Understanding Video Through the Lens of Language

Supervised by Professor Andrew Zisserman

Max Bain

Christ Church

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2023

Scientists discover the world that exists;

Engineers create the world that never was.

–Theodore von Kármán

Acknowledgements

Whilst much of this DPhil was spent in lone engagement with my machine, a ship needs its crew, and so did I.

Foremost, I owe my gratitude to my supervisor, Andrew Zisserman, whose enthusiasm and wisdom shaped me into the researcher I am today. From the earliest days, when plotting a precision-recall curve was an Everest of its own, his patience has been unmatched, perhaps only by my propensity to detour from the planned path.

To Arsha Nagrani, whose mentorship gave me confidence to tackle any research problem, no matter how daunting, or pursue any deadline, no matter its imminence. To Gül Varol, for teaching me to focus on doing good research and that impact will follow. To my collaborators: Tengda Han, Weidi Xie, Jaesung Huh, Andrew Brown, Alexander Shtedritski and Hannah Kirk. And to the all remarkable members of VGG, especially Guanqi, Liliane, Sagar, Samuel, Shu, Stan and Tim; as well as the supportive Abhishek and Ashish. It was truly the greatest honour to work in their midst.

We are, of course, moulded by the company we keep, and my dear friends and protectors played no small part. Despite me seldom voicing this appreciation, I am eternally in their debt. Asco, Prem and Tranter, forever standing by me, quirks and all. Henry and Ziggy, the joy in my triumphs and the calm in my tempests. Viktoria, who guided me towards sense. Fé and Ollie, for the ascents and tumbles. Sanctuaries of Christ Church, OU Gymnastics, UCS and teachers – notably Ms. Isaac and Mr. Wilkes. And to Gabrielle, for brightening my day, beyond any ocean's part.

Lastly, my family and kin. To my sister Shaska, for her noble work in keeping me humble. My brother Carlos, whose faith in me never waivers. Carol, Bryn and Harriet, for nurturing the engineer within me. And to my father, whose lessons and sacrifices warrant a thesis of their own.

In gratitude, I present this work.

Abstract

The increasing abundance of video data online necessitates the development of systems capable of understanding such content. However, building these systems poses significant challenges, including the absence of scalable and robust supervision signals, computational complexity, and multimodal modelling. To address these issues, this thesis explores the role of language as a complementary learning signal for video, drawing inspiration from the success of self-supervised Large Language Models (LLMs) and image-language models.

First, joint video-language representations are examined under the text-to-video retrieval task. This includes the study of pre-extracted multimodal features, the influence of contextual information, joint end-to-end learning of both image and video representations, and various frame aggregation methods for long-form videos. In doing so, state-of-the-art performance is achieved across a range of established video-text benchmarks.

Second, this work explores the automatic generation of audio description (AD) – narrations describing the visual happenings in a video, for the benefit of visually impaired audiences. An LLM, prompted with multimodal information, including past predictions, and pretrained with partial data sources, is employed for the task. In the process, substantial advancements are achieved in the following areas: efficient speech transcription, long-form visual storytelling, referencing character names, and AD time-point prediction.

Finally, audiovisual behaviour recognition is applied to the field of wildlife conservation and ethology. The approach is used to analyse vast video archives of wild primates, revealing insights into individual and group behaviour variations, with the potential for monitoring the effects of human pressures on animal habitats.

Keywords – video understanding, deep learning, vision & language, multimodal

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.

Max Bain, July 2023.

Contents

Chapter 1

Introduction and Background

Video data is one of the most abundant and fast-growing forms of digital information in the world today. Every second, hours of footage are uploaded onto online platforms, and video is increasingly integral to a myriad of applications and monitoring systems, spanning from entertainment to critical infrastructure. This ever-increasing trove of data is rich in nature, encapsulating the complexity of our world in a multisensory, temporal format that not only offers visual cues, but also encompasses audio, and complementary metadata. The applications of automated video understanding systems are wide-ranging. For instance, in autonomous driving systems, the ability to accurately interpret real-time video feeds is pivotal for safe navigation [Janai et al. 2020]. Similarly, in the entertainment industry, understanding video can allow for automatic content generation [J. Liu et al. 2021], enhanced user interaction, and it can even assist visually impaired individuals perceive the video content [Yuksel et al. 2020]. Given these factors, developing systems that can understand and interpret video data effectively has become a key objective in the field of computer vision.

However, harnessing the full potential of video data is not without its challenges. From a practical standpoint, video data, due to its high-dimensionality, is far more expensive computationally to train and predict on compared to other data types. Each video consists of numerous frames, each a high-resolution image in itself, leading to significant computational and storage costs. Furthermore, the selection of an appropriate supervision signal for video learning presents its own challenges. Thus far, self-supervised **video** models in computer vision have not

enjoyed the same level of success as their image or language counterparts, in part due to the difficulty in defining a robust and consistent supervision signal. Another daunting challenge lies in the evaluation of video understanding. The goal of long-form video understanding, while central to the advancement of computer vision, is elusive due to the broad and somewhat intangible nature of "understanding". Designing metrics to accurately gauge the success of such understanding poses significant difficulties, which further complicate the progression in this research area.

In light of these challenges, this thesis argues that the use of language as a supervisory and evaluatory signal for video understanding can bring significant benefits. By linking the visually rich medium of video with the conceptually dense medium of language, we can leverage the descriptive and interpretive power of language to guide video understanding systems. The subsequent chapters of this thesis will delve into the specifics of this approach, its benefits, and the ways in which it can potentially shape the future of video understanding in computer vision.

1.1 Motivation

Learning from Language. Language, a principal form of human communication, inherently carries the ability to anticipate and explain the world. This thesis is motivated by the possibility of employing language as a guide to video learning, inspired by the triumph of Large Language Models (LLMs) and image-language models. The advent of self-supervised LLMs bears testimony to the immense potential intrinsic to pure language data generated by humans, either as written text on the internet or as spoken dialogue. These models have exhibited remarkable competencies across a diverse range of tasks, such as coding [M. Chen et al. 2021], translation [Vaswani et al. 2017], summarisation [Raffel et al. 2020], and even passing high school examinations [T. Brown et al. 2020].

The prospect of applying these robust capabilities to video understanding is enticing and promises substantial progress. In addition to enhancing video understanding, language also provides an avenue for accessing a colossal store of human knowledge. The vastness and diversity of the available textual data, covering an

extensive range of topics, present a tremendous opportunity. By employing these large corpora, models can be trained to comprehend video content in a similar nature to humans - associating visual content with stored knowledge and experiences, thereby rendering the video understanding process more intuitive and effective.

In essence, the overarching motivation of this thesis is to bridge the gap between the world of videos and the realm of language, integrating the two to enhance video understanding, by harnessing the recent advancements in language understanding models. This combination offers a unique and highly promising avenue to tackle the complex challenge of large-scale video understanding.

Learning from complementary modalities. The remarkable accomplishments in computer vision tasks owe a significant debt to vast datasets of annotated training examples [Russakovsky et al. 2015], the backbone of deep learning. However, manually obtaining these annotations is a cost-intensive process, severely constrained by the burgeoning volume of data available online. Efforts to circumvent these issues have increasingly turned to self-supervised learning, an approach that exploits inherent structures in the data to enable 'learning from the data' [Hinton and Salakhutdinov 2006]. This strategy has been vigorously researched and applied to image-only [Zbontar et al. 2021; Oquab et al. 2023; K. He et al. 2022; Grill et al. 2020; Caron et al. 2021; T. Chen et al. 2020; Y. M. Asano et al. 2019] or video-only [D. Wei et al. 2018; T. Han et al. 2019] data with some degrees of success. Nevertheless, these attempts have struggled to scale effectively and exhibit the same level of effortless low-shot adaptation to downstream applications observed in Natural Language Processing (NLP) tasks [T. Brown et al. 2020]. An effective workaround to this constraint lies in leveraging complementary modalities as supervision, that has shown promising results in video-audio [Alwassel et al. 2020; Nagrani 2020; Afouras et al. 2022; Korbar et al. 2018; R. Arandjelovic and Zisserman 2018; Y. Asano et al. 2020] and vision-language [C. Jia et al. 2021; Karpathy and Fei-Fei 2015]. Herein lies the importance of human dialogue, speech, and textual descriptions that often accompany human-uploaded video data. These natural co-occurrences of language and visual data, when in correspondence, can serve as a scalable and potent learning signal.

Applications for Video Understanding. Another critical motivation for this thesis is the application of video understanding in the field of wildlife conserva-

tion and ethology - the scientific study of animal behavior. Technological advances have made it possible to accumulate large volumes of video data, capturing animal behaviors in unprecedented detail [Tinbergen 1963]. Large-scale video archives, encompassing both visual and audio information, present immense potential to identify individual and population-level variations, ontogenetic and cultural changes in behaviors over extensive temporal and spatial scales.

However, the scale and depth at which this data can be analyzed are currently limited due to the immense computational requirements and intensive human effort needed to process these large volumes of video data [Jens Krause et al. 2013]. The application of video understanding techniques can potentially automate the measurement of animal behavior, thereby opening up large-scale video archives for detailed analysis.

The novel field of computational ethology has rapidly emerged at the intersection of computer science, engineering, and biology, leveraging advanced deep learning methodologies to process massive volumes of data [D. J. Anderson and Perona 2014]. The goal is to automate animal behavior recognition in wild footage, a task that presents significant challenges due to motion blur, occlusion, vegetation, poor resolution, and challenging lighting conditions [Sturman et al. 2020; van Dam et al. 2020]. Successfully implementing these tools will revolutionize ethological research and conservation, offering detailed insights into the behaviors of wild animals and the impact of anthropogenic pressures on their habitats [Kaufhold and Van Leeuwen 2019; Cantor et al. 2021; Dominoni et al. 2020; Christiansen et al. 2013].

1.2 Key Ideas

1.2.1 Bottom-Up Multimodal Video Representations

A central concept of this thesis is recognising that a video is a symphony of multiple constituent modalities: visual frames (a sequence of images), audio, as well as accompanying metadata such as subtitles or user-generated annotations. This perspective motivates a bottom-up approach to video understanding, proposing

that a system should first comprehend these composite modalities individually before holistically understanding them in combination. Such a system should first possess the capability to reason about individual frames, audio files (which could be conceptually viewed as a video with a black screen), or textual data (such as subtitles or on-screen text).

This bottom-up approach presents several notable advantages. Firstly, it allows for the leveraging of existing datasets and pretrained models for each constituent modality (Chapter 2). For instance, image-alt text datasets can facilitate the understanding of individual frames (Chapter 3); pretrained audio models can be used to extract features (Chapter 8) and dialogue (Chapter 5); and LLMs pretrained on vast text corpora can be used to understand subtitles and additional metadata (Chapters 6 and 7).

Given that these resources are often available at a billion-scale level, they provide a robust representations for the video's constituent modalities and can do much of the heavy lifting. Such an approach can considerably reduce the required volume of labelled video data, which is often expensive to obtain, store, and train on. This aspect is particularly critical in academic settings where access to large-scale annotated video data and computational resources may be constrained.

1.2.2 Learning from Movies

The second key idea of this thesis is to learn and evaluate video understanding systems from movies. Many established tasks in the computer vision field, such as classification or retrieval, do not necessarily require holistic video modelling. In many instances, these tasks can be solved by examining a single or few correct frames (Chapter 4) or focusing on short, constrained video clips. Long-form video, however, presents a completely different set of challenges, including modelling objects and interactions over long temporal sequences. Adding to this complexity is the lack of sufficient training data for more involved movie understanding tasks.

Movies, in this context, present a unique solution. They embody long-form narrative structures, modelling complex character-focused dynamics, events, relationships, emotions, and actions [Cutting 2016; Papalampidi et al. 2019]. This complexity makes movies a robust resource for training video understanding systems

to handle intricate, real-world scenarios. Furthermore, movies come with a wealth of rich, complementary data such as plots (Chapter 2), scripts, and audio descriptions (Chapter 6 and 7). This additional information provides valuable context that can greatly enhance a system's understanding of the video content. The fact that these sources of data are produced naturally by the movie industry, makes them an ideal, scalable supervision signal alongside movies.

1.3 Thesis Outline and Contributions

In this section, the contributions of this thesis are summarised and an overview is provided for each chapter. The thesis is divided into three parts – (i) Joint Video-Text Representations for Retrieval; (ii) Automated Movie Audio Description; and (iii) Audiovisual Animal Behaviour Recognition.

Joint Video-Text Representations for Retrieval

In this research theme, video understanding is evaluated under the text-to-video retrieval task. This process involves querying a gallery of videos using a piece of text, such as a sentence, with the object of retrieving the video that best corresponds to the queried text. The performance of joint video-text embeddings and how well they can be aligned are assessed through various retrieval metrics – providing a basis video-language representaitons and more complex tasks.

In Chapter 2, an in-depth exploration of text-to-video retrieval, specifically within the context of captioned movie clip videos, is conducted. The study evaluates the effectiveness of (i) pre-extracted multimodal features, (ii) the identification of movie characters in the clip, and (iii) the impact of contextual information derived from adjacent clips within the same film.

In Chapter 3, we propose to move beyond pre-extracted video features for retrieval, and instead learn a visual encoder end-to-end for joint video-text representations. The paper investigates the effect of leveraging image-caption data to boost video-text learning, along with a study into curriculum learning on the number of frames to heavily reduce computational requirements of training a video model end-to-

end.

Building upon the insights derived from Chapter 3, the study in Chapter 4 employs strong pretrained image-text representations, initializing the per-frame image encoder and text encoder with CLIP [Radford et al. 2021]. Subsequently, the chapter explores a range of frame aggregation methods for retrieval and fine-tunes them on various benchmarks, surpassing previous state-of-the-art performance. These outcomes underscore the potency of robust image-language representations in tackling established video retrieval benchmarks.

Automated Movie Audio Description

Under this theme, the assessment of video understanding is conducted through the task of densely captioning long-form videos, specifically, the generation of audio descriptions for movies. Audio description (AD) is a form of narration used to provide information of the visual happenings in the scene for the benefit of blind and visually impaired audiences. It is a legal requirement for broadcasters in the US and UK to provide AD for a fraction of their content and therefore this is a complementary source of language supervision for movie data, making it an ideal learning candidate.

We first propose an efficient and accurate method for automatic speech transcription, in order to automatically extract AD narrations from a single audio file containing both movie dialogue and AD. The proposed *WhisperX* method enables rapid transcription at scale, that we use in a pipeline, combined with speaker diarization, to extract and transcribe AD for 8,000 movies in under 200 hours on a single GPU.

The acquired AD data is then explored in Chapter 6, which investigates the use of a pretrained large language model (LLM) prompted with multimodal information to autonomously generate audio descriptions for a proposed temporal segment. It suggests partial pretraining as a strategy to address the challenge of incomplete data (whereby e.g. audio and text data is available but not frames), a frequent occurrence due to copyright restrictions. It indicates that each module of the visual captioning system can benefit from pretraining with partial data for each modality, such as large-scale text-only corpora. Additionally, the chapter delves into the

influence of longer context, demonstrating that the AD generation task necessitates a more extended context to deliver consistent storytelling for the listener.

Chapter 7 further expands on this work, addressing some of the limitations identified in the initial AD work. This includes handling character naming, a critical aspect for successful storytelling and superior AD, alongside improved visual reasoning, and time-point prediction. This second aspect enables the model to predict not only 'what' to generate for AD but also 'when', further enhancing its applicability.

Audiovisual Animal Behaviour Recognition

Lastly, this thesis presents an interesting cross-disciplinary application of video understanding systems: the recognition of wild primate behavior using audiovisual data from video footage (Chapter 8). This research illustrates how bottom-up approaches to video understanding, such as action classification, can offer automatic methods for wildlife researchers and animal behavior analysts. These methods can subsequently be employed for statistical analysis and sequencing, thus contributing to the broader field of animal behavior studies.

1.3.1 Publications

Chapters 2 to 8 each contain a paper that has been peer-reviewed and accepted at a conference or journal, with the exception of the technical report in Chapter 4. These papers are presented in their original published forms, with the only alterations made being those related to formatting. For every publication, a corresponding statement of authorship can be found in Appendix A. The papers included in the thesis are as follows:

Chapter 2: Condensed Movies: Story-Based Retrieval with Contextual Embeddings

Max Bain, Arsha Nagrani, Andrew Brown, Andrew Zisserman

In *Asian Conference on Computer Vision (ACCV)*, 2020.

Chapter 3: Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval

Max Bain, Arsha Nagrani, Gül Varol, Andrew Zisserman

In *International Conference on Computer Vision (ICCV)*, 2021.

Chapter 4: A Clip-Hitchhiker's Guide to Long Video Retrieval

Max Bain, Arsha Nagrani, Gül Varol, Andrew Zisserman

Technical Report, 2022.

Chapter 5: WhisperX: Time-Accurate Speech Transcription of Long-Form Audio

Max Bain, Jaesung Huh, Tengda Han, Andrew Zisserman

In *INTERSPEECH*, 2023.

Chapter 6: AutoAD: Movie Description in Context

Tengda Han*, **Max Bain***, Arsha Nagrani, Gül Varol, Weidi Xie, Andrew Zisserman (*Equal contribution)

In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Chapter 7: AutoAD II: The Sequel – Who, When, and What in Movie Audio Description

Tengda Han, **Max Bain**, Arsha Nagrani, Gül Varol, Weidi Xie, Andrew Zisserman

In *International Conference on Computer Vision (ICCV)*, 2023.

Chapter 8: Automated Audiovisual Behavior Recognition in Wild Primates

Max Bain, Arsha Nagrani, Daniel Schofield, Sophie Berdugo, Joana Bessa, Jake Owen, Kimberley J. Hockings, Tetsuro Matsuzawa, Misato Hayashi, Dora Biro, Susana Carvalho, Andrew Zisserman

Science Advances 7, no. 46, 2021.

Publications not included:

"Count, Crop and Recognise: Fine-Grained Recognition in the Wild"
Max Bain, Arsha Nagrani, Daniel Schofield, Andrew Zisserman. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

"A Prompt Array Keeps the Bias Away: Debiasing Vision-Language Models with Adversarial Learning"[1]
Hugo Berg, Siobhan Mackenzie Hall, Yash Bhalgat, Wonsuk Yang, Hannah Rose Kirk, Alexander Shtedritski, **Max Bain**. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (AACL)*, 2022.

"Balancing the Picture: Debiasing Vision-Language Datasets with Synthetic Contrast Sets"[1]
Brandon Smith, Miguel Farinha, Siobhan Mackenzie Hall, Hannah Rose Kirk, Alexander Shtedritski, **Max Bain**. *Preprint (under review)*, 2023.

[1] These papers, focussed on the topic of societal bias in vision-language models, are only loosely related to the scope of the thesis and hence are excluded.

Part I

Joint Video-Text Representations for Retrieval

Chapter 2

Condensed Movies: Story-Based Retrieval with Contextual Embeddings

The paper has been accepted for publication as an oral presentation at the Asian Conference on Computer Vision (ACCV), 2020.

Condensed Movies:
Story Based Retrieval with Contextual Embeddings

Max Bain Arsha Nagrani
Andrew Brown Andrew Zisserman
Visual Geometry Group, University of Oxford

Abstract

Our objective in this work is long range understanding of the narrative structure of movies. Instead of considering the entire movie, we propose to learn from the 'key scenes' of the movie, providing a `condensed` look at the full storyline. To this end, we make the following three contributions: (i) We create the `Condensed Movies Dataset (CMD)` consisting of the key scenes from over 3K movies: each key scene is accompanied by a high level semantic description of the scene, character face-tracks, and metadata about the movie. The dataset is scalable, obtained automatically from YouTube, and is freely available for anybody to download and use. It is also an order of magnitude larger than existing movie datasets in the number of movies; (ii) We provide a deep network baseline for text-to-video retrieval on our dataset, combining character, speech and visual cues into a single video embedding; and finally (iii) We demonstrate how the addition of context from other video clips improves retrieval performance.

2.1 Introduction

Imagine you are watching the movie *'Trading Places'*, and you want to instantly fast forward to a scene, one where 'Billy reveals the truth to Louis about the Duke's bet, a bet which changed both their lives'. In order to solve this task automatically, an intelligent system would need to watch the movie up to this

Figure 2.1: **Condensed Movies:** The dataset consists of the key scenes in a movie (ordered by time), together with high level semantic descriptions. Note how the caption of a scene (far right) is based on the knowledge of past scenes in the movie – one where the Dukes exchange money to settle their bet (highlighted in yellow), and another scene showing their lives before the bet, homeless and pan-handling (highlighted in green).

point, have knowledge of Billy, Louis and the Duke's identities, understand that the Duke made a bet, and know the outcome of this bet (Fig. 7.1). This high level understanding of the movie narrative requires knowledge of the characters' identities, their relationships, motivations and conversations, and ultimately their behaviour. Since movies and TV shows can provide an ideal source of data to test this level of story understanding, there have been a number of movie related datasets and tasks proposed by the computer vision community [Tapaswi et al. 2014; A. Rohrbach et al. 2017b; Xiong et al. 2019; Vicol et al. 2018; Q. Huang et al. 2020b].

However, despite the recent proliferation of movie-related datasets, high level semantic understanding of human narratives still remains a challenging task. There are a number of reasons for this lack of progress: (i) semantic annotation is expensive and challenging to obtain, inherently restricting the size of current movie datasets to only hundreds of movies, and often, only part of the movie is annotated in detail [Tapaswi et al. 2014; A. Rohrbach et al. 2017b; Xiong et al. 2019]; (ii) movies are very long (roughly 2 hours) and video architectures struggle to learn over such large timescales; (iii) there are legal and copyright issues surrounding a majority of these datasets [Tapaswi et al. 2014; Xiong et al. 2019], which hinder their widespread availability and adoption in the community; and finally (iv) the subjective nature of the task makes it difficult to define objectives and metrics.

A number of different works have recently creatively identified that certain domains of videos, such as narrated instructional videos [Miech et al. 2019; Y. Tang et al.

2019; L. Zhou et al. 2018a] and lifestyle vlogs [Ignat et al. 2019; Fouhey et al. 2018] are available in large numbers on YouTube and are a good source of supervision for video-text models as the speech describes the video content. In a similar spirit, videos from the MovieClips channel on YouTube[1], which contains the key scenes or clips from numerous movies, are also accompanied by a semantic text description describing the content of each clip.

Our first objective in this paper is to curate a dataset, suitable for learning and evaluating long range narrative structure understanding, from the available video clips and associated annotations of the MovieClips channel. To this end, we curate a dataset of 'condensed' movies, called the Condensed Movie Dataset (CMD) which provides a *condensed* snapshot into the entire storyline of a movie. In addition to just the video, we also download and clean the high level semantic descriptions accompanying each key scene that describes characters, their motivations, actions, scenes, objects, interactions and relationships. We also provide labelled face-tracks of the principal actors (generated automatically), as well as the metadata associated with the movie (such as cast lists, synopsis, year, genre). Essentially, all the information required to (sparsely) generate a MovieGraph [Vicol et al. 2018]. The dataset consists of over 3000 movies.

Previous work on video retrieval and video understanding has largely treated video clips as independent entities, divorced from their context [A. Rohrbach et al. 2017b; J. Xu et al. 2016; Anne Hendricks et al. 2017]. But this is not how movies are understood: the meaning and significance of a scene depends on its relationship to previous scenes. This is true also of TV series, where one episode depends on those leading up to it (the season arc); and even an online tutorial/lesson can refer to previous tutorials. These contextual videos are beneficial and sometimes even necessary for complete video understanding.

Our second objective is to explore the role of context in enabling video retrieval. We define a text-to-video retrieval task on the CMD, and extend the popular Mixture of Embedding Experts model [Miech et al. 2018], that can learn from the subtitles, faces, objects, actions and scenes, by adding a *Contextual Boost Module* that introduces information from past and future clips. Unlike other movie related tasks – e.g. text-to-video retrieval on the LSMDC dataset [A. Rohrbach

[1] https://www.youtube.com/user/movieclips

et al. 2017b] or graph retrieval on the MovieQA [Tapaswi et al. 2016] dataset that ignore identities, we also introduce a character embedding module which allows the model to reason about the identities of characters present in each clip and description. Applications of this kind of story-based retrieval include semantic search and indexing of movies as well as intelligent fast forwards. The CMD dataset can also be used for semantic video summarization and automatic description of videos for the visually impaired (Descriptive Video Services (DVS) are currently available at a huge manual cost).

Finally, we also show preliminary results for aligning the semantic captions to the plot summaries of each movie, which places each video clip in the larger context of the movie as a whole. Data, code, models and features can be found at https://www.robots.ox.ac.uk/~vgg/research/condensed-movies/.

2.2 Related Work

Video Understanding from Movies: There is an increasing effort to develop video understanding techniques that go beyond action classification from cropped, short temporal snippets [Kay et al. 2017; C. Gu et al. 2018; Monfort et al. 2019], to learning from longer, more complicated videos that promise a higher level of abstraction [Sener et al. 2015; Alayrac et al. 2016; Miech et al. 2019; C. Sun et al. 2019]. Movies and TV shows provide an ideal test bed for learning long-term stories, leading to a number of recent datasets focusing exclusively on this domain [Tapaswi et al. 2014; Tapaswi et al. 2016; A. Rohrbach et al. 2017b; Xiong et al. 2019]. Early works, however, focused on using film and TV to learn human identity [Everingham et al. 2006; Naim et al. 2016; Cour et al. 2009; Sivic et al. 2009; Tapaswi et al. 2012a; Q. Huang et al. 2020c] or human actions [Bojanowski et al. 2013; Duchenne et al. 2009; Laptev et al. 2008; Marszałek et al. 2009; Nagrani et al. 2020] from the scripts or captions accompanying movies. Valuable recent works have proposed story-based tasks such as the visualization and grouping of scenes which belong to the same story threads [Ercolessi et al. 2012; Rao et al. 2020], the visualization of TV episodes as a chart of character interactions [Tapaswi et al. 2014], and more recently, the creation of more complicated movie graphs (MovieGraphs [Vicol et al. 2018] is the most exhaustively annotated

Table 2.1: Comparison to other movie and TV show datasets. For completeness, we also compare to datasets that *only* have character ID or action annotation. 'Free' is defined here as accessible online at no cost at the time of writing. *Refers to number of TV shows.

	#Movies	#Hours	Free	Annotation Type
Sherlock[Nagrani and Zisserman 2017]	1*	4		Character IDs
TVQA[Lei et al. 2019]	6*	460		VQA
AVA[C. Gu et al. 2018]	430	107.5	✓	Actions only
MovieGraphs[Vicol et al. 2018]	51	93.9		Descriptions, graphs
MovieQA(video)[Tapaswi et al. 2016]	140	381		VQA
MovieScenes[Rao et al. 2020]	150	250		Scene segmentations
LSMDC[A. Rohrbach et al. 2017b]	202	158		Captions
MSA[Xiong et al. 2019]	327	516		Plots
MovieNet[Q. Huang et al. 2020b]	1,100	2,000		Plots, action tags, character IDs
CMD (Ours)	**3,605**	**1,270**	✓	Descriptions, metadata, character IDs, plots

movie dataset to date). Such graphs have enabled explicit learning of interactions and relationships [Kukleva et al. 2020a] between characters. This requires understanding multiple factors such as human communication, emotions, motivation, scenes and other factors that affect behavior. There has also been a recent interest in evaluating story understanding through visual question answering [Tapaswi et al. 2016] and movie scene segmentation [Rao et al. 2020]. In contrast, we propose to evaluate story understanding through the task of text-to-video retrieval, from a set of key scenes in a movie that condense most of the salient parts of the storyline. Unlike retrieval through a complex graph [Vicol et al. 2018], retrieval via text queries can be a more intuitive way for a human to interact with an intelligent system, and might help avoid some of the biases present inherently in VQA datasets [Jasani et al. 2019].

