribution: • conception of research ideas • curation of the whole dataset • implementation of the baseline models • writing and presentation of the paper	
Signature and Date		Oct. 11th 2024

Supervisor Confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date		Oct. 11th 2024