Comparison to other Movie Datasets: Existing movie datasets often consist of short clips spanning entire, full length movies (which are subject to copyright and difficult for public release to the community). All such datasets also depend on exhaustive annotation, which limit their scale to hundreds of movies. Our dataset, in contrast, consists of only the key scenes from movies matched with high quality, high level semantic descriptions, allowing for a condensed look at the entire storyline. A comparison of our dataset to other datasets can be seen in Table 2.1. **Text-to-Video Retrieval:** A common approach for learning visual embeddings from natural language supervision is to learn a joint embedding space where visual and textual cues are adjacent if they are semantically similar [Miech

et al. 2018; Y. Liu et al. 2019]. Most of these works rely on manually annotated datasets in which descriptive captions are collected for short, isolated video clips, with descriptions usually focusing on low-level visual content provided by annotators [A. Rohrbach et al. 2017b; Anne Hendricks et al. 2017; J. Xu et al. 2016]. For example LSMDC [A. Rohrbach et al. 2017b], which is created from DVS, contains mostly low-level descriptions of the visual content in the scene, e.g. 'Abby gets in the basket', unlike the descriptions in our dataset. Most similar to our work is [Tapaswi et al. 2015b], which obtains story level descriptions for shots in full movies, by aligning plot sentences to shots, and then attempting video retrieval. This, however, is challenging because often there is no shot that matches a plot sentence perfectly, and shots cover very small timescales. Unlike this work our semantic descriptions are more true to the clips themselves.

Temporal Context: The idea of exploiting surrounding context has been explored by [Krishna et al. 2017a], for the task of video captioning, and by [C.-Y. Wu et al. 2019a] for video understanding. Krishna *et al.* [Krishna et al. 2017a] introduces a new captioning module that uses contextual information from past and future events to jointly describe all events, however this work focuses on short term context (few seconds before and after a particular clip). Wu *et al.* [C.-Y. Wu et al. 2019a] go further, and introduce a feature bank architecture that can use contextual information over several minutes, demonstrating the performance improvements that results. Our dataset provides the opportunity to extend such feature banks (sparsely) over an entire movie.

2.3 Condensed Movie Dataset

We construct a dataset to facilitate machine understanding of narratives in long movies. Our dataset has the following key properties:

(1) Condensed Storylines: The video data consists of over 33,000 clips from 3,600 movies (see Table 2.2). For each movie there is a set of ordered clips (typically 10 or so) covering the salient parts of the film (examples can be seen in Fig. 2.2, top row). Each around two minutes in length, the clips contain the same rich and complex story as full-length films but an order of magnitude shorter. The distribution of video lengths in our dataset can be seen in Fig. 2.2 – with just the

Figure 2.2: **The Condensed Movie Dataset (CMD)**. *Top:* Samples of clips and their corresponding captions from *The Karate Kid (1984)* film. In movies, as in real life, situations follow from other situations and the combination of video and text tell a concise story. Note: Every time a character is mentioned in the description, the name of the actor is present in brackets. We remove these from the figure in the interest of space. *Middle, from left to right:* Histogram of movie genres, movie release years, description length and duration of video clips. Best viewed online and zoomed in. *Bottom:* Example face-tracks labelled with the actor's name in the clips. These labels are obtained from cast lists and assigned to facetracks using our automatic labelling pipeline.

Table 2.2: Comparison to other video text retrieval datasets. MTLD is the Measure of Textual Lexical Diversity [Mccarthy and Jarvis 2010] for all of the descriptions in the dataset.

Dataset	#Videos/#Clips	Median caption len. (words)	MTLD	Median clip len. (secs)
MSVRTT[J. Xu et al. 2016]	7,180/10,000	7	26.9	15
DiDemo[Anne Hendricks et al. 2017]	10,464/26,892	7	39.9	28
LSMDC[A. Rohrbach et al. 2017b]	200/118,114	8	61.6	5
CMD (Ours)	3,605/33,976	**18**	**89.1**	**132**

key scenes, each movie has been condensed into roughly 20 minutes each. Each clip is also accompanied by a high level description focusing on intent, emotion, relationships between characters and high level semantics (Figures 2.2 and 2.3). Compared to other video-text datasets, our descriptions are longer, and have a higher lexical diversity [Mccarthy and Jarvis 2010] (Table 2.2). We also provide face-tracks and identity labels for the main characters in each clip (Figure 2.2, bottom row).

(2) Online Longevity and Scalability: All the videos are obtained from the licensed, freely available YouTube channel: MovieClips[2] We note that a common problem plaguing YouTube datasets today [Caba Heilbron et al. 2015; C. Gu et al. 2018; Kay et al. 2017; Nagrani et al. 2019] is the fast shrinkage of datasets as user uploaded videos are taken down by users (over 15% of Kinetics-400 [Kay et al. 2017] is no longer available on YouTube at the time of writing, including videos from the eval sets). We believe our dataset has longevity due to the fact that the movie clips on the licensed channel are rarely taken down from YouTube. Also, this is an actively growing YouTube channel as new movies are released and added. Hence there is a potential to continually increase the size of the dataset. We note that from the period of 1st Jan 2020, to 1st September 2020, only 0.3% of videos have been removed from the YouTube channel, while an additional 2,000 videos have been uploaded, resulting in a dataset growth of 5.8% over the course of 9 months.

[2] https://www.youtube.com/user/movieclips/.

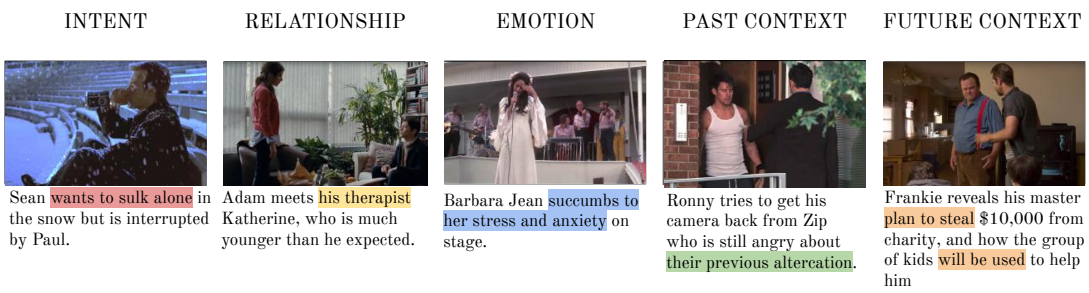


Figure 2.3: **Semantic descriptions:** Examples of high level semantic descriptions accompanying each video clip in our dataset (note: actor names are removed to preserve space). Our semantic descriptions cover a number of high level concepts, including intent/motivation, relationships, emotions and attributes, and context from surrounding clips in the storyline.

2.3.1 Dataset Collection Pipeline

In this section we describe the dataset collection pipeline.

Videos and Descriptions: Raw videos are downloaded from YouTube. Each video is accompanied by an outro at the end of the clip which contains some advertising and links to other movies. This is automatically removed by using the observation that each outro has a consistent length of either 10s (if the clip is uploaded before May 2017) or 30s if uploaded after. Approximately 1,000 videos from the channel were manually excluded from the dataset because they contained low quality descriptions or did not contain scenes from a movie. For each video, we also download the YouTube closed captions, these are a mix of high quality, human generated subtitles and automatic captions. Closed captions are missing for 36.7% of the videos. The MovieClips channel also provides a rich and high level description with each video, which we extract, clean (removing the movie title, links and advertising) and verify manually. We note that the videos also contain a watermark, usually at the bottom left of the frame. These can be easily cropped from the videos.

Metadata: For each clip, we identify its source movie by parsing the movie title from the video description and, if available, the release year (since many movies have non-unique titles). The title and release year are queried in the IMDb search engine to obtain the movie's IMDb ID, cast list and genre. IMDb identification enables correspondence to other popular movie datasets [Tapaswi et al. 2016; A. Rohrbach et al. 2017a]. Plot synposes were gathered by querying the movie title and release year in the Wikipedia search engine and extracting text within the

'Plot' section of the top ranked entry. For each movie we include: (i) the movie description (short, 3-5 sentences), accompaying the video clips on the MovieClips YouTube channel; (ii) Wikipedia plot summaries (medium, 30 sentences); and (iii) IMDB plot synopses (long, 50+ sentences).

Face-tracks and Character IDs: We note that often character identities are the focal point of any storyline, and many of the descriptions reference key characters. In a similar manner to [Nagrani and Zisserman 2017], we use face images downloaded from search engines to label detected and tracked faces in our dataset. Our technique involves the creation of a character embedding bank (CEB) which contains a list of characters (obtained from cast lists), and a corresponding embedding vector obtained by passing search engine image results through a deep CNN model pretrained on human faces [Cao et al. 2018]. Character IDs are then assigned to face-tracks in the video dataset when the similarity between the embeddings from the face tracks and the embeddings in the CEB (using cosine similarity) is above a certain threshold. This pipeline is described in detail in Section 2.8.1. We note that this is an automatic method and so does not yield perfect results, but a random manual inspection shows that it is accurate 96% of the time. Ultimately, we are able to recognize 8,375 different characters in 25,760 of the video clips.

2.3.2 Story Coverage

To quantitatively measure the amount of the story covered by movie clips in our dataset, we randomly sample 100 movies and manually aligned the movie clips (using the descriptions as well as the videos) to Wikipedia plot summaries (the median length of which is 32 sentences). We found that while the clips totalled only **15%** of the full-length movie in time duration, they cover **44%** of the full plot sentences, suggesting that the clips can indeed be described as key scenes. In addition, we find that the movie clips span a median range of **85.2%** of the plot, with the mean midpoint of the span being **53%**. We further show the distribution of clip sampling in Fig. 2.8 in the supplementary material of the ArXiV version, and find that in general there is an almost uniform coverage of the movie. While we focus on a baseline task of video-text retrieval, we also believe that the longitudindal nature of our dataset will encourage other tasks in long range movie

understanding.

2.4 Text-to-Video Retrieval

In this section we provide a baseline task for our dataset – the task of text-to-video retrieval. The goal here is to retrieve the correct 'key scene' over all movies in the dataset, given just the high level description. Henceforth, we use the term 'video clip' to refer to one key scene, and 'description' to refer to the high level semantic text accompanying each video clip. In order to achieve this task, we learn a common embedding space for each video and the description accompanying it. More formally, if V is the video and T is the description, we learn embedding functions f and g such that the similarity $s = \langle f(V), g(T) \rangle$ is high only if T is the correct semantic description for the video V. Inspired by previous works that achieve state-of-the-art results on video retrieval tasks [Miech et al. 2018; Y. Liu et al. 2019], we encode each video as a combination of different streams of descriptors. Each descriptor is a semantic representation of the video learnt by individual experts (that encode concepts such as scenes, faces, actions, objects and the content of conversational speech from subtitles).

Inspired by [Miech et al. 2018], we base our network architecture on a mixture of 'expert' embeddings model, wherein a separate model is learnt for each expert, which are then combined in an end-to-end trainable fashion using weights that depend on the input caption. This allows the model to learn to increase the relative weight of motion descriptors for input captions concerning human actions, or increase the relative weight of face descriptors for input captions that require detailed face understanding. We also note, however, that often the text query not only provides clues as to which expert is more valuable, but also whether it is useful to pay attention to a previous clip in the movie, by referring to something that happened previously, eg. 'Zip is *still* angry about their *previous altercation*'. Hence we introduce a Contextual Boost module (CBM), which allows the model to learn to increase the relative weight of a past video feature as well. A visual overview of the retrieval system with the CBM can be seen in Fig. 3.1. In regular movie datasets, the space of possible previous clips can be prohibitively large [Tapaswi et al. 2015b], however this becomes feasible with our *Condensed Movies* dataset.

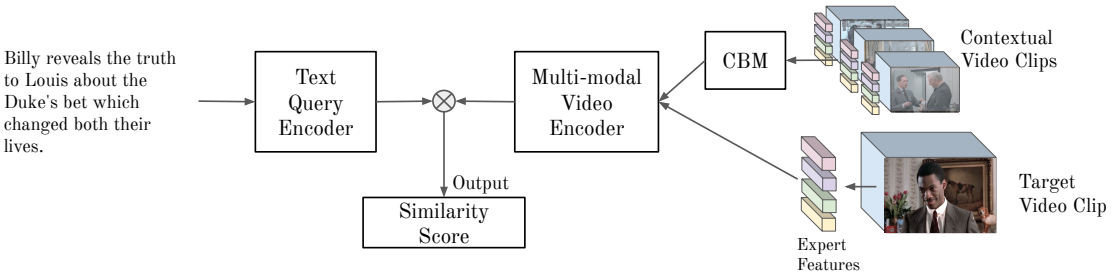


Figure 2.4: **Model architecture:** An overview of text-to-video retrieval with our Contextual Boost module (CBM) that computes a similarity score between a query sentence T and a target video. CBM receives contextual video features (which are previous clips from the same movie) to improve the multimodal encoding of the target video clip. The expert features are extracted using pre-trained models for speech, motion, faces, scenes and objects.

Besides doing just *cross-movie* retrieval, we also adapt our model to perform *within-movie* retrieval. We note that characters are integral to a storyline, and hence for the case of within-movie retrieval, we introduce a character module, which computes a weighted one-hot vector for the characters present in the description query and another for each video clip in the dataset. We note that for cross-movie retrieval, the retrieval task becomes trivial given the knowledge of the characters in each movie, and hence to make the task more challenging (and force the network to focus on other aspects of the story), we remove the character module for this case.

2.4.1 Model Architecture

Expert Features.

Stories in movies are communicated through many modalities including (but not limited to) speech, body language, facial expressions and actions. Hence we represent each input video V with K different expert streams (in our case, $K = 5$ – face, subtitles, objects, motion and scene, but our framework can be extended to more experts as required).

Each input stream is denoted as I_i, where $i = 1, ..., K$. Adopting the approach proposed by [Miech et al. 2018], we first aggregate the descriptors of each input stream over time, using a temporal aggregation module (see Sec. 2.5 for details), and the resulting time-aggregated descriptor is embedded using a gated embedding module (for the precise details of the gated embedding module, please see [Miech

et al. 2018]). We then finally project each embedding to a common dimension D using a fully connected layer, giving us one expert embedding E_{V_i} for each input stream i.Hence the final output is of dimensions $K \times D$.

Text Query Encoder.

The query description input is a sequence of BERT word embeddings [J. Devlin et al. 2019] for each input sentence. These individual word embedding vectors are then aggregated into a single vector $h(T)$ representing the entire sentence using a NetVLAD [R. Arandjelovic et al. 2016] aggregation module. This vector $h(T)$, is used to predict the mixture weights (described in the next section). We project $h(T)$ to the same dimensions as the video expert features using the same gated embedding module followed by a fully connected layer as for the video experts (described above), once for each input source i, giving us expert embeddings E_{T_i}. Hence the final output is also of dimensions $K \times D$.

Contextual Boost Module.

In both [Miech et al. 2018] and [Y. Liu et al. 2019], the resulting expert embeddings E_{V_i} are then weighted using normalized weights $w_i(T)$ estimated from the text description T. The final similarity score s is obtained by a weighted combination of the similarity scores $s_i(E_{T_i}, E_{V_i})$ between the embeddings E_{T_i} of the query sentence T and the expert embeddings E_{V_i} (obtained from the input video descriptors I_i). More formally, this is calculated as:

$$s(T,V) = \sum_{i=1}^{K} w_i(T) s_i(E_{T_i}, E_{V_i}), \qquad \text{where} \qquad w_i(T) = \frac{e^{h(T)^\intercal a_i}}{\sum_{j=1}^{K} e^{h(T)^\intercal a_j}} \tag{2.1}$$

where s_i is the scalar product, $h(T)$ is the aggregated text query representation described above and a_i, $i = 1, .., .K$ are learnt parameters used to obtained the mixture weights.

In this work, however, we extend this formulation in order to incorporate past context into the retrieval model. We would like the model to be able to predict weights for combining experts from previous clips – note we treat each expert separately in this formulation. For example, the model might want to heavily

weight the subtitles from a past clip, but downweight the scene representation which is not informative for a particular query. More formally, given the total number of clips we are encoding to be N, we modify the equation above as:

$$s(T, V) = \sum_{n=1}^{N} \sum_{i=1}^{K} w_{i,n}(T) s_{i,n}(E_{T_i}, E_{V_{i,n}}), \tag{2.2}$$

$$w_{i,n}(T) = \frac{e^{h(T)^\intercal a_{i,n}}}{\sum_{m=1}^{N} \sum_{j=1}^{K} e^{h(T)^\intercal a_{j,m}}}. \tag{2.3}$$

Hence instead of learning K scalar weights a_i, $i = 1, ..., K$ as done in [Miech et al. 2018] and [Y. Liu et al. 2019], we learn $K \times N$ scalar weights $a_{i,n}$, $i = 1, ..., K$, $n = 1, ..., N$ to allow combination of experts from additional clips.

Dealing with missing streams.

We note that these experts might be missing for certain videos, e.g. subtitles are not available for all videos and some videos do not have any detected faces. When expert features are missing, we zero-pad the missing experts and compute the similarity score. This is the standard procedure followed by existing retrieval methods using Mixture of Embedding Experts models [Miech et al. 2018; Y. Liu et al. 2019]. The similarity score is calculated only from the available experts by re-normalizing the mixture weights to sum to one, allowing backpropagation of gradients only to the expert branches that had an input feature. We apply this same principle when dealing with missing video clips in the past, for example if we are training our model with $N = 1$ past clips, for a video clip which is right at the start of the movie (has no past), we treat all the experts from the previous clip as missing so that the weights are normalized to focus only on the current clip.

Character Module.

The character module computes the similarity between a vector representation of the character IDs mentioned in the query y and a vector representation of the face identities recognised in the clip x. The vector representations are computed as follows: For the query, we search for actor names in the text from the cast list (supplied by the dataset) and create a one-hot vector y the same length as the cast list, where $y_i = 1$ if actor i is identified in any face track in the video and

$y_i = 0$ otherwise. For the face identities acquired in the face recognition pipeline (described earlier), we compare the following three methods: first, we encode a one-hot vector x in a manner similar to the query character encoding. While this can match the presence and absence of characters, it doesn't allow any weighting of characters based on their importance in a clip. Hence inspired by [Tapaswi et al. 2015c], we also propose a second method ("track-frequency normalised"), where x_i is the number of face tracks for identity i. Lastly, in "track length normalised", our vector encodes the total amount of time a character appears in a clip i.e. x_i is the sum of all track lengths for actor i, divided by the total sum of all track lengths in the clip. The performances of the three approaches are displayed and discussed in Table 2.5 and Section 2.5 respectively. The character similarity score $s_C = \langle y, x \rangle$ is then modulated by its own scalar mixture weight $w_C(T)$ predicted from $h(T)$ (as is done for the other experts in the model). This similarity score is then added to the similarity score obtained from the other experts to obtain the final similarity score, i.e. $s(T, V) = \sum_{i=1}^{K} w_i(T) s_i(E_{T_i}, E_{V_i}) + w_C(T) s_C(T, V)$.

Training Loss. As is commonly done for video-text retrieval tasks, we minimise the Bidirectional Max-margin Ranking Loss [Socher et al. 2014].

2.5 Experiments

2.5.1 Experimental Set-up

We train our model for the task of cross-movie and within-movie retrieval. The dataset is split into disjoint training, validation and test sets by movie, so that there are no overlapping movies between the sets. The dataset splits can be seen in Table 2.3. We report our results on the *test set* using standard retrieval metrics including median rank (lower is better), mean rank (lower is better) and R@K (recall at rank K—higher is better).

Cross-movie Retrieval: For the case of cross-movie retrieval, the metrics are reported over the entire test set of videos, i.e. given a text query, there is a 'gallery' set of 6,581 possible matching videos (Table 2.3). We report R@1, R@5, R@10, mean and median rank.

Within-movie Retrieval: In order to evaluate the task of within-movie retrieval,

Table 2.3: Training splits for cross-movie retrieval (left) and within-movie retrieval (right). For within-movie retrieval, we restrict the dataset to movies which have at least 5 video clips in total.

	Cross-Movie				Within-Movie			
	Train	Val	Test	Total	Train	Val	Test	Total
#Movies	2,551	358	696	3,605	2,469	341	671	3,481
#Video clips	24,047	3,348	6,581	33,976	23,963	3,315	6,581	33,859

we remove all movies that contain less than 5 video clips from the dataset. For each query text, the possible gallery set consists only of the videos in the same movie as the query. In this setting the retrieval metrics are calculated separately for each movie and then averaged over all movies. We report R@1, mean and median rank.

2.5.2 Baselines

The **E2EWS** (End-to-end Weakly Supervised) is a cross-modal retrieval model trained by [Miech et al. 2020] using weak supervision from a large-scale corpus of (100 million) instructional videos (using speech content as the supervisory signal). We use the video and text encoders without any form of fine-tuning on Condensed Movies, to demonstrate the widely different domain of our dataset.

The **MoEE** (Mixture of Embedded Experts) model proposed by [Miech et al. 2018] comprises a multi-modal video model in combination with a system of context gates that learn to fuse together different pretrained experts.

The **CE** model [Y. Liu et al. 2019] similarly learns a cross-modal embedding by fusing together a collection of pretrained experts to form a video encoder, albeit with pairwise relation network sub-architectures. It represents the state-of-the-art on several retrieval benchmarks.

Context Boosting Module: Finally, we report results with the addition of our Context Boosting module to both MoEE and CE. We use the fact that the video clips in our dataset are ordered by the time they appear in the movie, and encode previous and future 'key scenes' in the movie along with every video clip using the CBM. An ablation on the number of clips encoded for context can be found in the supplementary material.

We finally show the results of an ablation study demonstrating the importance

of different experts for this task on the task of cross-movie retrieval.

In the next sections, we first describe the implementation details of our models and then discuss quantitative and qualitative results.

2.5.3 Implementation Details

Expert Features: In order to capture the rich content of a video, we draw on existing powerful representations for a number of different semantic tasks. These are first extracted at a frame-level, then aggregated by taking the mean to produce a single feature vector per modality per video.

RGB object frame-level embeddings of the visual data are generated with an SENet-154 model [J. Hu et al. 2019] pretrained on ImageNet for the task of image classification. Frames are extracted at 25 fps, where each frame is resized to 224×224 pixels. Features collected have a dimensionality of 2048.

Motion embeddings are generated using the I3D inception model [Carreira and Zisserman 2017] trained on Kinetics [Kay et al. 2017], following the procedure described by [Carreira and Zisserman 2017].

Face embeddings for each face track are extracted in three stages: (1) Each frame is passed through a dual shot face detector [Jian Li et al. 2019] (trained on the Wider Face dataset [S. Yang et al. 2016]) to extract bounding boxes. (2) Each box is then passed through an SENet50 [J. Hu et al. 2019] trained on the VGGFace2 dataset [Cao et al. 2018] for the task of face verification, to extract a facial feature embedding, which is L2 normalised. (3) A simple tracker is used to connect the bounding boxes temporally within shots into face tracks. Finally the embeddings for each bounding box within a track are average pooled into a single embedding per face track, which is again L2 normalised. The tracker uses a weighted combination of intersection over union and feature similarity (cosine similarity) to link bounding boxes in consecutive frames.

Subtitles are encoded using BERT embeddings [J. Devlin et al. 2019] averaged across all words.

Scene features of 2208 dimensions are encoded using a DenseNet161 model [Iandola et al. 2014] pretrained on the Places365 dataset [B. Zhou et al. 2017], applied to 224×224 pixel centre crops of frames extracted at 1fps.

Table 2.4: Cross-movie text-video retrieval results on the CMD *test* set of 6,581 video clips, with varying levels of context. Random weights refers to the MoEE model architecture with random initialization. We report Recall@k (higher is better), Median rank and Mean rank (lower is better).

Method	Recall@1	Recall@5	Recall@10	Median Rank	Mean Rank
Random weights	0.0	0.1	0.2	3209	3243.5
E2EWS [Miech et al. 2020]	0.7	2.2	3.7	1130	1705.5
CE [Y. Liu et al. 2019]	2.3	7.4	11.8	190	570.0
MoEE [Miech et al. 2018]	4.7	14.9	22.1	65	285.3
CE + CBM (ours)	3.6	12.0	18.2	103	474.6
MoEE + CBM (ours)	**5.6**	**17.6**	**26.1**	**50**	**243.9**

Table 2.5: Within-Movie Retrieval results on the CMD test set. All movies with less than 5 video clips are removed. Metrics are computed individually for each movie and then averaged (m-MdR and m-MnR refers to the mean of the median and mean rank obtained for each movie respectively). R@1 denotes recall@1. We show the results of 3 different variations of embeddings obtained from the character module.

Method	m-R@1	m-MdR	m-MnR
Random weights	11.1	5.32	5.32
MoEE	38.9	2.20	2.82
MoEE + Character Module [one-hot]	45.5	1.91	2.60
MoEE + Character Module [track-len norm]	46.2	1.88	2.53
MoEE + Character Module **[track-freq norm]**	**47.2**	**1.85**	**2.49**

Descriptions are encoded using BERT embeddings, providing contextual word-level features of dimensions $W \times 1024$ where W is the number of tokens. These are concatenated and fed to a NetVLAD layer to produce a feature vector of length of 1024 times the number of NetVLAD clusters for variable length word tokens.

Training details and hyperparameters: All baselines and CBM are implemented with the PyTorch [Paszke et al. 2017] framework, and the optimizer used is Adam [Kingma and Ba 2014], using a learning rate of 0.001, and a batch size of 32. The margin hyperparameter m for the bidirectional ranking loss is set to a value of 0.121, the common projection dimension D to 512, and the description NetVLAD clusters to 10. For CBM, we select the number of past and future context videos to be N=3, ablations for hyperparameters and using different amounts of context are given in the supplementary material. Training is stopped when the validation loss stops decreasing.

Table 2.6: Expert ablations. The value of different experts in combination with a baseline for text-video retrieval (left) and (right) their cumulative effect (here Prev. denotes the experts used in the previous row). R@k: recall@k, MedR: median rank, MeanR: mean rank

Experts	R@1	R@5	R@10	MedR	MeanR	Experts	R@1	R@5	R@10	MedR	MeanR
Scene	0.8	3.2	5.9	329	776.3	Scene	0.8	3.2	5.9	329	776
Scene+Face	3.7	12.7	19.7	100	443.1	Prev.+Face	3.7	12.7	19.7	100	443.1
Scene+Obj	1.0	4.6	8.0	237	607.8	Prev.+Obj	3.9	13.1	20.5	79	245.5
Scene+Action	1.9	6.4	10.5	193	575.0	Prev.+Action	4.0	14.0	20.4	78	233.3
Scene+Speech	2.3	8.3	12.4	165	534.7	Prev.+Speech	5.6	17.7	25.7	50	243.9

Figure 2.5: **Qualitative results of the MoEE+CBM model for cross-movie retrieval.** On the left, we provide the input query, and on the right, we show the top 4 video clips retrieved by our model on the CMD *test set*. A single frame for each video clip is shown. The matching clip is highlighted with a green border, while the rest are highlighted in red (best viewed in colour). Note how our model is able to retrieve semantic matches for situations (row 1: male/female on a date), high level abstract concepts (row 2: the words 'stand' and 'truth' are mentioned in the caption and the retrieved samples show a courtroom, men delivering speeches and a policeman's office) and also notions of violence and objects (row 3: scythe).

2.5.4 Results

Results for cross-movie retrieval can be seen in Table 2.4. E2EWS performs poorly, illustrating the domain gap between CMD and generic YouTube videos from HowTo100M. Both the CE and MoEE baselines perform much better than random, demonstrating that story-based retrieval is achievable on this dataset. We show that the Contextual Boost module can be effectively used in conjunction with existing video retrieval architectures, improving performance for both CE and MoEE, with the latter being the best performing model. Results for within-movie retrieval can be seen in Table 2.5. We show that adding in the character module provides a significant boost (almost a 10% increase in Recall@1 compared to the MoEE without the character module), with the best results obtained from normalizing the character embeddings by the track frequency. The value of different experts is assessed in Table 2.6. Since experts such as subtitles and face are missing for many video clips, we show the performance of individual experts combined with the 'scene' expert features, the expert with the lowest performance that is consistently available for all clips (as done by [Y. Liu et al. 2019]). In Table 2.6, right, we show the cumulative effect of adding in the different experts. The highest boosts are obtained from the face features and the speech features, as expected, since we hypothesize that these are crucial for following human-centric storylines. We show qualitative results for our best cross-movie retrieval model (MoEE + CBM) in Fig. 8.2.

2.6 Plot Alignment

A unique aspect of the Condensed Movies Dataset is the story-level captions accompanying the ordered key scenes in the movie. Unlike existing datasets [A. Rohrbach et al. 2017b] that contain low level visual descriptions of the visual content, our semantic captions capture key plot elements. To illustrate the new kinds of capabilities afforded by this aspect, we align the video descriptions to Wikipedia plot summary sentences using Jumping Dynamic Time Warping [Feng et al. 2010] of BERT sentence embeddings. This alignment allows us to place each video clip in the global context of the larger plot of the movie. A qualitative example is

Figure 2.6: A sample Wikipedia movie plot summary (left) aligned with an ordered sample of clips and their descriptions (right). The alignment was achieved using Jumping Dynamic Time Warping [Feng et al. 2010] of sentence-level BERT embeddings, note how the alignment is able to skip a number of peripheral plot sentences.

shown in Fig. 2.6. Future work will incorporate this global context from movie plots to further improve retrieval performance.

2.7 Conclusion

In this work, we introduce a new and challenging *Condensed Movies Dataset* (CMD), containing captioned video clips following succinct and clear storylines in movies. Our dataset consists of long video clips with high level semantic captions, annotated face-tracks, and other movie metadata, and is freely available to the research community. We investigate the task of story-based text retrieval of these clips, and show that modelling past and previous context improves performance. Beside improving retrieval, developing richer models to model longer term temporal context will also allow us to follow the evolution of relationships [Kukleva et al. 2020a] and higher level semantics in movies, exciting avenues for future work.

Acknowledgements

This work is supported by a Google PhD Fellowship, an EPSRC DTA Studentship, and the EPSRC programme grant Seebibyte EP/M013774/1. We are grateful to Samuel Albanie for his help with feature extraction.

2.8 Appendix

2.8.1 Dataset

Character Identity Pipeline

We describe in detail the process of building the character embedding bank mentioned in Sec. 3.1 of the main paper, and state some figures on the number of annotations obtained. We follow a three step *scalable* pipeline to assign character IDs to each of the face-tracks where possible, crucially without any human annotation. First, we use the cast lists obtained for each of the featured movies from IMDb to get a total list of 28,379 actor names. Note we use the names of the *actors* and not characters (the cast lists provide us with the mapping between the two). 200 images are then downloaded from image search engines for each of these names. Faces are detected and face-embeddings extracted for each of the faces in the downloaded images. Second, we automatically remove embeddings corresponding to false positives from each set of downloaded images. We achieve this by clustering each of the face-embeddings in the downloaded images into identity clusters (we use agglomerative clustering [Jain and Dubes 1988] with a cosine distance threshold of 0.76[3] - embeddings that have a lower similarity than this threshold are *not* merged into the same cluster). We make the assumption that the largest cluster of face-embeddings corresponds to the actor ID that was searched for. If the largest cluster is smaller than a certain threshold (the value 30 is used[3]) then we remove the actor ID with the conclusion that too few images were found online (commonly the case for relatively unknown cast/crew members). Finally for the remaining actor IDs, the embeddings in the largest cluster are average pooled and L2 normalised into a single embedding. This process leaves us with 13,671 cast members in the *character embedding bank*. Facetracks are then annotated using the character embedding bank by assigning a character ID when the cosine similarity score between a facetrack embedding and character embedding is above a certain threshold (we use 0.8 as a conservative threshold to prioritize high precision).

The most frequent actors in terms of screen-time automatically labelled by our method can be found in Fig. 2.7

[3] value found empirically using cross-validation on a subset of manually annotated samples

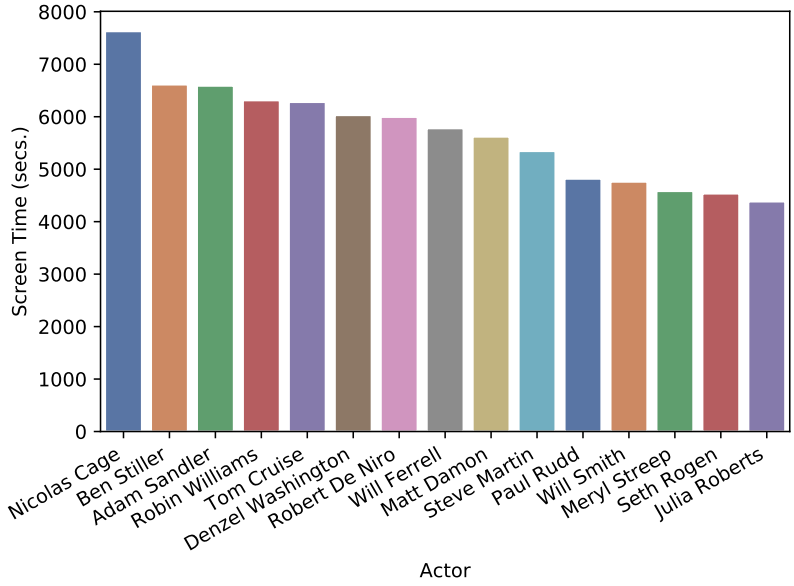


Figure 2.7: Screen Time of the top 15 most frequent actors recognised by our character identity pipeline, computed as the total duration of face-tracks.

Aligning Plots to Captions

Aligning video to text has been investigated by a long history of work [Bojanowski et al. 2015], in particular plot summaries/synopses with films or TV shows [Xiong et al. 2019]. These works assume complete and ordered data streams of both video and text, enabling use of the Dynamic Time Warping (DTW) algorithm introduced in [Sakoe and Chiba 1978], significantly constraining the problem. The text data for CMD however is Wikipedia plot summaries which do not contain descriptions of every scene in the movie, but rather succinct sentences describing the important events in the film. Further, the video in our case is not the full-length movie but instead key scenes sparsely sampled from the full video. This means that many of the assumptions of DTW do not hold true.

Instead, we assume that each video clip V should be matched with one plot synopsis sentence S. Since for our data $|V| < |S|$, some plot sentences are not matched with any video clip, but every video clip does have a matching sentence. This is setting is handled by Jumping Dynamic Time Warping (JDTW) [Feng et al. 2010]. We randomly sample 100 movies and manually align the movie clips from CMD to their Wikipedia plot summaries.

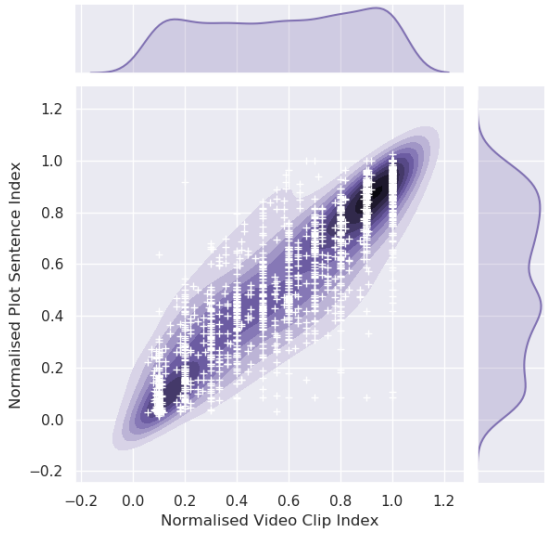


Figure 2.8: **Plot Coverage: Cumulative distribution of aligned video clips and plot synopsis sentences for 100 randomly sampled movies. Normalised position index indicates their actual index normalised by the total number of video clips / plot synopsis sentences for the movie.**

2.8.2 Experiments

Character Module

The character module, described in Section 4 of the main paper, uses automatically annotated facetracks in the video and actor names in the text to produce a single similarity score. An overview of the character module can be found in Fig. 2.9.

Context Ablations

We provide ablations for the best peforming model MoEE + *Contextual Boost Module* (CBM) using a variable number of context videos from the past and future, found in Table 2.7. The results show CBM's general robustness to the amount of context. Past context clips generally outperform those from the future, which is expected due to the causal nature of the story.

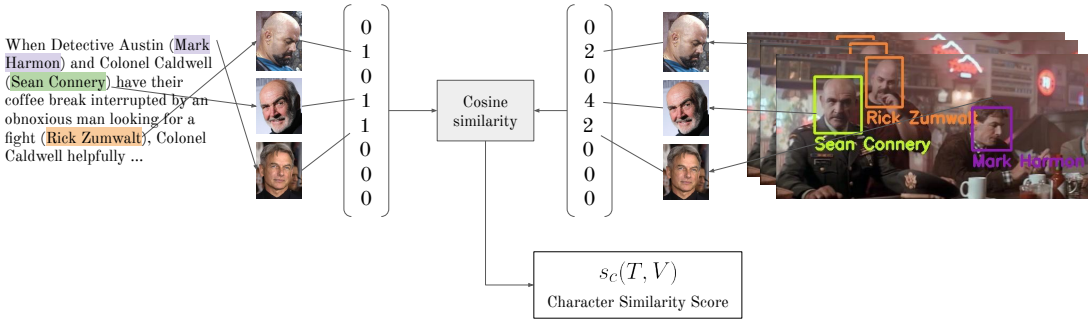


Figure 2.9: **Visual Representation of our Character Module.** We show how our character module matches actor names in the caption (left) to actors identified from the video clip (right) using our character embeddings banks. In this example, the video identities are represented by a vector x, where each element x_i is the number of facetracks for identity i, and the caption identities are represented by a binary vector y, where y_i is 1 if identity i is present in the caption and 0 otherwise.

Table 2.7: Context ablations of the best performing model (MoEE + CBM) on the CMD dataset. Where Px Fy denotes x past clips and y future clips used as input to the CBM per target video.

	Text $\Longrightarrow$ Video				
Context	R@1	R@5	R@10	MdR	MnR
P1	5.4	17.6	25.7	51	260.7
P2	5.0	16.1	24.5	53	250.3
P3	5.6	17.1	25.7	50	253.8
F1	4.5	15.3	23.1	58	258.7
F2	5.1	17.0	25.5	49	248.1
F3	5.4	17.1	25.9	50	247.0
P1F1	5.0	16.4	25.3	51	249.
P3F3	5.6	17.6	26.1	50	243.9

Chapter 3

Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval

The paper has been accepted for publication at the International Conference on Computer Vision (ICCV), 2021.

Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval

Max Bain[1] Arsha Nagrani[1] Gül Varol[1,2] Andrew Zisserman[1]

[1] Visual Geometry Group, University of Oxford

[2] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS

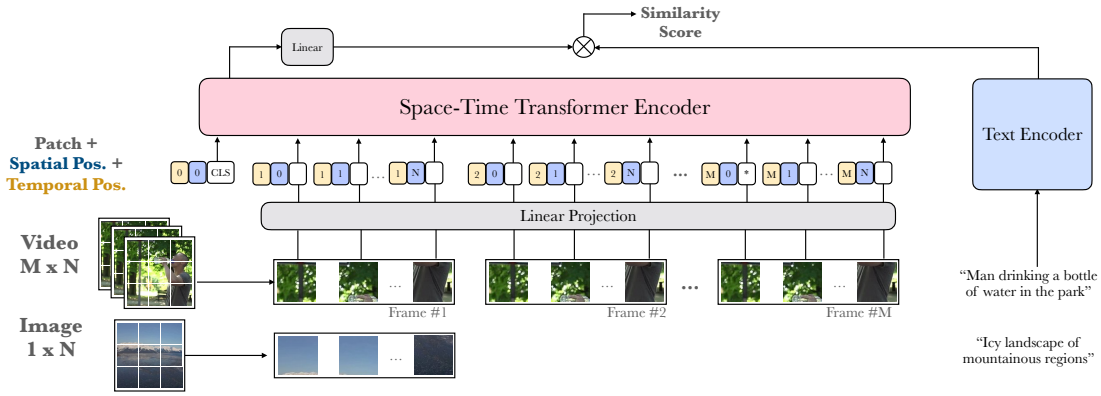


Figure 3.1: **Joint Image and Video Training:** Our dual encoding model consists of a visual encoder for images and video and a text encoder for captions. Unlike 2D or 3D CNNs, our space-time transformer encoder allows us to train flexibly on both images and videos with captions jointly, by treating an image as a single frame video.

Abstract

Our objective in this work is video-text retrieval – in particular a joint embedding that enables efficient text-to-video retrieval. The challenges in this area include the design of the visual architecture and the nature of the training data, in that the available large scale video-text training datasets, such as HowTo100M, are noisy and hence competitive performance is achieved only at scale through large amounts of compute.

We address both these challenges in this paper. We propose an end-to-end trainable model that is designed to take advantage of both large-scale `image` and video captioning datasets. Our model is an adaptation and extension of the recent ViT and Timesformer architectures, and consists of attention in both space and time. The model is flexible and can be trained on both image and video text datasets, either independently or in conjunction. It is trained with a curriculum learning schedule that begins by treating images as 'frozen' snapshots of video, and then gradually learns to attend to increasing temporal context when trained on video datasets. We also provide a new video-text pretraining dataset WebVid-2M, comprised of over two million videos with weak captions scraped from the internet. Despite training on datasets that are an order of magnitude smaller, we show that this approach yields state-of-the-art results on standard downstream video-retrieval benchmarks including MSR-VTT, MSVD, DiDeMo and LSMDC.

3.1 Introduction

Joint visual-text models have become increasingly popular as they enable a wide suite of downstream tasks, including text-to-visual retrieval [T.-Y. Lin et al. 2014; Liwei Wang et al. 2016; Miech et al. 2018; Y. Liu et al. 2019], visual captioning [Vinyals et al. 2016; You et al. 2016; Krishna et al. 2017a], and visual question and answering [Antol et al. 2015; Lei et al. 2018]. Their rapid development is due to the usual improvements on three fronts: new neural network architectures (e.g. transformers [Vaswani et al. 2017] for both text and visual inputs); new large-scale datasets; and new loss functions that are, for example, able to handle label noise [Miech et al. 2020]. However, their development mostly proceeds on two independent tracks: one for *images*, with its own architectures, training datasets and benchmarks [T.-Y. Lin et al. 2014; Krishna et al. 2017b; Sharma et al. 2018a]; and the other for *videos* with a similar separation of training datasets and benchmarks [J. Xu et al. 2016; Anne Hendricks et al. 2017; Krishna et al. 2017a; A. Rohrbach et al. 2017b; L. Zhou et al. 2018a; Bain et al. 2020a]. The only common link between the two is that often video networks are initialized by pre-training image networks on image datasets [Carreira and Zisserman 2017; Bertasius et al. 2021]. This separation of effort is suboptimal given the overlap in information that

images and video convey over multiple tasks. For example, although classifying some human actions requires the temporal ordering of video frames, many actions can be classified from just their distribution over frames or even from a single frame [Sevilla-Lara et al. 2021].

In this paper we take a step towards unifying these two tracks, by proposing a dual encoder architecture which utilises the flexibility of a transformer visual encoder to train from images-with-captions, from video clips-with-captions, or from both (Fig. 3.1). We do this by treating images as a special case of videos that are 'frozen in time'. Using a transformer-based architecture allows us to train with variable-length sequences, treating an image as if it was a single frame video, unlike in standard 3D CNNs [Carreira and Zisserman 2017; Hara et al. 2018; S. Xie et al. 2018] where to train on images jointly with videos one must incur the cost of actually generating a static video. Furthermore, unlike many recent methods [Miech et al. 2018; Y. Liu et al. 2019; Gabeur et al. 2020] for video-text dual encoding, we do not use a set of 'expert networks' that are pre-trained on external image datasets and then fixed, but instead train the model end-to-end.

This end-to-end training is facilitated by scraping the web for a new large-scale video-text captioning dataset of over two million video alt-text pairs (WebVid-2M). We also take advantage of large-scale image captioning datasets such as Conceptual Captions [Sharma et al. 2018a].

We make the following contributions: (i) we propose a new end-to-end model for video retrieval that does *not* rely on 'expert' features, but instead, inspired by [Bertasius et al. 2021] employs a transformer architecture with a modified divided space-time attention applied directly to pixels; (ii) because our architecture can gracefully handle inputs of different lengths, it is versatile and can be flexibly trained on both video and image datasets (by treating images as a single-frame video). We build on this flexibility by designing a curriculum learning schedule that begins with images and then gradually learns to attend to increasing temporal context when trained on video datasets through temporal embedding interpolation. We show that this increases efficiency, allowing us to train models with far less GPU time; (iii) we introduce a new dataset called WebVid-2M, consisting of 2.5M video-text pairs scraped from the web; and finally (iv) we achieve state-of-the-art performance by only using the video modality on MSR-VTT [J. Xu et al.

2016], MSVD [D. Chen and Dolan 2011], DiDeMo [Anne Hendricks et al. 2017] and LSMDC [A. Rohrbach et al. 2017b] – outperforming works that use pre-extracted experts from multiple modalities, as well as those that are pretrained on the noisy HowTo100M, which is 20x larger than our dataset in the number of video-text pairs.

3.2 Related Works

Pretraining for video-text retrieval. Given that most video-text retrieval datasets tend to be small-scale, the dominant paradigm for video retrieval has been to use a combination of pre-extracted features from 'expert' models, including models trained for various diverse tasks and on multiple modalities such as face, scene and object recognition, action classification and sound classification. MoEE [Miech et al. 2018], CE [Y. Liu et al. 2019], MMT [Gabeur et al. 2020] and concurrent work HiT [Song Liu et al. 2021] all follow this paradigm, with the overall similarity for a video-text pair obtained as a weighted sum of each expert's similarity with the text.

However, since the release of the HowTo100M dataset [Miech et al. 2019], a large-scale instructional video dataset, there has been a flurry of works leveraging large-scale pretraining to improve video-text representations for tasks such as video question-answering [Seo et al. 2021], text-video retrieval [Patrick et al. 2020] and video captioning [L. Zhou et al. 2018c]. Although semantically rich and diverse, text supervision from instructional videos is extremely noisy, and hence incurs a large computational cost, as scale is required for competitive results. A few approaches have been proposed to combat the noise – e.g. using loss functions such as MIL-NCE [Miech et al. 2020] or using the raw audio [Alayrac et al. 2020; Rouditchenko et al. 2020] directly to increase robustness. Given the large size of existing image-captioning datasets, some have naturally tried to overcome the lack of video-caption training data with joint image-text pretraining (such as in MoEE [Miech et al. 2018] and ClipBERT [Lei et al. 2021]). MoEE [Miech et al. 2018] trains on images jointly by feeding in zeros to all expert streams that require videos, such as the motion and audio features, while ClipBERT [Lei et al. 2021] restricts their feature extractors to 2D CNNs. Instead we propose an elegant

transformer-based encoder that works well with either images or videos and can be trained effectively on both.

Similar to our work, although only suitable for images is CLIP [Radford et al. 2021], which learns an effective joint image-text representation from millions of text-image pairs scraped from the internet using contrastive loss.

End-to-end video representation learning. A large number of architectural developments have been driven by action recognition on datasets such as Kinetics [Kay et al. 2017] where manual labelling has been relatively easier than obtaining textual descriptions for datasets. For a long time this space was dominated by spatio-temporal CNNs such as I3D [Carreira and Zisserman 2017], 3D ResNets [Hara et al. 2018], S3D [S. Xie et al. 2018] or 'R(2+1)D' CNNs [D. Tran et al. 2018]. Here, images are used simply to initialise video models, through inflation [Carreira and Zisserman 2017]. Multigrid scheduling has been proposed for efficient training [C.-Y. Wu et al. 2020].

Transformers for vision. A number of works use self-attention for images, either in combination with convolutions [Xiaolong Wang et al. 2018; Vaswani et al. 2017; Han Hu et al. 2018; Carion et al. 2020] or even replacing them entirely.

Works that use only self-attention blocks tend to apply them at an individual pixel level [Parmar et al. 2018; Ramachandran et al. 2019; Cordonnier et al. 2019], often requiring tricks to ensure computational tractability, including restricting the scope of self-attention to a local neighbourhood [Ramachandran et al. 2019], adding global self-attention on heavily downsized versions, or sparse key-value sampling [Child et al. 2019]. To increase efficiency, ViT [Dosovitskiy et al. 2021] decompose images into a sequence of patches and then feeds linear embeddings of these patches as inputs to a transformer, effectively adding a single convolutional layer to the image at the start. This idea has been extended in DeiT [Touvron et al. 2020]. For video, previous works also employ self-attention blocks together with CNN layers, for action recognition [Girdhar et al. 2017] and video classification [Y. Chen et al. 2018].

In contrast, our architecture consists entirely of self-attention units and is heavily inspired by ViT [Dosovitskiy et al. 2021] and particularly the Timesformer [Bertasius et al. 2021], which uses divided space and time attention. Unlike these works,

we use expandable temporal embeddings to allow flexible training of variable-length videos and images both jointly and separately. We are unaware of any previous works that use self-attention to train on both images and videos in the same model.

3.3 Method

In this section, we describe our transformer-based spatio-temporal model architecture (Section 3.3.1), and our training strategy (Section 3.3.2).

3.3.1 Model Architecture

Input. The visual encoder takes as input an image or video clip $X \in \mathrm{R}^{M \times 3 \times H \times W}$ consisting of M frames of resolution $H \times W$, where $M = 1$ for images. The text encoder takes as input a tokenised sequence of words.

Spatio-temporal patches. Following the protocol in ViT and Timesformer [Bertasius et al. 2021], the input video clip is divided into $M \times N$ non-overlapping spatio-temporal patches of size $P \times P$, where $N = HW/P^2$.

Transformer input. The patches $\boldsymbol{x} \in \mathrm{R}^{M \times N \times 3 \times P \times P}$ are fed through a 2D convolutional layer and the output is flattened, forming a sequence of embeddings $\boldsymbol{z} \in \mathrm{R}^{MN \times D}$ for input to the transformer, where D depends of the number of kernels in the convolutional layer.

Learned temporal and spatial positional embeddings, $\boldsymbol{E}^s \in \mathrm{R}^{N \times D}$, $\boldsymbol{E}^t \in \mathrm{R}^{M \times D}$ are added to each input token:

$$\boldsymbol{z}_{p,m}^{(0)} = \boldsymbol{z}_{p,m} + \boldsymbol{E}_p^s + \boldsymbol{E}_m^t, \tag{3.1}$$

such that all patches within a given frame m (but different spatial locations) are given the same temporal positional embedding E_m^t, and all patches in the same spatial location (but different frames) are given the same spatial positional embedding E_p^s. Thus enabling the model to ascertain the temporal and spatial position of patches.

In addition, a learned [CLS] token [J. Devlin et al. 2019] is concatenated to the beginning of the sequence, which is used to produce the final visual embedding output embedding of the transformer.

Space-time self-attention blocks. The video sequence is fed into a stack of space-time transformer blocks. We make a minor modification to the Divided Space-Time attention introduced by [Bertasius et al. 2021], by replacing the residual connection between the block input and the temporal attention output with a residual connection between the block input and the spatial attention output (see Section 3.7.3 of the Appendix for details). Each block sequentially performs temporal self-attention and then spatial self-attention on the output of previous block. The video clip embedding is obtained from the [CLS] token of the final block.

Text encoding. The text encoder architecture is a multi-layer bidirectional transformer encoder, which has shown great success in natural language processing tasks [J. Devlin et al. 2019]. For the final text encoding, we use the [CLS] token output of the final layer.

Projection to common text-video space. Both text and video encodings are projected to a common dimension via single linear layers. We compute the simliarity between text and video by performing the dot product between the two projected embeddings.

Efficiency. Our model has independent dual encoder pathways (such as in MIL-NCE [Miech et al. 2020] and MMV networks [Alayrac et al. 2020]), requiring only the dot product between the video and text embeddings. This ensures retrieval inference is of trivial cost since it is indexable, i.e. it allows application of fast approximate nearest neighbour search, and is scalable to very large scale retrieval at inference time. Given t text queries and v videos in a target gallery, our retrieval complexity is $O(t+v)$. In contrast, ClipBERT [Lei et al. 2021] which inputs both text and video as input to a single encoder, has retrieval complexity $O(tv)$ since every text-video combination must be inputted to the model. Other expert-based retrieval methods such as MoEE [Miech et al. 2018], CE [Y. Liu et al. 2019] and MMT [Gabeur et al. 2020] also contain a dual encoder pathway, however they still require query-conditioned weights to compute the similarity scores for each expert,

while our model does not.

3.3.2 Training Strategy

Loss. We employ [A. Zhai and H.-Y. Wu 2019] in a retrieval setting, where matching text-video pairs in the batch are treated as positives, and all other pairwise combinations in the batch are treated as negatives. We minimise the sum of two losses, video-to-text and text-to-video:

$$L_{v2t} = -\frac{1}{B}\sum_{i}^{B}\log\frac{\exp(x_i^\top y_i/\sigma)}{\sum_{j=1}^{B}\exp(x_i^\top y_j/\sigma)} \tag{3.2}$$

$$L_{t2v} = -\frac{1}{B}\sum_{i}^{B}\log\frac{\exp(y_i^\top x_i/\sigma)}{\sum_{j=1}^{B}\exp(y_i^\top x_j/\sigma)} \tag{3.3}$$

where x_i and y_j are the normalized embeddings of i-th video and the j-th text respectively in a batch of size B and σ is the temperature.

Joint image-video training. In this work, we train jointly on both image-text pairs as well as video-text pairs, taking advantage of both for larger-scale pretraining. Our joint training strategy involves alternating batches between the image and video datasets. Since the attention mechanism scales with the square of input frames $O(M^2)$, the alternate batch training allows the image batches $(M = 1)$ to be far greater in size.

Weight initialisation and pretraining. Following [Bertasius et al. 2021], we initialise the spatial attention weights in the space-time transformer model with ViT [Dosovitskiy et al. 2021] weights trained on ImageNet-21k, and initialise the temporal attention weights to zero. The residual connections mean that under these initialisation settings, the model is at first equivalent to ViT over each input frame – thereby allowing the model to learn to attend to time gradually as training progresses. Since transformer architectures have demonstrated most of their success from large-scale pretraining, we utilise two large-scale text-image/video datasets with a joint training strategy, resulting in large improvements in performance.

Temporal curriculum learning. The space-time transformer architecture allows a variable length input sequence and therefore a variable number of input

video frames. If the model has only trained on videos up to length m however, then the temporal positional embedding $\boldsymbol{E}^t$ will only be learned up to $\boldsymbol{E}^t_{:m}$. Therefore, applying the model to input video of sequences up to length M will result the addition of $\boldsymbol{E}^t_{m:M}$, which would not yet be learned.

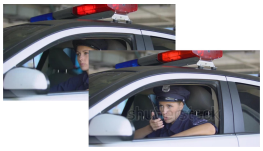
Two temporal expansion methods are investigated: *interpolation* and *zero-padding*. Zeros can be filled in, $\mathbf{0} \rightarrow \boldsymbol{E}^t_{m:M}$, allowing the model to learn the additional temporal positions from scratch during training. Alternatively, interpolation could be used to upsample the temporal embeddings in the temporal dimension, $\boldsymbol{E}^t_{:m} \rightarrow \boldsymbol{E}^t_{:M}$. We investigate two methods of interpolation: nearest neighbour and bilinear. The effects of these different initialisations can be found in the Appendix, Section 3.7.3.

We employ this expansion strategy in order to perform curriculum learning in the number of input frames. Initially training on fewer frames has drastic savings in computation, whilst having comparable or even better performance (see Section 3.4.5).

Frame sampling. Given a video containing L frames, we subdivide it into M equal segments where M is the desired number of frames for the video encoder. During training, we sample a single frame uniformly from each segment (in a similar manner to TSN [L. Wang et al. 2019] and GST [C. Luo and Yuille 2019]). At test time, we sample the i^{th} frame in every segment, to get a video embedding v_i. The values for i are determine using a stride S, resulting in an array of video embeddings $\boldsymbol{v} = [v_0, v_S, v_{2S}, v_M]$. The mean of these video embeddings is used as the final embedding for the video.

"Lonely beautiful woman sitting on the tent looking outside. wind on the hair and camping on the beach near the colors of water and shore. freedom and alternative tiny house for traveler lady drinking"

"Billiards, concentrated young woman playing in club"



"Female cop talking on walkie-talkie, responding emergency call, crime prevention"

"Get anchor for departure safari dive boat scuba diving maldives"

Figure 3.2: **Example video-caption pairs from the WebVid2M dataset:** Note the different captioning styles: from left to right, captions can be (i) long, slightly poetic, with disjoint sentences and phrases, (ii) succint and to the point, (iii) have a less defined sentence structure with keywords appended to the end, (iv) mention specific places ('maldives'). We show two randomly sampled frames for each video.

3.4 Experiments

We first describe the pretraining datasets including our WebVid-2M video-text dataset (Section 3.4.1), followed by the downstream datasets used for the evaluations in our experiments (Section 3.4.2). We then describe implementation details of our model (Section 3.4.3). Next, we ablate various training components on the MSR-VTT dataset, in particular the effects of pretraining and our space-time attention modification (Section 4.4.4), and our proposed curriculum strategy (Section 3.4.5). Then, we compare to the state of the art on four benchmarks: MSR-VTT, MSVD, DiDeMo and LSMDC (Section 3.4.6).

3.4.1 Pretraining Datasets

We jointly pretrain our model on image and video data.

Video pretraining: The WebVid-2M Dataset. We scrape the web for a new dataset of videos with textual description annotations, called WebVid-2M. Our dataset consists of 2.5M video-text pairs, which is an order of magnitude larger than existing video captioning datasets (see Table 3.1).

The data was scraped from the web following a similar procedure to Google Con-

ceptual Captions [Sharma et al. 2018a] (CC3M). We note that more than 10% of CC3M images are in fact thumbnails from videos, which motivates us to use such video sources to scrape a total of 2.5M text-video pairs. The use of data collected for this study is authorised via the Intellectual Property Office's Exceptions to Copyright for Non-Commercial Research and Private Study[1]. We are currently performing further analysis of the dataset on its diversity and fairness.

Figure 3.2 provides sample video-caption pairs. There are a variety of different styles used in caption creation, as can be seen from Figure 3.2 (left to right) where the first video has a longer, poetic description compared to the succinct description for the second video. The third video caption has a less defined sentence structure, with keywords appended to the end, while the fourth video mentions a specific place (maldives). Time-specific information is important for the second and third example, where details such as "talking on walkie-talkie" or "playing billiards" would be missed when looking at certain frames independently.

Table 3.1: **Dataset Statistics:** We train on a new dataset mined from the web called WebVid2M. Our dataset is an order of magnitude larger than existing video-text datasets in the number of videos and captions. HowTo100M (highlighted in blue) is a video dataset with noisy, weakly linked text supervision from ASR.

dataset	domain	#clips	avg dur. (secs)	#sent	time (hrs)
MPII Cook [M. Rohrbach et al. 2012]	cooking	44	600	6K	8
TACos [Regneri et al. 2013]	cooking	7K	360	18K	15.9
DideMo [Anne Hendricks et al. 2017]	flickr	27K	28	41K	87
MSR-VTT [J. Xu et al. 2016]	youtube	10K	15	200K	40
Charades [Sigurdsson et al. 2016a]	home	10K	30	16K	82
LSMDC15 [A. Rohrbach et al. 2017b]	movies	118K	4.8	118K	158
YouCook II [L. Zhou et al. 2018a]	cooking	14K	316	14K	176
ActivityNet [Krishna et al. 2017a]	youtube	100K	180	100K	849
CMD [Bain et al. 2020a]	movies	34K	132	34K	1.3K
WebVid-2M	open	**2.5M**	18	**2.5M**	**13K**
HT100M [Miech et al. 2019]	instruction	136M	4	136M	134.5K

We note that our video dataset is 10x smaller than HowTo100M in video duration and over 20x smaller in the number of paired clip-captions (Table 3.1). Our dataset consists of manually generated captions, that are for the most part well formed sentences. In contrast, HowTo100M is generated from continuous narration with incomplete sentences that lack punctuation. The clip-text pairs are obtained from subtitles and may not be temporally aligned with the video they refer to, or indeed

[1] www.gov.uk/guidance/exceptions-to-copyright/

may not refer to the video at all [Miech et al. 2019]. Our captions, on the other hand, are aligned with the video and describe visual content.

Moreover, there is no noise from imperfect ASR transcription and grammatical errors as is the case for HowTo100M. Our dataset also has longer captions on average (12 vs 4 words for HowTo) which are more diverse (Measure of Textual Lexical Diversity, MTLD [McCarthy and Jarvis 2010] = 203 vs 13.5).

Image pretraining: Google Conceptual Captions [Sharma et al. 2018a]. This dataset consists of about 3.3M image and description pairs. Unlike the curated style of COCO images, Conceptual Captions (CC3M) images and their raw descriptions are harvested from the web, and therefore represent a wider variety of styles. The raw descriptions are harvested from the Alt-text HTML attribute associated with web images.

3.4.2 Downstream Datasets

We now describe the downstream text-video datasets that our model is evaluated on.

MSR-VTT [J. Xu et al. 2016] contains 10K YouTube videos with 200K descriptions. Following other works [Y. Liu et al. 2019], we train on 9K train+val videos and report results on the 1K-A test set.

MSVD [D. Chen and Dolan 2011] consists of 80K English descriptions for 1,970 videos from YouTube, with each video containing 40 sentences each. We use the standard split of 1200, 100, and 670 videos for training, validation, and testing [Patrick et al. 2020; Y. Liu et al. 2019].

DiDeMo [Anne Hendricks et al. 2017] contains 10K Flickr videos annotated with 40K sentences. Following [Lei et al. 2021; Y. Liu et al. 2019], we evaluate paragraph-to-video retrieval, where all sentence descriptions for a video are concatenated into a single query. Since this dataset comes with localisation annotations (ground truth proposals), we report results with ground truth proposals (where only the localised moments in the video are concatenated and used in the retrieval set as done by [Lei et al. 2021]) as well as without (as done by [Y. Liu et al. 2019]).

LSMDC [**A. Rohrbach et al. 2015a**] consists of 118,081 video clips sourced from 202 movies. The validation set contains 7,408 clips and evaluation is done on a test set of 1,000 videos from movies disjoint from the train and val sets. This follows the protocol outlined in [A. Rohrbach et al. 2017b].

ActivityNet Captions [**Krishna et al. 2017a**] contains 20K YouTube videos focused on actions, annotated with 100K sentences. The training set consists of 10K videos, and we use the 'val1' set of 4.9K videos to report results. At test time we use paragraph-to-video retrieval as is standard protocol set by other works, where the segment descriptions are concatenated to give a video-level description.

Flickr30K [**Young et al. 2014**]. We also evaluate on a text-to-image retrieval benchmark to demonstrate the versatility of our model in that it can be used to achieve competitive performance in image settings as well as state-of-the art in video retrieval. The Flickr30K dataset contains 31,783 images with 5 captions per image. We follow the standard protocol of 1,000 images for validation, 1,000 images for testing and the remaining for training.

For downstream datasets with separate `val` and `test` splits, we train all models for 75 epochs and use the epoch with the lowest validation loss for reporting test results. For downstream datasets without a `val` set we report results at 50 epochs.

3.4.3 Implementation Details

All experiments are conducted with PyTorch [Paszke et al. 2019]. Optimization is performed with Adam, using a learning rate of 1×10^{-5}, we use batch sizes of 16, 24, and 96 for 8, 4, and 1-frame inputs respectively. The temperature hyperparameter σ for the loss defined in Eq. 3.2 & 3.3 is set to 0.05. The default pretraining is WebVid-2M and CC3M.

For the visual encoder, all models have the following: $|\ell| = 12$ attention blocks, patch size $P = 16$, sequence dimension $D = 768$, 12 heads and takes 4-frames as downstream input.

The text encoder of all models, unless specified otherwise, is instantiated as DistilBERT base-uncased [Sanh et al. 2019] pretrained on English Wikipedia and Toronto Book Corpus. The dimensionality of the common text-video space is set

Table 3.2: **Pretraining sources:** The effect of different pretraining sources. We use 4 frames per video in both pretraining and finetuning. Pretraining is performed for 1 full epoch only. Results are presented on the 1K-A MSR-VTT test set for text-video retrieval. **R@k:** Recall@K. **MedR:** Median Rank

Pre-training	**#pairs**	**R@1**	**R@10**	**MedR**
-	-	5.6	22.3	55
ImageNet		15.2	54.4	9.0
HowTo-17M subset	17.1M	24.1	63.9	5.0
CC3M	3.0M	24.5	62.7	5.0
WebVid2M	2.5M	26.0	64.9	5.0
CC3M + WebVid2M	5.5M	**27.3**	**68.1**	**4.0**

to 256. For visual augmentation, we randomly crop and horizontally flip during training, and center crop the maximal square crop at test time. All videos are resized to 224×224 as input. At test-time we compute clip-embeddings for the video with a stride of 2 seconds. For paragraph-retrieval settings, we employ text augmentation during training by randomly sampling and concatenating a variable number of corresponding captions per video.

Finetuning time. A large motivation for using pre-extracted expert models for video retrieval is to save computational cost. Finetuning our 4-frame model for 50 epochs on MSR-VTT takes 10 hours on 2 Quadro RTX 6000k GPUs (with 24GB RAM each), which is similar to other works using *pre-extracted expert features* [Patrick et al. 2020]. This shows that our model is lightweight and can be finetuned end-to-end on the downstream video datasets quickly with sufficient pretraining (which is of one-time cost).

3.4.4 Ablation Study

In this section we study the effect of different pretraining strategies. In the Section 3.7.3 of the Appendix, we provide architectural ablations on different temporal expansion methods, different visual backbones, different text backbones and the improvement when using our modified space-time attention block.

Effect of pretraining. We compare performance on MSR-VTT with our model (i) trained from scratch, (ii) initialised with ImageNet weights and then finetuned, as well as (iii) initalised with ImageNet, and then pretrained on a number of different visual-text datasets before finetuning. For the video data, 4 frames are

sampled at both pretraining and finetuning. Results on the MSR-VTT 1KA test set are shown in Table 3.2. For HowTo100M, we pretrain on a random 17M subset due to computational constraints (the largest subset we could obtain at the time of writing) totalling 19K hours. To generate text-video pairs, we sample 5 contiguous speech-video pairs and concatenate them to form a longer video. This allows for robustness to the noisy alignment of speech and vision. We find that training on CC3M alone does reasonably well, outperforming the HowTo-17M subset. This demonstrates the benefit of our flexible encoder that can be cheaply trained on images and easily applied to videos. Training on WebVid2M also outperforms training on the HowTo17M subset, despite being much smaller, confirming that the HowTo100M dataset is noisy. The best performance is achieved by jointly training on both CC3M and WebVid2M, effectively exploiting image and video data.

3.4.5 Curriculum strategy

Next, we evaluate the ability of our curriculum schedule to gradually learn the temporal dimension of videos by increasing the input number of frames. Table 3.3 summarises the results. Here, we show performance when pretraining on WebVid2M and finetuning on MSR-VTT. We explore two types of expansion in time: at pretraining and at finetuning stages. First, we observe that a single frame is not sufficient to capture the video content (18.8 R@1). Performing the temporal expansion at pretraining stage is better than doing so at finetuning (26.0 vs 24.9 R@1 with 4 frames). Finally, we obtain similar performance (slightly better at R@5) at half the computational cost in GPU hours by employing a curriculum strategy at pretraining (26.6 R@1). For 8 frames, the curriculum is even more useful, as we start training on 1 frame and then move to 4 before finally moving to 8 frames. Here, we obtain similar or better performance than training on 8 frames from the start, with almost a third of the computational cost. This is to be expected, as fewer frames significantly reduces forward pass times and enables larger batch sizes. Note that for a fair comparison, we allow the same number of training iterations for each row in the table. We further analyse our proposed temporal curriculum strategy and its effects on training time and accuracy. Figure 3.3 shows the zero-shot results on MSR-VTT for various checkpoints with and with-

Table 3.3: **Effect of #frames and curriculum learning:** The effect of a different number of input frames at pretraining and finetuning. ⇒ indicates a within-dataset curriculum learning strategy. Results are presented on the 1K-A MSR-VTT test set for text-video retrieval. Pretraining here is done on WebVid2M only, with a total budget of one epoch through the entire dataset. **PTT:** total pretraining time in hours.

PT #frames	FT #frames	R@1	R@10	MedR	PTT (hrs)
1	1	18.8	56.6	7.0	16.2
1	4	24.9	67.1	5.0	16.2
4	4	26.0	64.9	5.0	45.6
1⇒4	4	26.6	65.5	5.0	22.1
8	8	25.4	67.3	4.0	98.0
1⇒4⇒8	8	27.4	67.3	4.0	36.0

out curriculum. It shows that our curriculum method yields a significant training speedup with a gain in accuracy. Shorter frame models are able to pass through more of the dataset in a shorter amount of time, which can lead to significant performance benefits in a constrained setting.

Expansion of temporal embeddings. We experiment with both zero padding and interpolation, and find that our model is robust to the type of temporal expansion strategy. More detailed results are provided in the Appendix, Section 3.7.3.

3.4.6 Comparison to the State of the Art

Results on MSR-VTT can be seen in Table 3.4. We outperform all previous works, including many that pretrain on HowTo100M which is an order of magnitude larger than our pretraining dataset both in the number of hours (135K vs 13K) and in the number of caption-clip pairs (136M vs 5.5M). We also note that we outperform works that extract expert features (CE uses 9 experts, MMT uses 7) including object, motion, face, scene, sound and speech embeddings. We even outperform/perform on par with Support Set [Patrick et al. 2020], which uses expert features from a 34-layer, R(2+1)-D model pretrained on IG65M, concatenated with ImageNet ResNet152 features, after which they add a transformer network and train end-to-end on HowTo100M.

We also report zero-shot results (Table 3.4) with no finetuning on MSR-VTT,

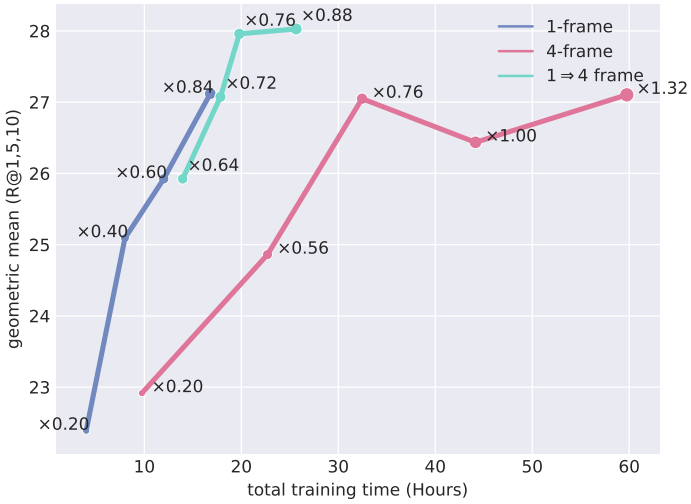


Figure 3.3: Plot showing the zero-shot performance (geometric mean of R@1,5,10) of various models on the MSR-VTT test set against their total training time in hours. ⇒ denotes a curriculum learning strategy. × denotes the multiple of dataset epochs completed.

outperforming both MIL-NCE and Support Set that trains on HowTo100M. This shows that our model is more generalisable, and can be used out of the box, and also perhaps that the domain of WebVid-2M is closer to that of MSR-VTT than HowTo100M. We will release the weights of our models publicly.

For both the zero-shot and finetuned setting we show that the addition of the COCO Captions image dataset further boosts our state-of-the-art MSR-VTT performance, indicating that the model is not yet saturated and additional pretraining dataset will lead to even better downstream performance.

For MSVD [D. Chen and Dolan 2011], we outperform all previous methods (Table 3.5). In particular, we outperform Support Set [Patrick et al. 2020] even though they train on an order of magnitude more data.

Results on DiDeMo can be found in Table 3.6. Note that on this dataset, our zero-shot performance is equivalent to CLIPBERT's results with finetuning, and after we finetune our model on the DiDeMo training set we get an additional 14.2% boost in R@1.

We demonstrate further state-of-the-art results on LSMDC text-to-video retrieval. We outperform all previous methods, except for MMT in Median Rank, which pretrains on HowTo100M, a dataset consisting of over 100M clip-text pairs and

Table 3.4: Comparison to state-of-the-art results on MSR-VTT for text-to-video retrieval, 1k-A split. †**E2E:** Works trained on pixels directly, without using pre-extracted expert features trained for other tasks. **Vis Enc. Init.:** Datasets used for pretraining visual encoders for tasks *other than visual-text retrieval*, eg object classification. **Visual-Text PT:** Visual-text pretraining data. Rows highlighted in blue use additional modalities such as sound and speech from the MSR-VTT test videos. † Object, Motion, Face, Scene, Speech, OCR and Sound classification features.

Method	E2E†	Vis Enc. Init.	Visual-Text PT	#pairs PT	R@1	R@5	R@10	MedR
JSFusion [Y. Yu et al. 2018]	✓	-	-	-	10.2	31.2	43.2	13.0
HT MIL-NCE [Miech et al. 2019]	✓	-	HT100M	136M	14.9	40.2	52.8	9.0
ActBERT [L. Zhu and Y. Yang 2020]	✓	VG	HT100M	136M	16.3	42.8	56.9	10.0
HERO [L. Li et al. 2020]	✓	IN1k,K400	HT100M	136M	16.8	43.4	57.7	-
VidTranslate [Korbar et al. 2020]	✓	IG65M	HT100M	136M	14.7	-	52.8	
NoiseEst. [Amrani et al. 2020]	✗	IN1k,K400	HT100M	136M	17.4	41.6	53.6	8.0
CE [Y. Liu et al. 2019]	✗	Experts†	-		20.9	48.8	62.4	6.0
UniVL [H. Luo et al. 2020]	✗	-	HT100M	136M	21.2	49.6	63.1	6.0
ClipBERT [Lei et al. 2021]	✓	-	COCO,VG	5.6M	22.0	46.8	59.9	6.0
AVLnet [Rouditchenko et al. 2020]	✗	IN1k,K400	HT100M	136M	27.1	55.6	66.6	4.0
MMT [Gabeur et al. 2020]	✗	Experts†	HT100M	136M	26.6	57.1	69.6	4.0
T2VLAD [Xiaohan Wang et al. 2021a]	✗	Experts†	-		29.5	59.0	70.1	4.0
Support Set [Patrick et al. 2020]	✗	IG65M,IN1k	-	-	27.4	56.3	67.7	3.0
Support Set [Patrick et al. 2020]	✗	IG65M,IN1k	HT100M	136M	30.1	58.5	69.3	**3.0**
Ours	✓	IN1k	CC3M	3M	25.5	54.5	66.1	4.0
Ours	✓	IN1k	CC3M,WV2M	5.5M	**31.0**	**59.5**	**70.5**	**3.0**
Ours	✓	IN1k	CC3M,WV2M,COCO	6.1M	**32.5**	**61.5**	**71.2**	**3.0**
Zero-shot								
HT MIL-NCE [Miech et al. 2019]	✓	-	HT100M	136M	7.5	21.2	29.6	38.0
SupportSet [Patrick et al. 2020]		IG65M,IN1k	HT100M	136M	8.7	23.0	31.1	31.0
Ours	✓	IN1k	CC3M,WV2M	5.5M	**23.2**	**44.6**	**56.6**	**7.0**
Ours	✓	IN1k	CC3M,WV2M,COCO	6.1M	**24.7**	**46.9**	**57.2**	**7.0**

contains multiple experts as well as audio modalities. Our model uses visual information alone.

To demonstrate the effectiveness of our model for downstream video and image tasks, we additionally report results on Flickr30K the image retrieval dataset in Table 3.8. Unlike other works [Lee et al. 2018; Hui Chen et al. 2020; Diao et al. 2021] which utilise high resolution regions extracted using a Faster-RCNN detector, our model is single stage and does not require any object detections. We compare to works with a similar number of training image-text pairs, and find that our model

Table 3.5: Text-to-video retrieval results on the MSVD [D. Chen and Dolan 2011] test set.

Method	R@1	R@5	R@10	MedR
VSE [Kiros et al. 2014]	12.3	30.1	42.3	14.0
VSE++ [Faghri et al. 2017]	15.4	39.6	53.0	9.0
Multi. Cues [Mithun et al. 2018]	20.3	47.8	61.1	6.0
CE [Y. Liu et al. 2019]	19.8	49.0	63.8	6.0
Support Set [Patrick et al. 2020]	23.0	52.8	65.8	5.0
Support Set [Patrick et al. 2020] (HowTo PT)	28.4	60.0	72.9	4.0
Ours	**33.7**	**64.7**	**76.3**	**3.0**

Table 3.6: Text-to-video retrieval results on the DiDeMo test set. We show results with and without ground truth proposals (GT prop.) as well as with finetuning and without (zero-shot).

Method	GT prop.	R@1	R@5	R@10	MedR
S2VT [Venugopalan et al. 2014]		11.9	33.6	-	13.0
FSE [B. Zhang et al. 2018]		13.9	36.0	-	11.0
CE [Y. Liu et al. 2019]		16.1	41.1	-	8.3
ClipBERT [Lei et al. 2021]	✓	20.4	44.5	56.7	7.0
Ours		**31.0**	**59.8**	**72.4**	**3.0**
Ours	✓	**34.6**	**65.0**	**74.7**	**3.0**
Zero-shot					
Ours		21.1	46.0	56.2	7.0
Ours	✓	20.2	46.4	58.5	7.0

Table 3.7: Text-to-video retrieval results on the LSMDC test set.

Method	R@1	R@5	R@10	MedR
JSFusion [Y. Yu et al. 2018]	9.1	21.2	34.1	36.0
MEE [Miech et al. 2018]	9.3	25.1	33.4	27.0
CE [Y. Liu et al. 2019]	11.2	26.9	34.8	25.3
MMT (HowTo100M) [Gabeur et al. 2020]	12.9	29.9	40.1	**19.3**
Ours	**15.0**	**30.8**	**40.3**	20.0

is comparable. We also note that training on WebVid2M provides a sizeable boost (5% improvement in R@1). Note that there are other recent text-image works such as UNITER [Y.-C. Chen et al. 2020] and OSCAR [X. Li et al. 2020], however these are trained on almost twice the number of samples. Recent works scale this up even further to billions of samples (ALIGN [C. Jia et al. 2021]).

Table 3.8: Text-to-**image** retrieval results on the Flickr30K test set. ++ indicates additional datasets: COCO Captions, SBU Captions. VisGenObjects denotes Visual Genome object bounding box annotations used to pretrain an FRCNN object feature extractor.

Method	Vis PT. size	R@1	R@5	R@10
SCANM [Lee et al. 2018]	VisGenObj (3.8M)	48.6	77.7	85.2
IMRAM [Hui Chen et al. 2020]	VisGenObj (3.8M)	53.9	79.4	87.2
SGRAF [Diao et al. 2021]	VisGenObj (3.8M)	58.5	83.0	88.8
Ours	CC (3.0M)	54.2	83.2	89.8
Ours	CC,WV-2M (5.5M)	61.0	87.5	92.7

3.5 Extension: Scaling up Further

To investigate the effects of downstream performance on additional pretraining datasets and increased scale, we train models on the following datasets:

WebVid-10M: An extension to our WebVid-2M dataset, we increase the size of the dataset fourfold to 10 million text-video pairs, following the same data collection protocol. The captions and video url's can also be found at https://m-bain.github.io/webvid-dataset/.

Conceptual-Captions 12M [Changpinyo et al. 2021]: A dataset comprising of 12 million captioned images, intended for large-scale vision language pretraining. It is larger and more diverse than the Conceptual Captions (CC3M), albeit with noisier captions.

COCO Captions [X. Chen et al. 2015]: A smaller dataset of 113.3k images with five captions per image, resulting in a total of 567k image-text pairs.

Downstream performance of these additional datasets can be found in Table 3.9. We find that restricting the model's pretraining to only a small number of text-image pairs (COCO Captions) expectedly performs worse on downstream data, but still achieves competitive results. Thereby demonstrating the strength of our proposed method and that reasonable performance can be achieved on downstream video data with image pretraining alone.

Increasing the number of pretraining pairs consistently improves downstream performance, albeit with diminishing returns. It appears to be more efficient to add smaller datasets from diverse sources rather than add an increasingly larger dataset

Table 3.9: **Pretraining sources extended:** The effect of different other pretraining sources. We use 4 frames per video when finetuning. Results are presented on the 1K-A MSR-VTT test set for text-video retrieval.

Pre-training	#pairs	R@1	R@5	R@10	MedR
COCO	0.6M	27.2	56.1	67.5	4.0
WV-2M	2.5M	27.5	56.6	67.6	4.0
WV-10M	10M	28.9	57.2	68.6	4.0
CC3M, WV2M	5.0M	31.0	59.5	70.5	3.0
CC3M, WV2M, COCO	5.6M	32.5	61.5	71.2	3.0
CC3M, WV10M	13.0M	33.4	59.2	70.7	3.0
CC3M, CC12M, WV10M	25.0M	34.0	61.4	73.1	3.0

from a single source, shown by the boost of adding COCO captions (567k pairs) to the CC3M+WV2M pretraining compared to adding an extra 7.5 million pairs (WebVid10M) of that same source of data.

3.6 Conclusion

To conclude, we introduce a dual encoder model for end-to-end training of text-video retrieval, designed to take advantage of both large-scale image and video captioning datasets. Our model achieves state-of-the-art performance on a number of downstream benchmarks, however we note that the performance of our model is not saturated yet, and performance could be further improved by training on the full HowTo100M dataset, larger weakly paired image datasets such as Google3BN [C. Jia et al. 2021], as well as multi-dataset combinations thereof.

Acknowledgements. The authors would like to thank Samuel Albanie for his useful feedback. We are grateful for funding from a Royal Society Research Professorship, EPSRC Programme Grant VisualAI EP/T028572/1, and a Google PhD Fellowship.

3.7 Appendix

3.7.1 Additional Benchmark Results

ActivityNet Captions [Krishna et al. 2017a] contains 20K YouTube videos focused on actions, annotated with 100K sentences. The training set consists of 10K videos, and we use the 'val1' set of 4.9K videos to report results. At test time we use paragraph-to-video retrieval as is standard protocol set by other works, where the segment descriptions are concatenated to give a video-level description. We compare to prior work in Table 3.10 and achieve comparable results to the state of the art by using much less training data.

Table 3.10: Text-to-video retrieval results on the ActivityNet val1k set. **R@k:** Recall@K. **MedR:** Median Rank.

Method	E2E	VT PT	**R@1**	**R@5**	**MedR**
FSE			18.2	44.8	8.3
CE [Y. Liu et al. 2019]			18.2	47.7	13.0
CLIPBERT	✓		21.3	49.0	6.0
MMT			22.7	54.2	5.0
SupportSet [Patrick et al. 2020]			26.8	58.1	**3.0**
MMT [Gabeur et al. 2020]		HowTo	28.7	61.4	**3.0**
SupportSet [Patrick et al. 2020]		HowTo	**29.2**	**61.6**	**3.0**
Ours	✓	CC,WebVid-2M	28.8	60.9	**3.0**

3.7.2 Architectural Details

Video Encoder

The video encoder is composed of: (i) the patch embedding layer; (ii) learnable positional space, time and [CLS] embeddings; and (iii) a stack of $|\ell| = 12$ space-time attention blocks

1. The patch embedding layer is implemented as a 2D convolutional layer with a kernel and stride size equivalent to the target patch size $P = 16$, and $d = 768$ output channels (the chosen embedding dimensionality of the video encoder).

2. The positional space and time embeddings are instantiated with shape $M \times d$ and $N \times d$ respectively, where M is the maximum number of input video frames and N is the maximum number of non-overlapping patches of size P within a frame (196 for a video resolution of 224×224). The [CLS] embedding is instantiated with shape $1 \times d$.

3. Each space-time attention block consists of norm layers, temporal and spatial self-attention layers, and an MLP. The order and connections of these layers is shown in Figure 3.4.

Text Encoder

Our text encoder is instantiated as `distilbert-base-uncased` [Sanh et al. 2019]. Distilbert follows the same general architecture as BERT [J. Devlin et al. 2019],

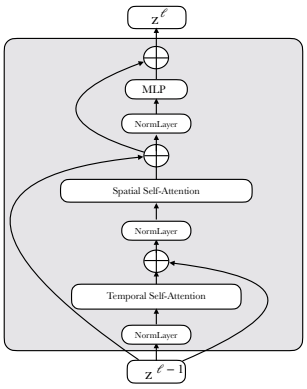


Figure 3.4: Detailed diagram of the space-time self attention block.

but with the number of layers reduced by a factor of 2 and the token-type embeddings and the pooler removed. We use the HuggingFace[2] transformers library implementation.

3.7.3 Architectural Ablations

Video Backbone

We investigate the effects of using different video backbone architectures (Table 3.11) and find that the space-time transformer encoder leads to large improvements in performance on MSR-VTT when compared to ResNets and 3D variants thereof.

During testing, all frame-variants see an equal number of frames, since the video embeddings are averaged over multiple strides.

For the video backbone ablation, we fix the text backbone to `distilbert-base-uncased`. For the text backbone ablation, we fix the video backone to the base space-time transformer with an input resolution of 224 and a patch size $P = 16$.

Text Backbone

The choice of text backbone has a significant impact on downstream performance (Table 3.12), with the t5 models performing significantly worse with more or similar numbers of parameters. DistilBERT and normal BERT achieve similar perfor-

[2] https://huggingface.co/

Table 3.11: **Video backbone.** Text-to-video retrieval results on MSR-VTT test set with different video backbones. All models were pretrained on WebVid-2M and finetuned on MSR-VTT train set. 4 frames were given as input, except for the ResNet-101 which only supports image (1-frame) inputs. The text backbone is fixed to distilbert-base-uncased.

Video Backbone	#params	R@1	R@10	MedR
ResNet-101	45M	11.5	44.1	14.5
S3D-G	76M	3.6	20.4	59.5
R(3D)-101	85M	9.3	38.3	20.0
S-Tformer 224_{16} B	114M	**26.8**	**68.2**	**4.0**

Table 3.12: **Text backbone.** Text-to-video retrieval results on MSR-VTT test set with different text backbones. All models were pretrained on WebVid-2M and finetuned on MSR-VTT train set. The video backbone is fixed to the base space-time transformer with an input resolution of 224 and a patch size $P = 16$.

Text Backbone	#params	R@1	R@10	MedR
t5-small	60.5M	15.1	51.4	10.0
t5-base	222.9M	24.0	62.8	6.0
distilbert-base-uncased	66.4M	26.8	**68.2**	**4.0**
bert-base-uncased	109.5M	**27.5**	67.3	**4.0**

mance, with DistilBERT having far fewer parameters, therefore we chose to use DistilBERT in our work for efficiency.

Space-Time Attention

Space-time attention. Our modified space-time attention block, shown in Fig. 3.5, improves retrieval performance, as show in Table 3.13. We compare both variants during pretraining on WebVid-2M by reporting zero-shot results on MSR-VTT. We find once again that our modification leads to modest performance gains.

Table 3.13: **Space-time attention method:** Zero-shot results are presented on 1K-A MSR-VTT test set for text-video retrieval. The models were trained on WebVid-2M.

Attention Method	R@1	R@10	MedR
Divided Space-Time [Lei et al. 2021]	13.0	40.2	18.0
Ours	14.6	42.7	16.0

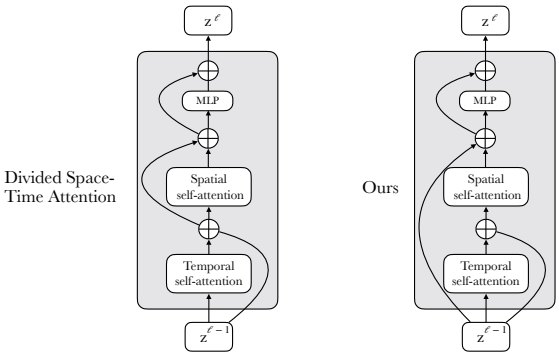


Figure 3.5: **Attention block:** The original divided block used in the Timesformer [Bertasius et al. 2021] architecture (left) compared to ours (right). We find that this minor modification of the input residual connection trains more quickly and is more stable than the original.

Table 3.14: **Temporal expansion method**. The effect of different expansion methods increasing the input number of frames from 4⇒8. Results are presented on 1K-A MSR-VTT test set for text-video retrieval. The models were pre-trained on CC3M & WebVid-2M and finetuned on MSR-VTT train set.

Method	R@1	R@10	MedR
Zero-pad	**30.7**	68.3	4.0
Nearest Neighbour	29.4	69.5	4.0
Bilinear	28.3	**69.9**	4.0

Temporal Expansion

We explore 3 different methods for expanding temporal positional embeddings (zero-padding and two interpolation methods), and observe robustness to all 3 (see Table 3.14).

3.7.4 WebVid-2M Dataset Details

In this section, we show further details of the new WebVid-2M dataset. More qualitative examples of video-text pairs can be found in Figure 3.6 and histograms of caption lengths and video durations can be found in Figure 3.7. Note that 275,000 videos are longer than 30 seconds, providing many examples of videos which can be used for training long-range video models.

1990s: man driving excavator, rotates seat, opens windows in cab. hand presses lever.

Frying pancakes in the kitchen at home. a woman is cooking traditional russian pancakes. modern kitchen, skillet and batter.

Twilight zhuhai famous mountain park top cityscape aerial panorama 4k timelapse china

A child with a suitcase. a happy little girl sits on a suitcase with a passport and money.

Kherson, ukraine - 20 may 2016: open, free, rock music festival crowd partying at a rock concert. hands up, people, fans cheering clapping applauding in kherson, ukraine - 20 may 2016. band performing'

Cockatoos on the fence

Runners feet in a sneakers close up. realistic three dimensional animation.

Ontario, canada january 2014 heavy pretty snow on tree branches

Figure 3.6: **WebVid-2M dataset examples:** We provide additional examples from our dataset by showing video-text pairs, using video thumbnails.

3.7.5 WebVid-10M Extension

To facilitate further text-video pre-training, we extend the WebVid dataset fourfold to 10 million text-video pairs, following the same data collection protocol.

Table 3.15: **WebVid-10M:**

dataset	#clips	avg dur. (secs)	#sent	time (hrs)
WebVid-2M	2.5M	18	2.5M	13K
WebVid-10M	10M	18	10M	52K

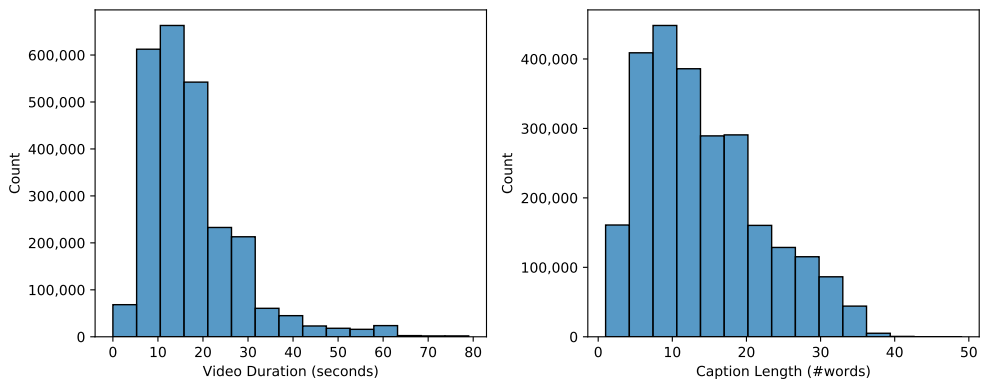


Figure 3.7: **WebVid-2M dataset statistics:** We report the histogram of video duration in seconds **(top)** and the histogram of caption length in words **(bottom)**.

Chapter 4

A Clip-Hitchhiker's Guide to Long Video Retrieval

The paper is a technical report published on ArXiV, 2022.

A CLIP-Hitchhiker's Guide to Long Video Retrieval

Max Bain[1] Arsha Nagrani[1] Gül Varol[1,2] Andrew Zisserman[1]

[1] Visual Geometry Group, University of Oxford

[2] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS

Abstract

Our goal in this paper is the adaptation of image-text models for long video retrieval. Recent works have demonstrated state-of-the-art performance in video retrieval by adopting CLIP, effectively *hitchhiking* on the image-text representation for video tasks. However, there has been limited success in learning temporal aggregation that outperform mean-pooling the image-level representations extracted per frame by CLIP. We find that the simple yet effective baseline of weighted-mean of frame embeddings via query-scoring is a significant improvement above all prior temporal modelling attempts and mean-pooling. In doing so, we provide an improved baseline for others to compare to and demonstrate state-of-the-art performance of this simple baseline on a suite of long video retrieval benchmarks.

4.1 Introduction

Pretrained vision-language models are becoming increasingly ubiquitous due to their impressive performance on a range of downstream tasks with minimal to no additional training data. These models have demonstrated near human-level performance on perception tasks including image classification [Radford et al. 2021], image retrieval [C. Jia et al. 2021], and even object detection [Esmaeilpour et al. 2021; X. Gu et al. 2021]. A major remaining research question is the successful

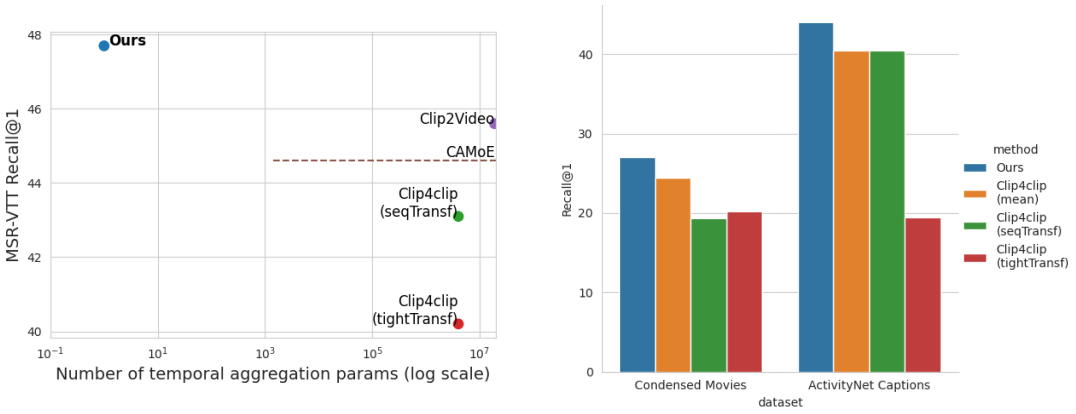


Figure 4.1: A comparison of performance on text-to-video retrieval between state-of-the-art work initialising with CLIP. Recall@1 performance on MSR-VTT [J. Xu et al. 2016] (left) and ActivityNet & Condensed Movies [Bain et al. 2020a] (right). Clip2Video [Fang et al. 2021], CAMoE [Cheng et al. 2021] and Clip4clip [H. Luo et al. 2021] use involved temporal aggregation methods with many learned parameters. We show that our simple query-scoring baseline (with no learned parameters) outperforms all these works by a large margin. We note the number of parameters for CAMoE is an estimation since there was no public implementation at the time of writing.

training and application of vision-language models to tasks requiring higher level cognitive reasoning. One such area, and the focus of this work, is long-form video understanding and its growing body of research [Zhao et al. 2019; C.-Y. Wu et al. 2019b; C.-Y. Wu et al. 2022; C.-Y. Wu and Krahenbuhl 2021a; T. Han et al. 2022].

Machines that can parse long-form videos, understand narrative and abstract concepts, e.g. a movie depicting *two friends falling out and then years later making amends*, is a step towards higher level cognitive reasoning. However, progress in this area has been less fruitful when compared to large-scale language models and tasks [T. Brown et al. 2020; M. Chen et al. 2021]. Thus far, vision-language models proven to be effective at analysing short video clips, achieving state of the art when finetuned on tasks such as video classification [M. Wang et al. 2021; K. Zhou et al. 2021; Castro and Heilbron 2022], text to video retrieval [H. Luo et al. 2021; R. Yan et al. 2021; Cheng et al. 2021] and video question answering [A. Yang et al. 2021; Fu et al. 2021].

Whilst recent works have employed pretrained video-text encoders for downstream video tasks [Bain et al. 2021; R. Yan et al. 2021; H. Xu et al. 2021; Ge et al. 2022], the current state-of-the-art in text-to-video retrieval employs purely image-text representations, specifically OpenAI's CLIP [Radford et al. 2021] – we aptly call

this *CLIP-hitchhiking*. This can largely be attributed to the greater scale of image-text datasets [Radford et al. 2021; C. Jia et al. 2021] compared to video-text, by several orders of magnitude. Adapting these models – originally trained for image data – to video tasks is still an open question and a growing area of research. Of particular note are tasks involving long-form videos, typically with much smaller amounts of training data, higher degrees of temporal structure and variation between frames. Recent work has proposed learning temporal aggregation layers on top of CLIP representations [H. Luo et al. 2021; Cheng et al. 2021; M. Wang et al. 2021]; however, the performance is comparable or even worse than simply taking the mean of the image representation across all the frames in the video. For the case of long-form videos, with duration of several minutes or more, interesting events might only last a few seconds. Mean-pooling in this case is clearly sub-optimal.

We address this limitation in this paper. Motivated by the fact that long videos can contain many redundant frames, such as a long video of a student studying a math problem, as well as occasionally highly-informative frames, such as a few seconds of footage where said student solves the problem and raises their fist in excitement: we show that predicting the relevance of each frame and using these scores to perform a simple weighted mean of the frame embeddings outperforms all the more complex temporal modelling attempts and achieves state-of-the-art text-to-video retrieval on ActivityNet Captions [Krishna et al. 2017a], MSR-VTT [J. Xu et al. 2016], and Condensed Movies [Bain et al. 2020a]. We investigate three methods for computing the frame relevance scores: (1) Query-scoring, the simplest with no learned parameters, using the frame-level similarity to the text query; (2) Self-attention scoring, a sequence transformer taking frame embeddings as input and outputting scores per frame, conditioned only on video information; and (3) Joint-attention scoring, with the same setup as (2) but additionally with the text query embedding appended to the end of the sequence and thereby conditioning on the query. We demonstrate the improvement of our simple baseline method on using CLIP for long-video classification on the Charades dataset [Sigurdsson et al. 2016a]. Our proposed method acts as an improved baseline of mean-pooling for other methods to compare to, especially those which propose aggregation methods on top of CLIP. We further provide insight into the reasons behind the effectiveness

of this simple baseline, namely (i) the mean of frame embeddings are mapped to entirely new locations in the embeddings space and (ii) the effect of performance on datasets with differing amounts of data.

4.2 Related Work

We provide a brief overview of the relevant literature on visual-text representation learning, video-text retrieval, and long video representation learning.

Visual-text representation learning. Learning joint visual-text representation learning is a widely studied and growing area of research [Radford et al. 2021; C. Jia et al. 2021; Junnan Li et al. 2021; Liwei Wang et al. 2016; Y.-C. Chen et al. 2020; X. Li et al. 2020; Alayrac et al. 2022; Junnan Li et al. 2021; Junnan Li et al. 2022]. Such representations have widespread applications in the real world ranging from semantic video search, zero-shot image classification, and human-robot interaction [Goodwin et al. 2022]. Large-scale models trained contrastively on paired visual-text web data has demonstrably shown to learn state-of-the-art image representations capable of impressive zero-shot performance [Radford et al. 2021], although these representations are not without their biases [Berg et al. 2022; Agarwal et al. 2021]. Works have learned how to leverage noisy speech supervision to learn a better video encoder [Miech et al. 2019], as well as incorporating self-supervised learning to improve the visual-text representation [Mu et al. 2021]. We investigate adopting the large-scale pretrained model CLIP [Radford et al. 2021] to the domain of videos, particularly those of long-form.

Video-text retrieval. The text-to-video retrieval area has seen rapid progress over the past few years. First attempts utilised pre-extracted features [Miech et al. 2018], from classification networks trained for example on ImageNet [Russakovsky et al. 2015] and Kinetics [Carreira and Zisserman 2017]. A large portion of these works investigate how best to aggregate features from these different networks [Y. Liu et al. 2019], exploring vector quantization [Xiaohan Wang et al. 2021b], incorporating additional modalities such as audio [Gabeur et al. 2022; Y.-B. Lin et al.

2022; Shvetsova et al. 2021], as well as how to aggregate them over the full video duration (since these features typically have a temporal resolution of a single frame or a couple of seconds).

Most recently, state-of-the-art works have employed CLIP as an image-text backbone and have adapted it to the video setting. Whilst joint text-video pretrained models would be a seemingly better fit, no such models have been made available – in large part due to the lack of available large scale text-video data. Whilst today's publicly available video-text datasets (Howto100M [Miech et al. 2019], Youtube8M [Abu-El-Haija et al. 2016], and WebVid10M [Bain et al. 2021]) make some headway, this is a far cry from the 400M [Radford et al. 2021] or 3BN [X. Zhai et al. 2021] diverse image-text pairs today's models are trained on, which far outperform the video-text models trained on less pretraining data. Current state of the art in video tasks do not utilise video-text pretraining effectively whether this is due the infeasible compute required, the marginal gains over image-text pretraining, or even the scale and quality of the visual-text pairs. [H. Luo et al. 2021; Fang et al. 2021; Cheng et al. 2021] all propose methods using CLIP with additional temporal modelling on top – however the temporal modelling performs comparably or worse than taking the mean embedding across the frames [Castro and Heilbron 2022]. A temporal transformer on top of image-level embeddings, has shown demonstrable improvements in other tasks, but typically require vast amounts of training data and cannot be extensively pretrained due to prohibitive cost and lack of large-scale long video-text data. However, *some* temporal modelling must be done since the mean representation can not intelligently aggregate multiple events over a long video.

To overcome this limitation, we focus on the most restricted form of temporal aggregation of frame embeddings: the weighted-mean. In our experimental study, we explore different ways to obtain these weights.

Long video representation learning. Aggregating temporal information from long videos has been investigated by many works, primarily for video classification [Yue-Hei Ng et al. 2015; Miech et al. 2017; Gaidon et al. 2013; Pirsiavash and Ramanan 2014; Varol et al. 2018; Jue Wang and Cherian 2018; Limin Wang et al. 2016]. Closest to our work is SCSampler [Korbar et al. 2019] which samples the

top-k most salient clips from a long video to use for video classification. SCsampler tackles the sampling with a separate, cheaper model, which is learned separately from model training. The clips selected by the sampling model are then fed to the actual classification network to average scores over the top-k frames. This differs from our work in that it focuses on efficient sampling for video classification by using a cheaper model, learned separately. Our investigation involves weighting frame samples online during training to improve performance and learning.

The joint image-text representation offers a unique advantage to these works where the text-query representation can be used to guide the temporal aggregation. Although more costly, this approach can use the query to guide the temporal aggregation. Works have used query scoring in this way to perform weakly-supervised action and moment localisation, by thresholding frames past a certain similarity with the text query. In a similar vein to these works, we show that using the query to guide the linear weighting of the frame embeddings can achieve state-of-the-art performance without any additional learning.

4.3 Temporal Aggregation of Image-Text Representations

We consider the problem of learning joint text-video representations from a set of video-text pairs (V, T) where V is a video of K frames and T is the corresponding text describing the video. Specifically, we consider the case where representations are extracted for the text, $T \in \mathbb{R}^d$ and every frame of the video, $V = [I^{(1)}, I^{(2)}, ..., I^{(K)}] \in \mathbb{R}^{K \times d}$, via a pretrained image-text model, such as CLIP [Radford et al. 2021]. Our goal is to find an aggregation method Φ that combines the frame representations into a single video-level representation, $\bar{V} = \Phi(V) \in \mathbb{R}^d$, such that semantically similar instances of $\bar{V}, T \in \mathbb{R}^d$ are close to each other.

Prior works have instantiated Φ with self-attention networks [Fang et al. 2021], squeeze-and-excitation networks [Cheng et al. 2021], and even cross-transformer layers with the query [H. Luo et al. 2021]. However, it has been shown that simply taking the mean of every frame embedding achieves comparable or even superior performance to these temporal aggregation attempts on many benchmarks.

This failure of temporal modelling for videos, especially those of long-form is sub-optimal. Video frames have varying degrees of relevance. Motivated by this, as well as the effectiveness of mean-pooling, we propose a straightforward but effective improvement to the uniform mean, inspired by weakly-supervised moment localisation [Mithun et al. 2019], by using query-frame scoring to perform the weighted-mean of frame embeddings. Given a sequence of corresponding per-frame relevance scores, $S = [s_1, s_2, ..., s_K] \in \mathbb{R}^K$ where $s_i = I^{(i)} \cdot T$, we can compute a final embedding for the whole video $\bar{V} \in \mathbb{R}^d$ via the weighted-mean.:

$$\bar{V} = \sum_{k \in K} w_k I^{(k)} \quad \text{where} \quad w_k = \frac{e^{s_k/\tau}}{\sum_{j \in K} e^{s_j/\tau}}, \tag{4.1}$$

where the softmax temperature τ can be interpreted as a hyperparameter towards the highest scoring frames. For very small values of τ, this becomes an argmax operation, where the final video embedding is simply the single most relevant frame. Equally, for very large values of τ, the weights become uniform, effectively ignoring the scores. Formally:

$$\lim_{\tau \to 0} \bar{V} \to I^{(k')} \quad \text{where} \quad k' = \operatorname*{argmin}_k S \quad \text{and} \quad \lim_{\tau \to \infty} \bar{V}_i \to \frac{1}{K} \sum_{k \in K} I^{(k)}. \tag{4.2}$$

In practice we find the some middle range offers a good balance between weighted relevant frames more as well as capturing the full temporal of content in the video. We explore different values of τ in our empirical evaluation.

In the following, we describe alternative scoring methods and their complexity (Sec. 4.3.1), alternative aggregation methods (Sec. 4.3.2), as well as the framing of this problem to video classification (Sec. 4.3.3).

4.3.1 Alternative Scoring Methods

Whilst the above scoring method is parameter-free, except for the choice of τ, we also investigate more involved methods to predict the scores of each frame. Since the scores can only be used to linearly combine the original frame embeddings, the model is heavily regularised in what it can do and therefore it allows heavy temporal modelling networks to be used but constrains their influence to only linear combinations of the original image-text representation.

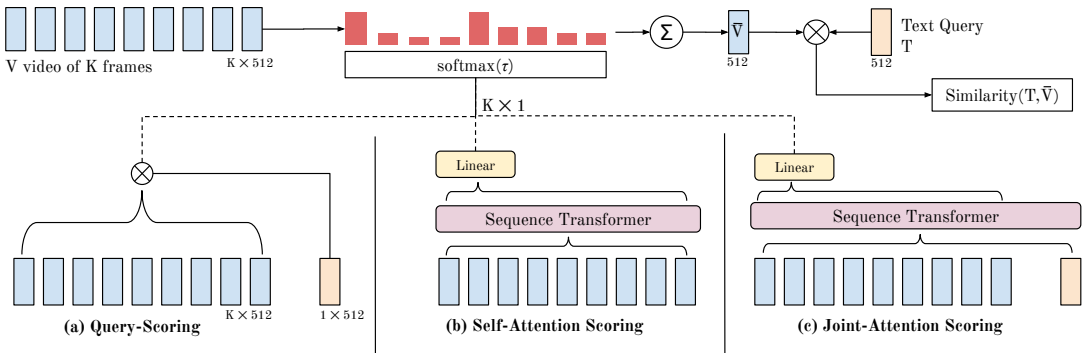


Figure 4.2: The different scoring methods used to predict the relevance scores of each frame embedding from a video, these are softmaxed as used to compute the weighted-mean single video representation. **Query-scoring (a)**, the simplest scoring method with no learned parameters, scores each frame by the similarity of each frame embedding with the text query. **Self-attention scoring (b)**, uses a sequence transformer on the $K \times 512$ frame embeddings, and scores each frame by feed each output frame embedding output sequence through a linear layer $\mathbb{R}^{512} \rightarrow \mathbb{R}^1$. **Joint-attention scoring (c)**, uses the same approach as (b) with the addition of the text query embedding appended to the input sequence of the transformer.

Self-Attention

layers can be used on the frame embeddings to predict the relevance scores, as shown in Figure 4.2. The output frame embeddings of the self-attention layers are fed through a linear layer $\mathbb{R}^d \rightarrow \mathbb{R}$ to produce scalar relevance scores per frame. This method has the advantage in that the frame scores S are independent from the query, and therefore also the final video representation $\bar{V}$. This keeps the retrieval complexity to constant time $\mathcal{O}(1)$ (see Table 4.1).

Joint-Attention Scoring

is an extension of the above method, this additionally includes the query as input to the attention layers and performs both cross and self-attention between the query and frame embeddings. Whilst higher in complexity, conditioning the video specific to the query makes sense, as otherwise the definition of frame relevance may be ambiguous.

Neither self-attention layers nor joint-attention layers are new for temporalling modelling, however our instantiation differs in that we do not use the final output embeddings of the attention layers as the video representation, but instead map their output $\mathbb{R}^d \rightarrow \mathbb{R}$ to scalars used to weight the frame embedding averaging.

Table 4.1: v is the number of videos in the retrieval set, k is the number of frames per video, n is the number of layers in the transformer.

Scoring Method	Retrieval Complexity	Model Complexity	Video Space Complexity
Mean-pooling	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(v)$
Query	$\mathcal{O}(v)$	$\mathcal{O}(1)$	$\mathcal{O}(vk)$
Temporal Self-Attn.	$\mathcal{O}(1)$	$\mathcal{O}(nk^2)$	$\mathcal{O}(v)$
Joint Attn.	$\mathcal{O}(v)$	$\mathcal{O}(nk^2)$	$\mathcal{O}(v)$

Complexity.

The space and time complexities of the different scoring methods are shown in Table 4.1, with increasing complexity down the rows. Since query-scoring needs no learnable parameters (like uniform mean), the method can be applied to zero-shot video tasks using image-text only embeddings. The query-conditioned frame aggregation increases retrieval complexity to $\mathcal{O}(\sqsubseteq)$, since the weighted-mean is specific to each query – however in practice we find that 64 or 120 frame embeddings is sufficient for a long video of several minutes. Storing such an array per video is a small increase in space and the dot product operation is marginal.

The retrieval complexity of query-dependent aggregation can factored down by only employing it for the top K ranked results, and using query-independent aggregation for the full ranking [Miech et al. 2021]. First a rough ranking is performed on mean embeddings, without query-specific aggregation, and then a more costly query scoring method can be used.

Temporal self-attention and joint attention have the same retrieval and model complexity as prior work temporal modelling attempts – only with an additional linear layer to map the embeddings to frame scores.

4.3.2 Alternative Aggregation Methods

Hard Top-K.

An alternative to taking the weighted-mean via the softmax scores, can be to take the mean of the hard top-K frames. This is the approach adopted by [Korbar et al. 2019] to select which clips should be aggregated for video classification. Unlike the soft query-scoring which, with the exception of very low values of τ, still includes

some amount of information from *every* frame (due to the soft operation), top-k entirely removes them from the aggregation and treats all the top k frames equally.

Averaging per-frame logits rather than embeddings.

Similarly, rather than taking the weighted-mean of frame-level embeddings for a single video representation, one can average the similarity logits in order to calculate their similarity to the text. We find that although this performs comparably under zero-shot setting, the hard top-k performance worse when finetuning – we show results in Section 4.4.4.

4.3.3 Video-to-text retrieval and video classification

Whilst the discussed methods have been described in a video-text retrieval setting, it can be equally applied to video classification by formulating the classification task as video-to-text retrieval. The only differences to note would be the complexity analysis, where the video space complexity is no longer a concern since video embeddings need not be stored for text retrieval. Additionally, query-condition aggregation is less of a concern since the number of text queries is fixed to the number to video action labels, which tends to be small.

4.4 Experiments

In this section, we start by presenting the downstream datasets (Sec. 4.4.1) and the experiment protocol (Sec. 4.4.2). Next, we report state-of-the-art results across the chosen suite of long-form video retrieval benchmarks (Sec. 4.4.3). Then we perform investigation into the effectiveness of the simple weighted-mean aggregation and compare to alternative methods (Sec. 4.4.4).

4.4.1 Downstream Datasets

We now describe the downstream text-to-video retrieval datasets our model is evaluated on, focused on those with long durations, as well as additionally a long

video classification dataset.

MSR-VTT [**J. Xu et al. 2016**] benchmark contains 10K videos from YouTube with 5 captions per video, we trained on 9K videos and report results on the 1K-A [Y. Yu et al. 2018] test set, this dataset contains the shortest videos of which we evaluate on, averaging 15 seconds.

Condensed Movies Dataset (CMD) [**Bain et al. 2020a**] is a long-form text-video dataset consisting of 34K videos of movie scenes with an average duration of 132 seconds and corresponding high-level semantic textual descriptions. We train, validate and test on the recent challenge split [*Condensed Movies Challenge* n.d.] of 32K, 2K, and 1K videos, respectively and compare to the Codalab leaderboard as well as evaluate results on other competing methods. Within a long movie scene there is a large variation among the events that occur and the information between shots, aggregating this over several minutes to match to a high-level semantic description requires a high degree of video understanding.

ActivityNet Captions [**Krishna et al. 2017a**] consists of 20K videos from YouTube focused primarily on actions, annotated with 100K sentences. The training set consists of 10K videos, and we use the val1 set of videos to report results. With an average video duration of 180 seconds, these are the longest videos we evaluate on – capturing a diverse range of events and actions within the two minutes. The captions consist of a sequence of descriptions of localised moments within the video. We employ the standard paragraph-to-video retrieval [Bain et al. 2021] protocol when training and testing by concatenating the text sequences and evaluating on the whole long-form video.

Charades [**Sigurdsson et al. 2016a**] is a video classification dataset consisting of daily activities with an average duration of 30 seconds. The classification is multilabel and multiclass in that a video can contain multiple different actions at different times. This is a valuable setting for long-form video understanding since the action classes vary over the duration.

4.4.2 Experiment Protocol

We use CLIP ViT-B/16 [Radford et al. 2021] in all experiments and finetune the model end-to-end with the Adam optimizer [Kingma and Ba 2014] (learning rate

set to 5e-7). Finetuning is done on two GPUs with a batch size of eight per GPU, and sample 16 frames from each video during training, randomly sampled within the video clip (we found this superior to the sub-segment sampling in [Bain et al. 2021]). At test time 120 frames are sampled uniformly from the video, irrespective of their length, unless specified otherwise. For the text, words are dropped randomly during training with a probability of 10%. For query-scoring, we use the $\tau = 0.1$ across all datasets to demonstrate the consistent performance (although optimal values might vary slightly between datasets). For the self and joint attention scoring methods, we use single-layer networks as was minimal difference in performance when using additional layers.

4.4.3 Results

Comparison to the state of the art.

We present the text-to-video retrieval results of the query-scoring method on MSR-VTT, ActivityNet and CondensedMovies in Tables 4.2 and 4.3. We achieve state-of-the art performance on all three datasets, significantly outperforming prior work aggregation methods using a CLIP backbone with millions of learned parameters. In contrast, our query-scoring method has only a single parameter. These results demonstrate the surprising effectiveness of *weighted-mean* embeddings, and the limitations of current, more involved temporal aggregation methods. Query-scoring acts an improved baseline to proposed temporal aggregation methods.

We also compare to results on the long video classification dataset Charades in Tab. 4.4. CLIP with query-scoring achieves the same performance as ActionCLIP [M. Wang et al. 2021] which uses temporal modelling, averages predictions over 320 total frames, and a set of prompt templates. Query-scoring uses none of these yet offers similar performance, and outstanding improvements to CLIP4CLIP *seqTransf* and the baseline of mean pooling the frame embeddings.

Test-time normalisation. Recent concurrent works achieve state of the art performance via CLIP with test-time normalisation such as QueryBank [Cheng et al. 2021; Bogolin et al. 2022] and dual softmax normalisation. The latter however requires access to all queries at test-time, which is not appropriate for real-world

Table 4.2: Comparison to state-of-the-art results on MSR-VTT 1k-A for text-to-video retrieval. The bottom section compares to methods using CLIP as the backbone image-text encoder, their different temporal aggregation methods (agg.), and the number of parameters learned for the aggregation. "−" indicates an unknown value either due to no official public implementation or lack of reporting in the paper. **R@k:** Recall@K,**MedR:** Median Rank, **MnR:** Mean Rank.

Method			R@1↑	R@5↑	R@10↑	MedR↓	MnR↓
JSFusion [Y. Yu et al. 2018]			10.2	31.2	43.2.4	12	-
CE [Y. Liu et al. 2019]			20.9	48.8	62.4	6	-
MMT [Gabeur et al. 2020]			26.6	57.1	69.6	4	-
SSB [Patrick et al. 2020]			30.1	58.5	69.3	3	-
TeachText [Croitoru et al. 2021]			29.6	61.6	74.2	3	-
Frozen [Bain et al. 2021]			32.5	61.5	71.2	3	-
Method	**agg.**	**#agg. params**					
Clip4clip [H. Luo et al. 2021]	mean	0	43.1	70.4	80.8	2	16.2
Clip4clip [H. Luo et al. 2021]	tightTransf	4M	40.2	71.5	70.5	2	13.4
Clip4clip [H. Luo et al. 2021]	seqTransf	4M	44.5	71.4	81.6	2	15.3
CAMoE [Cheng et al. 2021]	S.E attn.	-	44.6	72.6	81.8	2	13.3
Clip2Video [Fang et al. 2021]	TDB,TAB	19M	45.6	72.6	81.7	2	14.6
Ours	Q-score	1	**47.7**	**74.1**	**82.9**	2	**11.5**

Table 4.3: Comparison to the start-of-the-art results on Condensed Movies and ActivityNet Challenge for text-to-video retrieval.

	Condensed Movies					ActivityNet				
Method	**R@1**	**R@5**	**R@10**	**MdR**	**MnR**	**R@1**	**R@5**	**R@10**	**MdR**	**MnR**
TeachText	12.1	27.4	37.5	-	-	25.0	58.7	-	4	-
Frozen	12.6	28.4	36.3	25	-	28.8	-	60.9	3	-
Clip4clip (mean)	24.4	48.2	58.2	6	46.2	40.5	72.4	-	2	7.4
Clip4clip (seqTransf)	19.3	44.4	55.3	9	55.2	40.5	72.4	-	2	7.5
Ours (q-score)	**27.0**	**52.3**	**61.2**	**5**	**41.2**	**44.0**	**74.9**	**86.1**	2	**5.8**

tasks. Both of these can be added to our method to achieve superior performance, but is not the focus of this work since it is separate to temporal modelling.

4.4.4 Ablation study

Alternative scoring methods. In Table 4.5 we show the performance of alternative scoring methods, and that the performance is notably high across the board – even without query information. This indicates that the strong baseline of weighted-mean of image-level CLIP embeddings is the best current approach of temporal modelling. The seemingly consistent boost no matter the scoring method indicates that the benefit is mainly afforded by restricting the aggregation Φ to linear weighted of the image embeddings. For ActivityNet, temporal self-attention scoring performs significantly better than the other scoring methods which is in contrast to CMD and MSR-VTT. One possible explanation for this is that ActivityNet captions are long paragraphs containing dense descriptions on the video by

Table 4.4: Multi-label classification results on the Charades dataset, where mAP is mean average precision. ActionCLIP results are computed by average 32 frame predictions over 10 (spatial) by 3 (temporal) views.

Backbone	Aggregation	Frame	Finetune	mAP
CLIP (ViT-B/16)	ActionCLIP [M. Wang et al. 2021]	$32_{\times 10 \times 3}$	✓	44.6
	Clip4clip$_{seqTr}$	32	✓	32.0
	Mean	32	✓	33.0
	Q-scoring	32	✓	**44.9**
	Temp. S-A	32	✓	36.3
	Joint. S-A	32	✓	42.2
CLIP (ViT-B/16)	Mean	32	✗	17.5
	Q-scoring	32	✗	**21.1**

Table 4.5: Comparison of the different proposed scoring methods on MSR-VTT, ActivityNet Captions and Condensed Movies test sets for text-to-video retrieval..

Scoring Method	**MSR-VTT**				**Condensed Movies**				**ActivityNet Captions**			
	R@1	**R@5**	**R@10**	**MnR**	**R@1**	**R@5**	**R@10**	**MnR**	**R@1**	**R@5**	**R@10**	**MnR**
Baseline	44.4	71.6	79.8	12.8	24.4	48.2	58.2	46.2	42.0	73.1	84.6	7.4
Query	**47.7**	**74.1**	82.9	11.5	**27.0**	**52.3**	**61.2**	**41.2**	44.0	74.9	86.1	5.8
Temp. S-Attn.	46.2	71.4	81.6	12.9	26.2	51.9	62.4	41.5	**44.9**	**75.9**	**86.9**	**5.6**
Joint Attn.	45.7	73.8	**83.6**	**10.6**	26.4	51.1	62.0	43.0	43.1	74.1	85.5	6.7

concatenating localised descriptions. Such a dense video description would be less useful when query-scoring since most frames relate to the query. This is unlike Charades where an action class might only correspond to a few seconds of a 30-second video amongst a sequence of other actions. Instead, the dense description setting of ActivityNet becomes a case of removing noisy frames, which we believe can be done without query-level information.

In contrast CMD, movies with long videos but extremely concise descriptions pertaining to a sequence of a few events in the video, affords the greatest benefits.

Alternative aggregation methods. Comparing different aggregation methods in Table 4.6, we find that averaging over features is considerably better in both zero-shot and finetuning settings. Hard top-K seems to offer a comparable improvement over the baseline in the zero-shot setting, however we find soft scoring features for the finetune setting is most optimal. A smoother feature selection of frames seems to be more favourable for training.

Effect of number of frames. Unsurprisingly performance of both the baseline and query-scoring improves with increasing the number of input frames at test time, albeit with diminishing after 1fps, shown in Figure 4.3 (left). The improvement of query-scoring over the baseline also increases with the number of frames.

Table 4.6: Comparison of different frame aggregation methods and aggregation sources and their respective performance on zero-shot Condensed Movies, zero-shot MSR-VTT, and finetuned MSR-VTT text-to-video retrieval settings denoted by CMD ZS, MSR ZS and MSR FT repsectively. The reported value is the geometric mean of R@{1,5,10} to text-to-video retrieval performance. HParam denotes the hyperparameter selection for the chosen method. † Since training , training with $K = 60$ is not possible, so this setting is trained on $K = 8$ and evaluated on $K = 60$.

Aggregation Method	**HParam**	**CMD ZS**		**MSR ZS**		**MSR FT**	
		Agg. source		Agg. source		Agg. source	
		Score	Feature	Score	Feature	Score	Feature
Mean-pooling		27.6	29.5	48.3	49.5	60.5	62.2
Top-K	K=1	28.8	28.8	48.8	48.8	63.3	63.4
	K=8	32.6	33.3	50.9	50.3	63.3	64.5
	K=60†	30.0	30.0	51.2	50.2	61.9	64.0
Query-scoring	τ=0.01	27.1	30.6	48.9	49.0	62.4	64.4
	τ=0.1	28.8	30.9	50.7	50.5	**63.8**	**65.4**
	τ=1.0	27.7	29.5	48.5	49.6	60.1	62.8

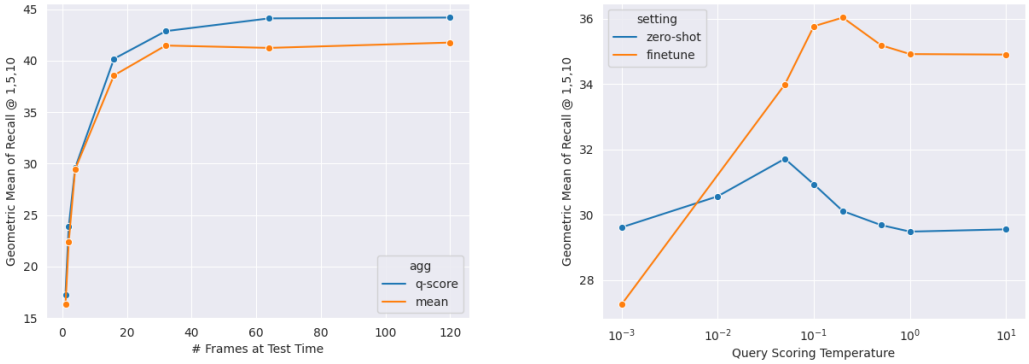


Figure 4.3: Downstream performance effects of varying the number of frames at test time (left) and the query-scoring temperature τ (right), on the CMD test set for text-to-video retrieval.

Intuitively this makes sense, more input frames means there are more relevant frames to pick from and less relevant frames to ignore. This further motivates the use case of frame relevance scoring for long-form videos.

Effect of scoring temperature. We find a temperature range between 0.05 and 0.15 is consistently optimal across long range datasets and tasks. Figure **??** (right) shows the effect of temperature on the CMD test set under both zero-shot and finetuning conditions. Zero-shot performance tends to be slightly better with smaller values of τ, which indicates the effect of τ during the learning process. For extremely large values of τ, gradients only pass from the most similar text query which might cause drifting errors in the learning process (for the case where the initial similarity is wrong).

Table 4.7: Classification accuracy between normalised single-frame embeddings and 16-frame mean-pooled embeddings from CMD (training on the train set and testing on the test set). A single linear layer with high accuracy can discriminate between single-frame and 16-frame embeddings.

	Test accuracy (%)	
Classifier	Zero-Shot	Finetuned
Linear	89.0	90.4
MLP	97.4	98.2

Why are weighted-mean frame embeddings so effective?

a) Insufficient training data for learning new long-video text representations. We find the relative performance boost of mean-weighted embeddings compared to more complex and learned temporal aggregations is reduced with the larger scale downstream datasets. This implies that with enough long video-text pairs, the more complex modelling attempts can outperform this simple baseline.

b)The mean of frame embeddings captures distinct information. Given that CLIP embeddings are trained for the single image-text setting, it is surprising that taking the mean over many frames with vastly different content still performs well. For example, it is possible that the mean of embeddings from two semantically different frames maps to a semantically incorrect new space in the embeddings. To investigate whether this happens, we train both a linear classifier and a multi-layer perceptron (MLP) to classify between single-frame embeddings and mean embeddings from 16 frames sampled from a long video in CMD. We find that both are able to easily classify between these two, even with zero-shot embeddings (Table 4.7). These results suggest the mean-frame embeddings are mapped to entirely new locations in the embedding space, disjoint from the single-frame embeddings. This is encouraging since it suggests that CLIP can learn to capture multi-frame information within the 512 embeddings – and hence the strong baseline out weighted-mean performing so well.

c) Query scoring during training improves single-frame representation. The performance boost of query scoring after finetuning could be attributed to either (i) test time improvements by ignoring irrelevant improvements and/or (ii) improvements to the image-text level representation during training by contrastively learning on more semantically relevant frames. In order to investigate whether (ii)

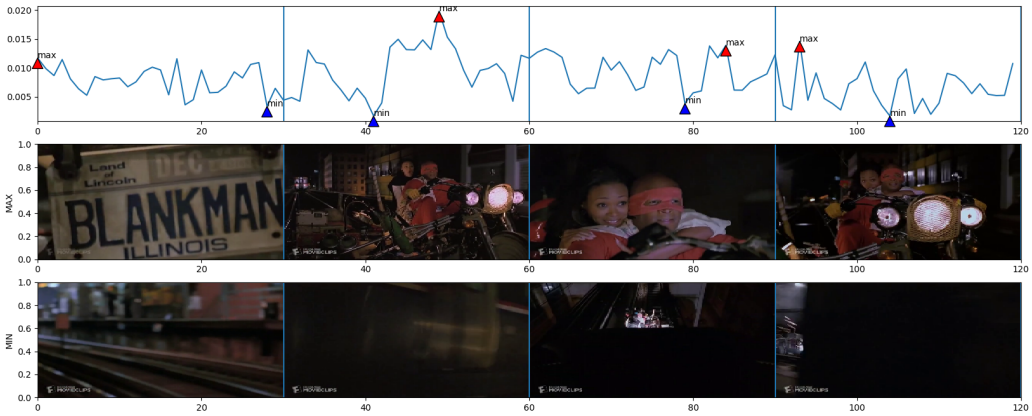
Table 4.8: Single-frame retrieval results on CMD test set. Finetuning with query scoring improves the image-level representation – showing the effectiveness of constrastive learning on video-text pairs with relevance scoring on frames.

Scoring Method	R@1	R@5	R@10	MedR	MnR
Mean	8.6	19.6	25.8	56.0	151
Q-scoring	9.0	21.1	27.3	52.5	148

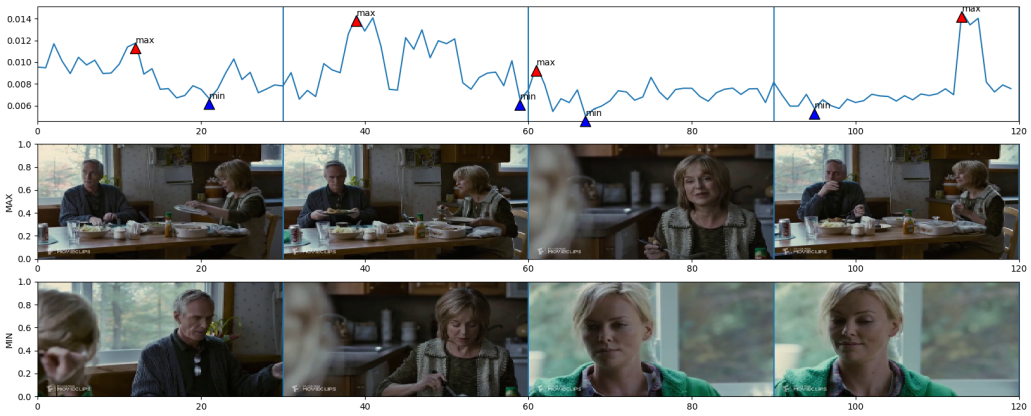
is true we evaluate on CMD test set retrieval in the 1-frame setting, to evaluate the single frame representation (Table 4.7). We find that query-scoring performance provides notable improvement to the image-text representation, indicating that such a method is valuable during the video-text learning process over the baseline of mean-pooling.

Are the frame scores semantically meaningful?

The notable improvement gains via frame scores implies some semantic relevance of the higher scoring frames. To investigate this we show qualitative results of the query scoring for CMD unseen videos. In Figure 4.4 we see that highest scoring frames are those with semantic similarity to the test, for example the frames containing the motorbike as well as the license registration of the character name in the title. Additionally, we see that the lowest scoring frames contain less information and have less relevance to the query. By utilising frame scoring during training, the constrastive loss is weighted less towards these irrelevant frames which could otherwise harm the representation.



(a) "Blankman gives Kimberly a ride on his wacky motorcycle."



(b) "Mavis has an awkward lunch with her parents."

Figure 4.4: Visualisation of query-scoring from unseen videos in the Condensed Movies test set and their corresponding textual descriptions. The first row shows the weights assigned to each frame, with the maximum and minimum scores per segment marked and their corresponding frames in the middle and bottom rows respectively. The highest scoring frames in both (a) and (b) have high relevance to the text query and contain more information than the lowest scoring frames containing background frames or single person close-up shots.

4.5 Conclusion

To conclude, we propose three simple ways to mean-weight frame embeddings from a joint image-text representation for long video retrieval and classification – with and without query information, in doing so picking out the most salient frames. Our method provides a strong baseline, outperforming all prior works across four datasets, including attempts at more complicated temporal modelling. Our experiments uncover some insight into the benefits afforded by this highly constrained temporal aggregation and the challenges posed for more involved temporal modelling. Future work could look into tackling the lack of large-scale data needed to effectively learn effective long-form video representations. This could be done by employing self-supervised learning in addition to the scarcer textual supervision

for long-form video data.

4.6 Appendix

4.6.1 MSR-VTT Full Split

In Table 4.9 We additionally report results on the MSR-VTT full split which consists of 6513, 497 and 2990 videos for training, validation and testing respectively. Query-scoring outperforms all prior work, except in mean rank, which could be attributed to the smaller amounts of training data.

Table 4.9: Comparison to state-of-the-art results on MSR-VTT full split with 7k training for text-to-video retrieval. The bottom section compares to methods using CLIP as the backbone image-text encoder, their different temporal aggregation methods (agg.), and the number of parameters learned for the aggregation. "−" indicates an unknown value either due to no official public implementation or lack of reporting in the paper.

Method			**R@1↑**	**R@5↑**	**R@10↑**	**MedR↓**	**MnR↓**
JSFusion [Y. Yu et al. 2018]			10.2	31.2	43.2.4	12	-
CE [Y. Liu et al. 2019]			10.0	29.0	41.2	16	86.8
TeachText [Croitoru et al. 2021]			15.0	38.5	51.7	10	-
Method	**agg.**	**#agg. params**					
Clip2Video [Fang et al. 2021]	TDB,TAB	19M	29.8	55.5	66.2	4	45.4
CAMoE [Cheng et al. 2021]	S.E attn.	-	32.9	58.3	68.4	3	42.6
Ours	Q-score	1	34.9	59.4	68.7	3	49.4

Extended Comparison of Aggregation Methods

Finetuning on Condensed Movies also shows a pronounced boost for query-scoring when compared to alternative aggregation methods: hard top-k and mean-pooling (see Table 4.10). Further demonstrating the benefit of query-scoring in the finetuning setting.

Table 4.10: Comparison between different scoring methods when finetuning on Condensed Movies, results are shown for text-to-video retrieval.

Aggregation Method	**Hparam**	**R@1↑**	**R@5↑**	**R@10↑**	**MedR↓**	**MnR↓**
Mean-pooling		25.6	49.2	59.9	6	42.2
Top-K	K=1	19.6	43.8	52.9	8	54.0
	K=8	25.4	50.2	58.1	5	51.6
	K=60	26.0	51.6	61.1	6	41.6
Query-scoring	τ=0.01	24.0	47.4	57.8	6	52.8
	τ=0.1	**27.3**	**52.8**	**62.0**	**4**	**40.4**
	τ=1.0	25.5	49.5	60.3	6	42.3

Part II

Automated Movie Audio Description

Chapter 5

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

The paper has been accepted for publication as an oral presentation at INTERSPEECH, 2023.

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

Max Bain Jaesung Huh Tengda Han

Andrew Zisserman

Visual Geometry Group, University of Oxford

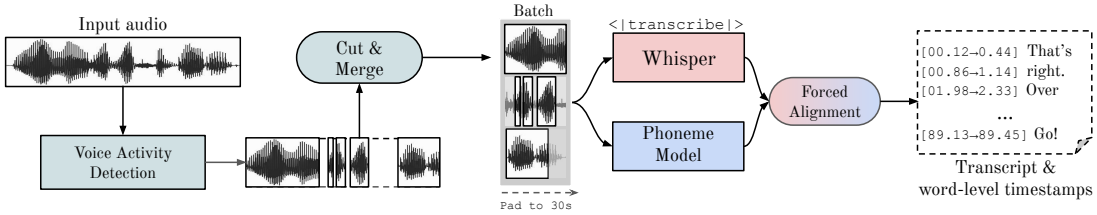


Figure 5.1: **WhisperX**: We present a system for efficient speech transcription of long-form audio with *word-level time alignment*. The input audio is first segmented with Voice Activity Detection and then cut & merged into approximately 30-second input chunks with boundaries that lie on minimally active speech regions. The resulting chunks are then: (i) transcribed in parallel with Whisper, and (ii) forced aligned with a phoneme recognition model to produce accurate word-level timestamps at high throughput.

Abstract

Large-scale, weakly-supervised speech recognition models, such as Whisper, have demonstrated impressive results on speech recognition across domains and languages. However, the predicted timestamps corresponding to each utterance are prone to inaccuracies, and word-level timestamps are not available out-of-the-box. Further, their application to long audio via buffered transcription prohibits batched inference due to their sequential nature. To overcome the aforementioned challenges, we present *WhisperX*, a time-accurate speech recognition system with word-level timestamps utilising voice activity detection and forced phoneme alignment. In doing so, we demonstrate state-of-the-art performance on long-form transcription and word segmentation benchmarks. Additionally, we show that pre-segmenting audio with our proposed VAD Cut & Merge strategy improves transcription quality and enables a *twelve-fold* transcription speedup via batched

inference. The code is available open-source[1].

5.1 Introduction

With the availability of large-scale web datasets, weakly-supervised and unsupervised training methods have demonstrated impressive performance on a multitude of speech processing tasks; including speech recognition [Y. Jia et al. 2019; Baevski et al. 2020; Sanyuan Chen et al. 2022], speaker recognition [Kang et al. 2022; Hui Chen et al. 2023], speech separation [Wisdom et al. 2020], and keyword spotting [Gong et al. 2022; Prajwal et al. 2021]. Whisper [Radford et al. 2022b] utilises this rich source of data to another scale. Leveraging 680,000 hours of noisy speech training data, including 96 other languages and 125,000 hours of English translation data, it showcases that weakly supervised pretraining of a simple encoder-decoder transformer [Vaswani et al. 2017] can robustly achieve *zero-shot* multilingual speech transcription on existing benchmarks.

Most of the academic benchmarks are comprised of short utterances, whereas real-world applications typically require transcribing long-form audio that can easily be hours or minutes long, such as meetings, podcasts and videos. Automatic Speech Recognition (ASR) models are typically trained on short audio segments (30 seconds for the case of Whisper) and the transformer architectures prohibit transcription of arbitrarily long input audio due to memory constraints.

Recent works [Chiu et al. 2019] employ heuristic sliding window style approaches that are prone to errors due to overlapping or incomplete audio (e.g. words being cut halfway through). Whisper proposes a buffered transcription approach that relies on accurate timestamp prediction to determine the amount to shift the subsequent input window by. Such a method is prone to severe drifting since timestamp inaccuracies in one window can accumulate to subsequent windows. The hand-crafted heuristics employed have achieved limited success.

A plethora of works exist on "forced alignment", aligning speech transcripts with audio at the word or phoneme level. Traditionally, this involves training acoustic phoneme models in a Hidden Markov Model (HMM) [Brugnara et al. 1993;

[1] https://github.com/m-bain/whisperX

Gorman et al. 2011; J. Yuan et al. 2013; McAuliffe et al. 2017] framework using external boundary correction models [Kim and Conkie 2002; Stolcke et al. 2014]. Recent works employ deep learning strategies, such as a bi-directional attention matrix [Jingbei Li et al. 2022] or CTC-segmentation with an end-to-end trained model [Kürzinger et al. 2020]. Further improvements may come from combining a state-of-the-art ASR model with a light-weight phoneme recognition model, both of which are trained with large-scale datasets.

To address these challenges, we propose *WhisperX*, a system for efficient speech transcription of long-form audio with accurate word-level timestamps. It consists of three additional stages to Whisper transcription: (i) pre-segmenting the input audio with an external Voice Activity Detection (VAD) model; (ii) cut and merging the resulting VAD segments into approximately 30 seconds input chunks with boundaries lying on minimally active speech regions enabling batched whisper transcription; and finally (iii) forced alignment with an external phoneme model to provide accurate word-level timestamps.

5.2 WhisperX

In this section we describe *WhisperX* and its components for long-form speech transcription with word-level alignment.

5.2.1 Voice Activity Detection

Voice activity detection (VAD) refers to the process of identifying regions within an audio stream that contain speech. For *WhisperX*, we first pre-segment the input audio with VAD. This provides the following three benefits: (1) VAD is much cheaper than ASR and avoids unnecessary forward passes of the latter during long inactive speech regions. (2) The audio can be sliced into chunks with boundaries that do not lie on active speech regions, thereby minimising errors due to boundary effects and enabling parallelised transcription. Finally, (3) the speech boundaries provided by the VAD model can be used to constrain the word-level alignment task to more local segments and remove reliance on Whisper timestamps – which we show to be too unreliable.

VAD is typically formulated as a sequence labelling task; the input audio waveform is represented as a sequence of acoustic feature vectors extracted per time step $\mathbf{A} = \{a_1, a_2, ..., a_T\}$ and the output is a sequence of binary labels $\mathbf{y} = \{y_1, y_2, ..., y_T\}$, where $y_t = 1$ if there is speech at time step t and $y_t = 0$ otherwise.

In practice, the VAD model $\Omega_V : \mathbf{A} \rightarrow \mathbf{y}$ is instantiated as a neural network, whereby the output predictions $y_t \in [0, 1]$ are post-processed with a *binarize* step – consisting of a smoothing stage (onset/offset thresholds) and decision stage (min. duration on/off) [Gelly and Gauvain 2018].

The binary predictions can then represented as a sequence of active speech segments $\mathbf{s} = \{s_1, s_2, ..., s_N\}$, with start and end indexes $s_i = (t_0^i, t_1^i)$.

5.2.2 VAD Cut & Merge

Active speech segments $\mathbf{s}$ can be of arbitrary lengths, much shorter or longer than the maximum input duration of the ASR model, in this case Whisper. These Longer segments cannot be transcribed with a single forward pass. To address this, we propose a 'min-cut' operation in the smoothing stage of the binary post-processing to provide an upper bound on the duration of active speech segments.

Specifically, we limit the length of active speech segments to be no longer than the maximum input duration of the ASR model. This is achieved by cutting longer speech segments at the point of minimum voice activation score (min-cut). To ensure the newly divided speech segments are not exceedingly short and have sufficient context, the cut is restricted between $\frac{1}{2}|\mathcal{A}_{\text{train}}|$ and $|\mathcal{A}_{\text{train}}|$, where $|\mathcal{A}_{\text{train}}|$ is the maximum duration of input audio during training (for Whisper this is 30 seconds).

With an upper bound now set on the duration of input segments, the other extreme must be considered: very short segments, which present their distinct set of challenges. Transcribing brief speech segments eliminates the broader context beneficial for modelling speech in challenging scenarios. Moreover, transcribing numerous shorter segments increases total transcription time due to the increased number of forward passes required.

Therefore, we propose a 'merge' operation, performed after 'min-cut', merging

neighbouring segments with aggregate temporal spans less than a maximal duration threshold τ where $\tau \leq |\mathcal{A}_{\text{train}}|$. Empirically we find this to be optimal at $\tau = |\mathcal{A}_{\text{train}}|$, maximizing context during transcription and ensures the distribution of segment durations is closer to that observed during training. Pseudo-code outlining both the min-cut and merge operations can be found in the ArXiv version of the paper.

5.2.3 Whisper Transcription

The resulting speech segments, now with duration approximately equal to the input size of the model, $|s_i| \approx |\mathcal{A}_{\text{train}}| \; \forall i \in N$, and boundaries that do not lie on active speech, can be efficiently transcribed in parallel with Whisper Ω_W, outputting text for each audio segment $\Omega_W : \mathbf{s} \rightarrow \mathcal{T}$. We note that parallel transcription must be performed without conditioning on previous text, since the causal conditioning would otherwise break the independence assumption of each sample in the batch. In practice, we find this restriction to be beneficial, since conditioning on previous text is more prone to hallucination and repetition. We also use the no timestamp decoding method of Whisper.

5.2.4 Forced Phoneme Alignment

For each audio segment s_i and its corresponding text transcription $\mathcal{T}_i$, consisting of a sequence of words $\mathcal{T}_i = [w_0, w_1, ..., w_m]$, our goal is to estimate the start and end time of each word. For this, we leverage a phoneme recognition model, trained to classify the smallest unit of speech distinguishing one word from another, *e.g. the element p in "tap"*. Let $\mathcal{C}$ be the set of phoneme classes in the model $\mathcal{C} = \{c_1, c_2, ..., c_K\}$. Given an input audio segment, a phoneme classifier, takes an audio segment S as input and outputs a logits matrix $L \in \mathbb{R}^{K \times T}$, where T varies depending on the temporal resolution of the phoneme model.

Formally, for each segment, $s_i \in \mathbf{s}$, and its corresponding text $\mathcal{T}_i$: **(1)** Extract the unique set of phoneme classes in the segment text $\mathcal{T}_i$ common to the phoneme model, denoted by $\mathcal{C}_{\mathcal{T}_i} \subset \mathcal{C}$. **(2)** Perform phoneme classification over the input segment s_i, with the classification restricted to $\mathcal{C}_{\mathcal{T}_i}$ classes. **(3)** Apply Dynamic

Time Warping (DTW) on the resulting logits matrix $L_i \in \mathbb{R}^{\mathcal{C}_{\mathcal{T}_i} \times T}$, to obtain the optimal temporal path of phonemes in $\mathcal{T}_i$. **(4)** Obtain start and end times for each word w_i in $\mathcal{T}_i$ by taking the start and end time of the first and last phoneme within the word respectively.

For transcript phonemes not present in the phoneme model's dictionary $\mathcal{C}$, we assign the timestamp from the next nearest phoneme in the transcript. The *for loop* described above can be batch processed in parallel, enabling fast transcription and word-alignment of long-form audio.

5.2.5 Multi-lingual Transcription and Alignment

WhisperX can also be applied to multilingual transcription, with the caveat that (i) the VAD model should be robust to different languages, and (ii) the alignment phoneme model ought to be trained on the language(s) of interest. Multilingual phoneme recognition models [Conneau et al. 2020] are also a suitable option, possibly generalising to languages not seen during training – this would just require an additional mapping from language-independent phonemes to the phonemes of the target language(s).[2]

5.2.6 Translation

Whisper also offers a "translate" mode that allows for translated transriptions from multiple languages into English. The batch VAD-based transcription can also be applied to the translation setting, however phoneme alignment is not possible due to there no longer being a phonetic audio-linguistic alignment between the speech and the translated transcript.

5.2.7 Word-level Timestamps without Phoneme Recognition

We explored the feasibility of extracting word-level timestamps from Whisper directly, without an external phoneme model, to remove the need for phoneme map-

[2]We were unable to find non-English ASR to evaluate multilingual word segmentation, but we show successful qualitative examples in the open-source repository.

ping and reduce inference overhead (in practice we find the alignment overhead is minimal, approx. <10% in speed). Although attempts have been made to infer timestamps from cross-attention scores [Louradour 2023], these methods underperform when compared to our proposed external phoneme alignment approach, as evidenced in Section 5.3.4, and are prone to timestamp inaccuracies.

5.3 Evaluation

Our evaluation addresses the following questions: (1) the effectiveness of *WhisperX* for long-form transcription and word-level segmentation compared to state-of-the-art ASR models (namely Whisper and wav2vec2.0); (2) the benefit of VAD Cut & Merge pre-processing in terms of transcription quality and speed; and (3) the effect of the choice of phoneme model and Whisper model on word segmentation performance.

5.3.1 Datasets

The AMI Meeting Corpus. We used the test set of the AMI-IHM from the AMI Meeting Corpus [Carletta et al. 2006] consisting of 16 audio recordings of meetings. Manually verified word-level alignments are provided for the test set used to evaluate word segmentation performance. **Switchboard-1 Telephone Corpus (SWB).** SWB [Godfrey and Holliman 1993] consists of ∼2,400 hours of speech of telephone conversations. Ground truth transcriptions are provided with manually corrected word alignments. We randomly sub-sampled a set of 100 conversations. To evaluate long-form audio transcription, we report on **TEDLIUM-3** [Hernandez et al. 2018] consisting of 11 TED talks, each 20 minutes in duration, and **Kincaid46** [Kincaid 2018] consisting of various videos sourced from YouTube.

5.3.2 Metrics

For evaluating long-form audio transcription, we report word error rate (**WER**) and transcription speed (**Spd.**). To quantify the amount of repetition and hallucination, we measure insertion error rate (**IER**) and the number of 5-gram word

duplicates within the predicted transcript (**5-Dup.**) respectively. Since this does not evaluate the accuracy of the predicted timestamps, we also evaluate word segmentation metrics, for datasets that have word-level timestamps, jointly evaluating both transcription and timestamp quality. We report the Precision (**Prec.**) and Recall (**Rec.**) where a true positive is where a predicted word segment overlaps with a ground truth word segment within a collar, where both words are an exact string match. For all evaluations we use a collar value of 200 milliseconds to account for differences in annotation and models.

5.3.3 Implementation Details

WhisperX: Unless specified otherwise, we use the default configuration in Table 5.1 for all experiments. **Whisper [Radford et al. 2022b]:** For Whisper-only transcription and word-alignment we inherit the default configuration from Table 5.1, and use the official implementation[3] for inferring word timestamps. **Wav2vec2.0 [Baevski et al. 2020]:** For wav2vec2.0 transcription and word-alignment we use the default settings in Table 5.1 unless specified otherwise. We obtain the various model versions from the official torchaudio [Y.-Y. Yang et al. 2022] repository[4]. Base_960h and Large_960h models were trained on Librispeech [Panayotov et al. 2015] data, whereas the VoxPopuli model was trained on the Voxpopuli [Changhan Wang et al. 2021] corpus. For benchmarking inference speed, all models are measured on an NVIDIA A40 gpu, as multiples of Whisper's speed.

5.3.4 Results

Word Segmentation Performance

Comparing to previous state-of-the-art speech transcription models (Table 5.2), Whisper and wav2vec2.0, we find that *WhisperX* substantially outperforms both in word segmentation benchmarks, WER, and transcription speed. Especially with batched transcription, *WhisperX* even surpasses the speed of the lightweight

[3]https://github.com/openai/whisper/releases/tag/v20230307
[4]https://pytorch.org/audio/stable/pipelines.html#module-torchaudio.pipelines

Table 5.1: Default configuration for WhisperX.

Type	Hyperparameter	Default Value
VAD	Model	`pyannote` [Bredin et al. 2020b]
	Onset threshold	0.767
	Offset threshold	0.377
	Min. duration on	0.136
	Min. duration off	0.067
Whisper	Model version	`large-v2`
	Decoding strategy	greedy
	Condition on previous text	False
Phoneme Model	Architecture	wav2vec2.0
	Model version	`BASE_960H`
	Decoding strategy	greedy

Table 5.2: **State-of-the-art comparison of long-form audio transcription and word segmentation** on the TED-LIUM [Hernandez et al. 2018], Kincaid46 [Kincaid 2018], AMI [Carletta et al. 2006], and SWB [Godfrey and Holliman 1993] corpora. **Spd** denotes transcription speed, **WER** denotes Word Error Rate, **5-Dup** denotes the № 5-gram duplicates, Precision & Recall are calculated with a collar value of 200ms. †Word timestamps from Whisper are not directly available but are inferred via Dynamic Time Warping of the decoded tokens attention scores.

Model	TED-LIUM				Kincaid46			AMI		SWB	
	Spd.↑	WER↓	IER↓	5-Dup.↓	WER↓	IER↓	5-Dup.↓	Prec.↑	Rec.↑	Prec.↑	Rec.↑
wav2vec2.0	10.3×	19.8	8.5	**129**	28.0	5.3	**29**	81.8	45.5	92.9	54.3
Whisper	1.0×	10.5	7.7	221	12.5	3.2	131	78.9	52.1	85.4	62.8
WhisperX (ours)	**11.8**×	**9.7**	**6.7**	189	**11.8**	**2.2**	75	**84.1**	**60.3**	**93.2**	**65.4**

Table 5.3: **Effect of WhisperX's VAD Cut & Merge and batched transcription on long-form audio transcription** on the TED-LIUM benchmark and AMI corpus. Full audio input corresponds to WhisperX without any VAD pre-processing, VAD-CM$_\tau$ refers to VAD pre-processing with Cut & Merge, where τ is the merge duration threshold in seconds.

Input	Batch Size	TED-LIUM		AMI	
		WER↓	Spd.↑	Prec.↑	Rec.↑
Full audio	1	10.52	1.0×	82.6	53.4
	32	78.78	7.1×	43.2	25.7
VAD-CM$_{15}$	1	9.72	2.1×	84.1	56.0
	32		7.9×		
VAD-CM$_{30}$	1	**9.70**	2.7×	**84.1**	**60.3**
	32		**11.8**×		

wav2vec2 model. However, solely using Whisper for word-level timestamps extraction significantly underperforms in word segmentation precision and recall on both SWB and AMI corpuses, even falling short of wav2vec2.0, a smaller model with less training data. This implies the insufficiency of Whisper's large-scale noisy training data and current architecture for learning accurate word-level timestamps.

Effect of VAD Chunking

Table 5.3 demonstrates the benefits of pre-segmenting audio with VAD and Cut & Merge operations, improving both transcription-only WER and word segmentation precision and recall. Batched transcription without VAD chunking, however, degrades both transcription quality and word segmentation due to boundary effects.

Batched inference with VAD, transcribing each segment independently, provides a nearly twelve-fold speed increase without performance loss, overcoming the limitations of buffered transcription [Radford et al. 2022b]. Batch inference without VAD, using a sliding window, significantly degrades WER due to boundary effects, even with heuristic overlapped chunking as in huggingface[5].

The optimal merge threshold value for Cut & Merge operations τ is found to be the input duration that Whisper was trained on $|\mathcal{A}_{\text{train}}| = 30$, which provides the fastest transcription speed and lowest WER. This confirms that maximum context yields the most accurate transcription.

Hallucination & Repetition

In Table 5.2, we find that *WhisperX* reports the lowest IER on the Kincaid46 and TED-LIUM benchmarks, confirming that the proposed VAD Cut & Merge operations reduce hallucination in Whisper. Further, we find that repetition errors, measuring by counting the total number of 5-gram duplicates per audio, is also reduced by the proposed VAD operations. By removing the reliance on decoded timestamp tokens, and instead using external VAD segment boundaries, *WhisperX* avoids repetitive transcription loops and hallucinating speech during inactivate

[5]https://huggingface.co/openai/whisper-large

Table 5.4: **Effect of whisper model and phoneme model on WhisperX on word segmentation.** Both the choice of whisper and phoneme model has a significant effect on word segmentation performance.

Whisper Model	Phoneme Model	AMI		SWB	
		Prec.	**Rec.**	**Prec.**	**Rec.**
base.en	Base_960h	83.7	58.9	**93.1**	**64.5**
	Large_960h	84.9	56.6	**93.1**	62.9
	VoxPopuli	**87.4**	**60.3**	86.3	60.1
small.en	Base_960h	84.1	59.4	92.9	62.7
	Large_960h	84.6	55.7	**94.0**	**64.9**
	VoxPopuli	**87.7**	**61.2**	84.7	56.3
large-v2	Base_960h	84.1	60.3	93.2	65.4
	Large_960h	84.9	57.1	**93.5**	**65.7**
	VoxPopuli	**87.7**	**61.7**	84.9	58.7

speech regions.

Whilst wav2vec2.0 underperforms in both WER and word segmentation, we find that it is far less prone to repetition errors compared to both Whisper and *WhisperX*. Further work is needed to reduce hallucination and repetition errors.

Effect of Chosen Whisper and Alignment Models

We compare the effect of different Whisper and phoneme recognition models on word segmentation performance across the AMI and SWB corpuses in Table 5.4. Unsurprisingly, we see consistent improvements in both precision and recall when using a larger Whisper model. In contrast, the bigger phoneme model is not necessarily the best and the results are more nuanced. The model trained on the VoxPopuli corpus significantly outperforms other models on AMI, suggesting that there is a higher degree of domain similarity between the two corpora.

The large alignment model does not show consistent gains, suggesting the need for additional supervised training data. Overall the base model trained on LibriSpeech performs consistently well and should be the default alignment model for *WhisperX*.

5.4 Conclusion

To conclude, we propose *WhisperX*, a time-accurate speech recognition system enabling within-audio parallelised transcription. We show that the proposed VAD Cut & Merge preprocessing reduces hallucination and repetition, enabling within-audio batched transcription, resulting in a twelve-fold speed increase without sacrificing transcription quality. Further, we show that the transcribed segments can be forced aligned with a phoneme model, providing accurate word-level segmentations with minimal inference overhead and resulting in time-accurate transcriptions benefitting a range of applications (e.g. subtitling, diarisation etc.). A promising direction for future work is the training of a single-stage ASR system that can efficiently transcribe long-form audio with accurate word-level timestamps.

Acknowledgement This research is funded by the EPSRC VisualAI EP/T028572/1 (M. Bain, T. Han, A. Zisserman) and a Global Korea Scholarship (J. Huh). Finally, the authors would like to thank the numerous open-source contributors and supporters of *WhisperX*.

Chapter 6

AutoAD: Movie Description in Context

The paper has been accepted for publication as a highlight at the Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

AutoAD: Movie Description in Context

Tengda Han[1*] Max Bain[1*]

Arsha Nagrani Gül Varol[1,2] Weidi Xie[1,3]

Andrew Zisserman[1]

[1]Visual Geometry Group, University of Oxford

[2]LIGM, École des Ponts, Univ Gustave Eiffel, CNRS

[3]CMIC, Shanghai Jiao Tong University

https://www.robots.ox.ac.uk/vgg/research/autoad/

Abstract

The objective of this paper is an automatic Audio Description (AD) model that ingests movies and outputs AD in text form. Generating high-quality movie AD is challenging due to the dependency of the descriptions on context, and the limited amount of training data available. In this work, we leverage the power of pretrained foundation models, such as GPT and CLIP, and only train a mapping network that bridges the two models for visually-conditioned text generation. In order to obtain high-quality AD, we make the following four contributions: (i) we incorporate context from the movie clip, AD from previous clips, as well as the subtitles; (ii) we address the lack of training data by pretraining on large-scale datasets, where visual or contextual information is unavailable, e.g. text-only AD without movies or visual captioning datasets without context; (iii) we improve on the currently available AD datasets, by removing label noise in the MAD dataset, and adding character naming information; and (iv) we obtain strong results on the movie AD task compared with previous methods.

∗: equal contribution

Context AD:
As Karen stares groomly out of the window, a man approaches toying with a lighter. She turns her head, and finds Jack standing beside her.

Subtitles:
> Can I buy you a drink?
> Yeah I'd love one. Sit down.

Target AD: He takes the seat opposite, then places his lighter on the table

Figure 6.1: **Movie audio description (AD)** consists of sentences describing movies for the visually impaired. Note how it is heavily influenced by various types of context – the visual frames, the previous AD, and the subtitles of the movie.

6.1 Introduction

That of all the arts, the most important for us is the cinema.

–Vladimir Lenin

One of the long-term aims of computer vision is to understand long-form feature films. There has been steady progress towards this aim with the identification of characters by their face and voice [Bojanowski et al. 2013; A. Brown et al. 2021a; Everingham et al. 2006; Tapaswi et al. 2012b; Q. Huang et al. 2018], the recognition of their actions and inter-actions [Laptev et al. 2008; Marszałek et al. 2009; Patron-Perez et al. 2010; Vondrick et al. 2016], of their relationships [Kukleva et al. 2020a], and 3D pose [Pavlakos et al. 2022]. However, this is still a long way away from story understanding. Movie *Audio Description (AD)*, the narration describing visual elements in movies, provides a means to evaluate current movie understanding capabilities. AD was developed to aid visually impaired audiences, and is typically generated by experienced annotators. The amount of AD on the internet is growing due to more societal support for visually impaired communities and its inclusion is becoming an emerging legal requirement.

AD differs from image or video captioning in several significant respects [Research and Center 2013], bringing its own challenges. First, AD provides dense descriptions of important visual elements *over time*. Second, AD is always provided on

a separate soundtrack to the original audio track and is highly *complementary* to it. It is complementary in two ways: it does not need to provide descriptions of events that can be understood from the soundtrack alone (such as dialogue and ambient sounds), and it is constrained in time to intervals that do not overlap with the dialogue. Third, unlike dense video captioning, AD aims at *storytelling*; therefore, it typically includes factors like a character's name, emotion, and action descriptions.

In this work, our objective is automatic AD generation – a model that takes continuous movie frames as input and outputs AD in text form. Specifically, we generate text given a temporal interval of an AD, and evaluate its quality by comparing with the ground-truth AD. This is a relatively unexplored task in the vision community with previous work targeting ActivityNet videos [Y. Wang et al. 2021], a very different domain to long-term feature films with storylines, and the LSMDC challenge [A. Rohrbach et al. 2017b], where the descriptions and character names are treated separately.

As usual, one of the challenges holding back progress is the lack of suitable training data. Paired image-text or video-text data that is available at scale, such as alt-text [Radford et al. 2021; Sharma et al. 2018a] or stock footage with captions [Bain et al. 2021], does not generalize well to the movie domain [Bain et al. 2022]. However, collecting high-quality data for movie understanding is also difficult. Researchers have tried to hire human annotators to describe video clips [X. Chen et al. 2015; J. Xu et al. 2016; Krishna et al. 2017a] but this does not scale well. Movie scripts, books and plots have also been used as learning signals [Bojanowski et al. 2013; Sigurdsson et al. 2016b; Yukun Zhu et al. 2015] but they do not ground on vision closely and are limited in number.

In this paper we address the AD and training data challenges by – Spoiler Alert – developing a model that uses temporal context together with a visually conditioned generative language model, while providing new and cleaner sources of training data. To achieve this, we leverage the strength of large-scale language models (LLMs), like GPT [Radford et al. 2019], and vision-language models, like CLIP [Radford et al. 2021], and integrate them into a video captioning pipeline that can be effectively trained with AD data.

Our contributions are the following: (i) inspired by ClipCap [Mokady et al. 2021] we propose a model that is effectively able to leverage both temporal context (from previously generated AD) and dialogue context (in particular the names of characters) to improve AD generation. This is done by bridging foundation models with lightweight adapters to integrate both types of context; (ii) we address the lack of large-scale training data for AD by pretraining components of our model on partially missing data which are typically available in large quantities e.g. text-only AD without movie frames, or visual captioning datasets without multiple sentences as context; (iii) we propose an automatic pipeline for collecting AD narrations at scale using speaker-based separation; and finally (iv) we show promising results on automatic AD, as seen from both qualitative and quantitative evaluations, and also achieve impressive *zero-shot* results on the LSMDC multi-description benchmark comparable to the finetuned state-of-the-art.

6.2 Related Works

Image Captioning. Image captioning is a long-standing problem in computer vision [X. Chen and Zitnick 2014; Jeffrey Donahue et al. 2015; Karpathy and Fei-Fei 2015; Kiros et al. 2014; Lu et al. 2018; P. Anderson et al. 2018; X. Chen et al. 2015]. Early pioneering works learn to associate images and words within a limited vocabulary and a set of images [Barnard and Forsyth 2001; Barnard et al. 2003; Lavrenko et al. 2003]. Large-scale image captioning datasets have been collected by scraping images from the internet and their corresponding alt-texts with quality filters as a post-processing [Sharma et al. 2018a]. In doing so, strong joint image-text representations can be learned [Radford et al. 2021], and image captioning from raw pixels, with impressive results [Jiahui Yu et al. 2022; Junnan Li et al. 2022]. Recent work [Mokady et al. 2021; Nukrai et al. 2022] learns a bridge between strong joint image-text representations (CLIP) and the natural language representation (GPT-2) for image captioning, obtaining promising results that generalise well across domains. In this work, we extend this approach to perform automatic AD from videos.

Video Captioning. Video captioning presents additional challenges due to the lack of quality large-scale video-text data and increased complexity from the tem-

poral axis. Early video caption datasets [D. Chen and Dolan 2011; J. Xu et al. 2016] adopt manual annotations, a far from scalable collection method. ASR (automated speech recognition) from YouTube instructional videos is collected at scale for video-language datasets [Miech et al. 2019], but contains high levels of noise due to the weak correspondence between the narration and visual content. VideoCC [Nagrani et al. 2022] transfers captions from images to videos, but this method is still limited by the existing seed image captioning dataset used. Earlier video captioning models lack generalisation capabilities due to limited training data [Venugopalan et al. 2015; Park et al. 2019]. Some recent methods [Seo et al. 2022; G. Huang et al. 2020; H. Luo et al. 2020] train on ASR from the HowTo100M dataset, while others expand image-text representations [M. Tang et al. 2021] to multiple frames.

A task more related to AD is that of dense video captioning [Krishna et al. 2017a], which involves producing a number of captions and their corresponding grounded timestamps in the video. To enrich inter-task interactions, recent works for this task [Chadha et al. 2021; Shaoxiang Chen and Jiang 2021; C. Deng et al. 2021; Y. Li et al. 2018; Mun et al. 2019; Rahman et al. 2019; Shen et al. 2017; Shi et al. 2019; Jingwen Wang et al. 2018; T. Wang et al. 2021; L. Zhou et al. 2018c] jointly train both a captioning and localization module. Our task differs in that the captions are: made with the intent to aid storytelling; specific to the movie domain; and complementary to the audio track.

Visual Storytelling. Most similar in vein to the AD task is visual storytelling [T.-H. Huang et al. 2016; Junnan Li et al. 2020; Ravi et al. 2021], in which the goal is to generate coherent sentences for a sequence of video clips or images. LSMDC [A. Rohrbach et al. 2017b] proposes the multi-description task of generating captions for a set of clips from a movie, with character names anonymized. In contrast, movie AD takes as input a continuous long video and describes the visual happenings complementary to the story, characters, dialogue and audio. Most similar to our model is TPAM [Y. Yu et al. 2021] which prompts a frozen GPT-2 with local visual features. Ours differs in that: (i) it is not restricted to local visual context but rather global by recurrently conditioning on previous outputs; and (ii) we additionally pretrain GPT on in-domain text-only AD data.

Movie Understanding. Previous works investigate storyline understanding by

aligning movies to additional data sources such as plots [Xiong et al. 2019; Yidan Sun et al. 2022], books [Tapaswi et al. 2015a; Yukun Zhu et al. 2015], scripts [Papalampidi et al. 2019], and YouTube summaries [Bain et al. 2020a]. However, these sources are limited in number and often do not closely relate to the visual elements in the frame. Using existing movie AD as the data source for videos is an emergent direction for movie understanding. LSMDC [A. Rohrbach et al. 2017b], M-VAD dataset [Torabi et al. 2015] and MPII-MD [A. Rohrbach et al. 2017b], gather AD and scripts from movies to provide captions for short video clips, several seconds in duration. QuerYD [Oncescu et al. 2021] provides high-quality textual descriptions for longer videos by scraping AD from YouDescribe [Research and Center 2013], an online community of AD contributors. Recently, the MAD dataset [Soldan et al. 2022] collects movie AD at scale to provide dense textual annotations for movies with a focus on visual grounding task.

Prompt Tuning and Adapters. Originally for language modelling, prompt tuning is a lightweight approach to adapt a pretrained model to perform a downstream task. Early works [T. Brown et al. 2020; Lester et al. 2021; X. L. Li and Liang 2021; Ju et al. 2022] learn prompt vectors that are shared within the targeted dataset and task. A similar line of works to ours is *visual-conditioned* prompt tuning, in which the prompt vectors are conditioned on the visual inputs. Visual-conditioned prompts are used for adapting pretrained image-language models [Bahng et al. 2022; M. Jia et al. 2022], and for few-shot learning [Tsimpoukelli et al. 2021; Alayrac et al. 2022]. Training lightweight feature adapters between pretrained vision and text encoders is another approach to adapt pretrained models [P. Gao et al. 2021; R. Zhang et al. 2021]. The adapter layers can also be inserted into the pretrained language model in an interleaved way [A. Yang et al. 2022]. Our work adopts prompt tuning in order to condition a language generation model on visual information (frames), and textual context (subtitles and previous AD).

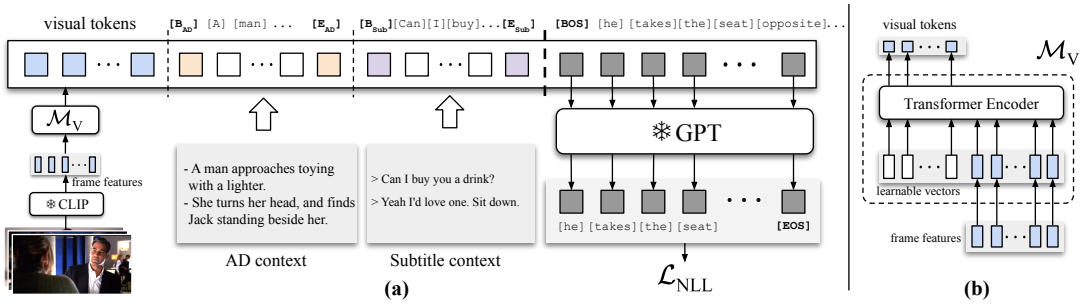


Figure 6.2: **(a) Overview of AutoAD:** AutoAD consists of a *frozen* visual encoder (CLIP) and a *frozen* LLM (GPT) for generating captions. We introduce a lightweight mapping network to map CLIP features into visual tokens, which are then combined with previous AD context and subtitle context, before being fed into the GPT model. $\mathcal{M}_V$ refers to the visual mapping network, $[B_*]$ and $[E_*]$ denote the learnable special tokens for contextual AD and subtitle sequences. **(b) Detail of the visual mapping network:** A transformer encoder takes as input multiple frame features and outputs a few visual tokens which are further fed to a text generation model.

6.3 Method

Given a long-form movie $\mathcal{V}$ segmented into multiple short clips $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$, our goal is to generate the audio description (AD) in text form for every movie clip. Note that each movie clip is cut from the raw movie based on the timestamp $[t_{\text{start}}, t_{\text{end}}]$ given by the AD annotation. Specifically, for the i-th movie clip consisting of multiple frames $\mathbf{x}_i = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_N\}$, we aim to produce text $\mathcal{T}_i$ that describes the visual elements in such a way that helps the visually impaired follow the storyline. To this purpose, an ideal AD generation system must be able to exploit the full contextual information leading up to the i-th movie clip. One method for this, which we adopt, is to use previous AD $\mathcal{T}_{t<i}$ and subtitles $\mathcal{S}_{t\leq i}$ to generate the text $\mathcal{T}_i$. In the following sections, we first give an overview of our visual captioning pipeline with prompt tuning (Sec. 6.3.1), followed by our contextual components (Sec. 6.3.2), and finally the pretraining methods with partial data (Sec. 6.3.3).

6.3.1 Visual Captioning with Prompt Tuning

In order to describe our method, we first present the typical pipeline for an image captioning model, and then detail how we extend this to ingest multiple frames and additional text context. Given an image-caption pair $\{\mathcal{I}_i, \mathcal{C}_i\}$, where the cap-

tion consists of a sequence of language tokens $\mathcal{C}_i = \{c_1, c_2, ..., c_k\}$, the standard objective of an image captioning model is to generate text tokens $\hat{\mathcal{C}}_i$ that are close to the target $\mathcal{C}_i$. Technically, the captioning models are trained to maximize the joint probability of predicting the ground-truth language tokens, or equivalently minimize the following negative log-likelihood (NLL) loss,

$$\mathcal{L}_{\text{NLL}} = -\log p_\theta(\mathcal{C}_i|\mathbf{h}_{\mathcal{I}_i}) = -\log p_\theta(c_1, c_2, ..., c_k|\mathbf{h}_{\mathcal{I}_i})$$

where θ denotes the parameters of the model, and $\mathbf{h}_{\mathcal{I}_i}$ denotes the extracted image features of $\mathcal{I}_i$. Previous works like ClipCap [Mokady et al. 2021] fit a powerful text generation model and visual encoding model into this image captioning pipeline. Specifically, strong visual encoding models, such as CLIP [Radford et al. 2021], are used to extract the visual features from the input image $\mathbf{z}_i = f_{\text{CLIP}}(\mathcal{I}_i)$, then a visual mapping network $\mathcal{M}_{\text{V}}$ is trained to map the visual features to 'prompt vectors' that adapt to the text generation model, $\mathbf{h}_{\mathcal{I}_i} = \mathcal{M}_{\text{V}}(\mathbf{z}_i)$. Finally these prompt vectors $\mathbf{h}_{\mathcal{I}_i}$ are fed to a pretrained text generation model, such as GPT [Radford et al. 2019], for the captioning task. We adapt this visual captioning pipeline, which uses pretrained feature extractor CLIP and langauge model GPT, for movie AD generation and propose key components that support contextual understanding.

6.3.2 Benefiting from Temporal Context

Here, we describe how we extend this single-frame captioning model to include different forms of context, including multiple frames, previous AD text, and subtitles. Compared to image captioning where the annotation describes 'what is in the image', movie AD describes the visual happenings in the scene that are relevant to the broader story – often centered around events, characters and the interactions between them. Factors like these cannot be accurately described from a static image alone and therefore a successful automatic AD system must utilize the context of prior events and character interactions.

To tackle these temporal dependencies, we propose to include three components to incorporate the essential contextual information from movies: (i) immediate visual context in the current movie clip (multiple frames), (ii) the previous movie AD, and (iii) the movie subtitles. The architecture of our model is shown in Fig. 6.2.

Multiple frames (immediate visual context). In contrast to the image captioning method, the visual mapping network $\mathcal{M}_{\text{V}}$ takes as input multiple frame features from the current movie clip $\mathbf{x}_i$ rather than a single image feature, and outputs prompt vectors for the movie clip,

$$\mathbf{h}_{\mathbf{x}_i} = \mathcal{M}_{\text{V}}(\{\mathbf{z}_1, ..., \mathbf{z}_N\}); \quad \mathbf{z}_i = f_{\text{CLIP}}(\mathcal{I}_i).$$

In detail, the mapping network consists of a multi-layer transformer encoder that enables modelling temporal relations among multiple frame features, as shown in Fig 6.2.

Previous AD text. The sequence of events leading up to the present contain contextual information which are crucial for generating AD of current scene that helps the viewer follow the story. We input this contextual knowledge to our model in the form of the past ADs. Specifically, our model takes the past K movie ADs $\{\mathcal{T}_{i-K}, ..., \mathcal{T}_{i-1}\}$ to generate the AD for the current clip. The past movie ADs are a few sentences, which are first concatenated into a single paragraph, then tokenized and converted to a sequence of word embeddings. Inspired by the design of special tokens in language models, we wrap the context AD embeddings with *learnable* special tokens to indicate the beginning and end of the AD sequence. Formally, the contextual AD embedding is a sequence,

$$\mathbf{h}_{\text{AD}} = [\texttt{B}_{\text{AD}}; \mathbf{h}_{\mathcal{T}_{i-K}}; ...; \mathbf{h}_{\mathcal{T}_{i-1}}; \texttt{E}_{\text{AD}}] \tag{6.1}$$

where $\texttt{B}_{\text{AD}}$ and $\texttt{E}_{\text{AD}}$ are the learnable special tokens indicating the beginning and end, the symbol ';' denotes concatenation, and $\mathbf{h}_{\mathcal{T}_j} \in \mathbb{R}^{n \times C}$ denotes the word embedding of the j-th movie ADs.

Previous subtitles. Our model also takes the movie subtitles as additional contextual information, which can be sourced either from the official movie metadata or automatically transcribed with an ASR model. The character dialogues, contained with the subtitles, provide complementary information to movie description, including the character names, relationships and emotions. Similar to the context ADs, we concatenate multiple subtitle sentences into a single paragraph and wrap them with learnable special tokens. Practically, since the timing of movie AD does

Manual Veri.*	She stands and the little warrior takes in her size, about twice his own.	As she steps past him, he defensively grips his spear	Leia sits on a moss covered log.
MAD-v1	Angola, she stands in the Little Warrior, takes in her size about twice his own.	As she steps past me. Defensively grips his spear.	I'm not gon na. Leah sits on a Moss covered log.
MAD-v2 (ours)	She stands and the little warrior takes in her size about twice his own.	As she steps past him, he defensively grips his spear.	Leia sits on a moss-covered log.

Figure 6.3: **Qualitative comparison of MAD annotations.** We compare the original MAD-v1 [Soldan et al. 2022] and our proposed MAD-v2. Note MAD-v1's erroneous transcriptions of AD and dialogue leakage (highlighted in red text). The samples are taken from Star Wars VI: Return of the Jedi (1983) [Marquand 1983]. *We verify this example by manually transcribing the AD narration from the audio track.

not overlap with the subtitles, we take the most recent L subtitles within a certain time range as the context,

$$\mathbf{h}_{\text{Sub}} = [\texttt{B}_{\text{Sub}}; \mathbf{h}_{\mathcal{S}_{i-L}}; ...; \mathbf{h}_{\mathcal{S}_{i-1}}; \texttt{E}_{\text{Sub}}]$$

Due to the weak correlation between the subtitles and the visual elements in the scene, we also experiment with a variant that only encodes the character names occurring in the recent subtitles.

Summary. Overall, the movie AD for the current movie clip $\mathcal{T}_{\mathbf{x}_i}$ is generated by conditioning on all the previously described visual and contextual information using a pretrained GPT. The conditional information is fed to GPT as prompt vectors as shown in Fig. 6.2. The model is trained with NLL loss,

$$\mathcal{L}_{\text{NLL}} = -\log p_\Theta(\mathcal{T}_{\mathbf{x}_i} | \mathbf{h}_{\mathbf{x}_i}, \mathbf{h}_{\text{AD}}, \mathbf{h}_{\text{Sub}}). \tag{6.2}$$

During training, we input the ground-truth past AD. During inference, we experiment with two methods to incorporate the past AD: an **oracle** setting where the *ground-truth* past ADs are used in Eq. 6.2 to generate the current AD, and a **recurrent** setting where the *predicted* past ADs are used instead.

6.3.3 Pretraining with Partial Data

A major challenge for generating AD is the lack of training data, since the model requires the corresponding visual, textual and contextual data to all be jointly trained. However since our model is modular, components of it can be pretrained

with *partial data* – when a certain type of data is missing, the remaining modules can still be trained. We experiment with partial-data pretraining under two settings: visual-only pretraining and AD-only pretraining.

Visual-only Pretraining. In the absence of contextual data, the visual mapping network can be pretrained with abundant image captioning or (short) video captioning datasets. In this case, the context modules (both contextual AD and subtitles) are deactivated. The training objective of Eq. 6.2 is turned into $\mathcal{L} = -\log p_\Theta(\mathcal{T}_{\mathbf{x}_i}|\mathbf{h}_{\mathbf{x}_i})$ for visual-only pretraining. Note that the language model is kept frozen here since we find image/video captioning datasets have a clear domain gap with movie AD in both the vision and text modalities.

AD-only Pretraining. Movie AD datasets with corresponding visual information (e.g. frames or frame features) are limited at scale due to potential copyright issues. However, abundant *text-only* movie ADs are available online as described in Sec. 6.5. In the absence of visual data, the contextual AD module and the language model can still be pretrained. The training objective in this case becomes $\mathcal{L} = -\log p_\Theta(\mathcal{T}_{\mathbf{x}_i}|\mathbf{h}_{\text{AD}})$, which is similar to training a story completion objective [Mostafazadeh et al. 2016] by finetuning GPT on *text-only* movie AD data but with a few additional special tokens. This text-only movie AD pretraining is also related to [Gururangan et al. 2020], which shows a second stage of language model pretraining on in-domain data improves downstream performance.

6.4 Denoising MAD Dataset

Our main objective is to generate movie audio descriptions. For this goal, the model is trained on the MAD training set [Soldan et al. 2022], a dataset of AD caption-video clip pairs from 488 movies. MAD provides the video data in the form of CLIP visual features in order to avoid copyright restrictions. The AD annotations for each movie are automatically collected from AudioVault[1], a large open-source database of audio files containing the full-length original movie track mixed with the AD narrator's voice. The MAD authors transcribe a subset of this data using ASR, and also have access to the official DVD subtitles. Their

[1] https://audiovault.net

automated method then uses *text-based* speaker separation of the transcribed audio by using subtitles to know when dialogue is present, and assuming all other speech is AD.

This however introduces *significant noise* because (i) the outdated ASR model results in erroneous transcriptions; and (ii) official DVD subtitles are not exhaustive of all speech in the movie and thus such a method frequently misidentifies character dialogue as AD narration (an example is provided in Fig. 6.3). Further, obtaining official subtitles from DVDs presents additional challenges when collecting this data at scale.

We propose an improved automated data collection method for AD, requiring only the audio track as input (no DVD subtitles), that tackles both issues by using *audio-based* speaker separation and an improved ASR model. We then use this method to collect improved annotations for the MAD dataset. Briefly, taking the mixed audio containing both AD narrations and original movie sound track as input, our automated AD collection pipeline contains five stages: (1) speech recognition using WhisperX [Bain et al. 2023] resulting in punctuated transcriptions with word-level timestamps; (2) sentence tokenization using nltk [Bird 2006] to provide sentence-level segmentation; (3) speaker diarization [Bredin et al. 2020b; Bredin and Laurent 2021] to assign speaker labels to each sentence, where the sentence timestamps are used as oracle voice-activity-detection (VAD); (4) labelling the speaker ID of the AD narrator by selecting the cluster with the lowest proportion of first-person pronouns (e.g. 'I' and 'we'); and finally (5) synchronization of the segment timestamps with the visual features by comparing audio. Further details are in the Appendix.

Henceforth we refer to the original MAD annotation [Soldan et al. 2022] as **MAD-v1** and our new denoised annotations as **MAD-v2**. A qualitative comparison is shown in Fig. 6.5, we find that our MAD-v2 is much more robust and contains less errors and less character dialogue leakage. Both LSMDC and MAD-v1 post-process their annotations by replacing character names in the annotations with 'someone' via entity recognition, and release both variants of annotations which we refer to as **`Named`** and **`Unnamed`**. Similarly, we propose two variants of our denoised annotations:

MAD-v2-`Named`: It contains the raw collected AD narrations *without* any post-

Dataset	Total movies	Total duration (hrs)	Total AD captions	Subtitles	Visual Features
QueryD [Oncescu et al. 2021]	-	207	31K	✗	✓
LSMDC [A. Rohrbach et al. 2017b]	200	147	128K	✗	✓
MAD-v1 [Soldan et al. 2022]	488	892	280K	✓	✓
MAD-v2 (ours)	488	892	264K	✓	✓
AudioVault (ours)	7,057	12,510	3.3M	✓	✗

Table 6.1: **Statistics of Audio Description datasets.** We report relevant statistics to compare our MAD-v2 and Audiovault datasets.

processing on the character names.

MAD-v2-`Unnamed`: Following the character name anonymisation performed in earlier works, we identify character names using a Named Entity Recognition (NER) model [Polle n.d.] and replace them with 'someone'.

6.5 Partial Pretraining with AudioVault Dataset

Paired AD and corresponding visual data are difficult to obtain especially due to movie copyrights, whereas a large number of movie ADs audio tracks are available online for free (e.g. AudioVault). To demonstrate the effect of partial pretraining in Sec. 6.3.3, we collect a large-scale *text-only* movie AD dataset from AudioVault. In detail, we source mixed audio files from over 7,000 movies from AudioVault that are not included in MAD-v1, and use a denoising pipeline similar to that described in Sec. 6.4 to obtain the movie ADs (detailed in Appendix). Additionally we obtain a proxy for the movie subtitles by assuming the ASR from all the non-AD speakers are the characters' dialogues. To ensure no test-time leakage, we remove all movies present in either LSMDC or MAD from the dataset.

Overall, our AudioVault dataset is an order of magnitude larger than prior AD datasets (see Table 6.1), from which we provide two sets of data:

AudioVault-AD. The AD narrations from AudioVault and their corresponding timestamps within each movie, totalling 3.3 million AD utterances.

AudioVault-Sub. The subtitles data from AudioVault and their corresponding timestamps within each movie, totalling 8.7 million subtitle utterances.

6.6 Experiments

In this section we first outline the experimental details for the AD task, the datasets used for training & testing, the architectural details, and the evaluation metrics (Sec. 6.6.1). We then report results and discuss the findings, perform ablations on our model, and compare to prior works (Sec. 6.6.2).

6.6.1 Implementation Details

Datasets

Training Datasets. **CC3M** (Conceptual Caption) [Sharma et al. 2018a] is a large image alt-text dataset that contains 3.3M web images. **WebVid** [Bain et al. 2021] is a large video-caption dataset that contains 2.5M short stock footage videos. We use them for the partial-data pretraining for visual modules. Additionally, we use our **AudioVault-AD** to pretrain the textual modules, as described in Sec. 6.3.3. For the main Movie AD task, we train with original **MAD-v1** and our cleaned version **MAD-v2**, detailed in Sec. 6.4.

Test Datasets. **LSMDC** [A. Rohrbach et al. 2017b] contains 118K short video clips with descriptions from 202 movies, of which 182 of them are public. The original MAD-val&test split inherits LSMDC annotations after filtering out 20 lower-quality movies, resulting in 162 movies from all the LSMDC-train/val/test splits. We propose an evaluation split named **MAD-eval** by further excluding LSMDC train&test movies from these 162 movies, which gives a subset consisting of 10 movies. The reason is twofold: (i) LSMDC-train is commonly used by other works as training data, and (ii) the character names of LSMDC-test are not public. Similarly, we use both **MAD-eval-`Named`** and **MAD-eval-`Unnamed`** versions. The 'Unnamed' version corresponds to the standard LSMDC annotation style – where the characters' titles and names in the descriptions are replaced by the word 'someone'; the 'Named' version is constructed from the original character names provided by LSMDC. Additionally, subtitles are not provided with MAD-val/test or LSMDC, so we transcribe them from the full-length audio tracks using WhisperX [Bain et al. 2023].

Architecture

For **visual features**, we use the CLIP ViT-B-32 model [Radford et al. 2021], which is a 12-layer transformer encoder that outputs 1×512 feature vectors for each input frame. These features are provided by the MAD dataset. For the **visual mapping network**, we use a 2-layer transformer encoder with 8 attention heads and 512 hidden dimensions, followed by a linear projection layer that projects 512-d features into 768-d. We use ten prompt vectors. For the **language model**, we use GPT-2 [Radford et al. 2019], specifically the version from HuggingFace. The GPT-2 model takes as input 768-d token embeddings, passes through a 12-layer transformer with a causal attention map, and outputs the next token embedding for every input token. We limit the generated number of tokens to 36, since most movie ADs are less than 36 tokens. The GPT-2 is frozen in most of our experiments unless otherwise stated. Each special token (e.g. $\texttt{B}_{\text{AD}}$) is a learnable 768-d vector. We take at most 64 past AD tokens and 32 subtitle tokens, and short text samples are padded. Specifically for subtitles, we take the most recent four dialogues within a one-minute time window.

Training and Inference Details

On the MAD-v1 and MAD-v2 datasets, we use a batch size of 8 sequences, each of which contains 16 consecutive video-AD pairs from a movie. Overall that gives 8×16 video-AD pairs for every batch. From each video clip, 8 frame features are uniformly sampled. By default, the model is trained for 10 epochs. One epoch means the model has seen *all* the audio descriptions once. Additional implementation details are in the Appendix.

We use the AdamW optimizer [Loshchilov and Hutter 2017] and a cosine-decay learning rate schedule with a linear warm-up. The starting learning rate is 10^{-4} and is decayed to 0. For each experiment, we use a single Nvidia A-40 for training. For text generation, greedy search and beam search are commonly used sampling methods. We stop the text generation when a full stop mark is predicted, otherwise we limit the sequence length to 67 tokens. We use beam search with a beam size of 5 and mainly report results by the top-1 beam-searched outputs, since beam search performs slightly better than greedy search on multiple scenarios. Note

Temporal Context	Partial Data Pretrain	R-L	C	S	BertS
None (1 frame)	None	7.1	4.0	1.0	13.2
V (8 frames)	None	9.3	6.7	2.4	15.6
	CC3M [Sharma et al. 2018a]	9.9	8.4	2.4	16.8
	WV [Bain et al. 2021]	9.9	10.0	2.0	17.3
V+AD	None	11.1 (13.3)	12.6 (17.8)	5.1 (5.8)	18.6 (22.1)
	AV-AD	12.1 (13.9)	14.1 (19.0)	4.2 (4.8)	23.0 (23.7)
	AV-AD, WV	11.9 (13.9)	14.3 (21.9)	4.4 (4.8)	24.2 (23.8)
V+AD+Sub	AV-AD, WV	11.3	13.3	4.7	22.2
V+AD+SubN*	AV-AD, WV	11.9	14.2	5.1	23.6

Table 6.2: **Ablative experiments of our AD captioning method.** We ablate our model with different types of temporal context and partial pretraining. All models are trained on MAD-v2-`Named` and evaluated on MAD-eval-`Named`. For models with AD context we report recurrent results with oracle in parentheses. 'V' refers to visual context by taking multi-frame inputs, 'WV' refers to WebVid2M dataset, 'AV-AD' here refers to our partial-data pretraining with text-only AudioVault-AD dataset. *'SubN' denotes the variant of subtitle module that only takes names as input.

that under the 'recurrent' setting, we feed the past greedy-searched text outputs to the model to generate the current AD, which we find gives more stable results.

Evaluation Metrics

To evaluate the quality of text compared with the ground-truth, we use classic metrics including ROUGE-L [C.-Y. Lin 2004] (**R-L**), CIDEr [Vedantam et al. 2015] (**C**) and SPICE [P. Anderson et al. 2016] (**S**). We also report BertScore [T. Zhang et al. 2020] (**BertS**), which evaluates word matching between a candidate sentence and reference sentence with pretrained BERT embeddings. A higher value indicates better text generation compared with the ground-truth.

6.6.2 Experiments on Movie Audio Descriptions

Effect of Temporal Context. In Table 6.2 we show that visual context from multiple frames brings a clear gain for the AD task (C 6.7 vs 4.0). AD context provides a consistent performance improvement under both oracle (C 17.8 vs 6.7) and recurrent settings (C 12.6 vs 6.7). Note that we find feeding AD context as text tokens works better than training a textual feature mapping network, we conjecture the ADs in their original text form carry the most key information like

MAD Train Set		MAD-eval-Unnamed				MAD-eval-Named			
		R-L	C	S	BertS	R-L	C	S	BertS
v1	Unnamed	15.1	12.7	9.5	22.4	12.7	15.9	4.7	22.0
	Named	11.3	10.9	3.0	24.0	12.8	17.0	5.2	21.8
v2	Unnamed	**15.9**	**14.5**	**10.5**	**26.7**	12.9	18.0	4.7	22.0
	Named	11.4	10.0	3.1	22.5	**13.3**	**17.8**	**5.8**	**22.1**

Table 6.3: **Effect of denoising MAD training data annotation.** We train a model with 6 contextual ADs on MAD-v1 [Soldan et al. 2022] or MAD-v2 sources without any pretraining. The model is evaluated on both the **Named** and **Unnamed** versions of MAD-eval under the **oracle** setting. Cross-domain testing results (when the model is trained and tested on different types of annotations) are provided for reference and marked in gray.

Methods	Pretraining Data	R-L	C	S	BertS
ClipCap [Mokady et al. 2021]	CC3M	8.5	4.4	1.1	11.8
CapDec* [Nukrai et al. 2022]	AV-AD	8.2	6.7	1.4	14.3
AutoAD (ours)	AV-AD	**12.1**	14.1	4.2	23.0
AutoAD (ours)	AV-AD & WebVid	11.9	**14.3**	**4.4**	**24.2**

Table 6.4: Compared with other works on movie AD generation task on MAD-v2. We obtain results from other methods by finetuning their models on MAD-v2-Named dataset, and evaluated on MAD-eval-Named. *CapDec [Nukrai et al. 2022] proposes text-only pretraining to adapt the style for text generation, we pretrained their model on the text-only AudioVault-AD dataset then applied it to MAD-v2.

Methods	Paired Training Data	C	M
Baseline [Park et al. 2019]	LSMDC	11.9	8.3
TAPM [Y. Yu et al. 2021]	LSMDC	15.4	**8.4**
AutoAD (ours)	MAD-v2-Unnamed	16.7	7.4
AutoAD (ours)	MAD-v2-Unnamed & LSMDC	**17.5**	7.5

Table 6.5: **Results on the LSMDC 2019 Multi-Sentence Description public test set.** We report our method with different amounts of training data and without subtitles for comparison under similar settings. Official challenge metrics (CIDEr and METEOR) are reported with the 'sentence' setting as described in [A. Rohrbach et al. 2015b; Y. Yu et al. 2021].

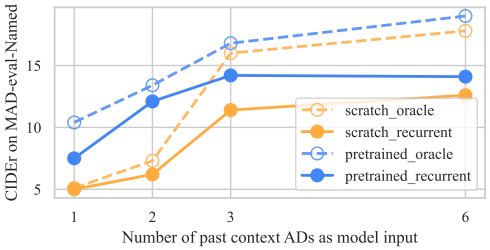


Figure 6.4: **Effect of the length of context AD.** We use the model 'V+AD' in Table 6.2, and train with different number of past AD sentences. 'scratch' indicates no partial-data pretraining; 'pretrained' refers to pretraining with text-only AudioVault-AD.

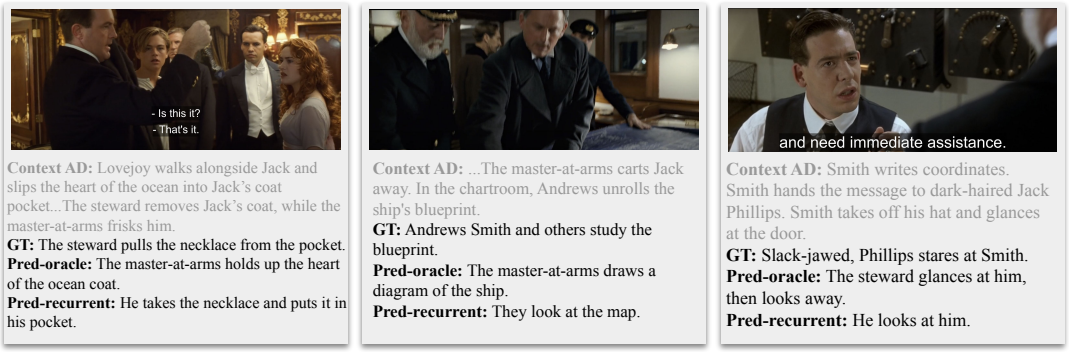


Figure 6.5: **Qualitative examples of automatically generated AD by AutoAD.** We highlight AD predictions under both the oracle and recurrent settings. Previous AD context is shown in gray. For ease of visualisation, a single frame from each movie clip is shown with subtitles overlaid. Samples are taken from Titanic (1997) [Cameron 1997].

the names and places. However, subtitle context provides no gain for our model (C 13.3 vs 14.3) under the recurrent setting, which we attribute to the very weak correspondence between the visual elements in the scene and the character dialogue. When the subtitles are filtered and contain only character names (denoted as 'SubN'), they provide a slight performance gain (C 14.2 vs 13.3). Since the subtitles used are without speaker identities, the model may struggle to know which character in the frame spoke each subtitle. Overcoming these challenges will be considered in future work.

Effect of MAD data cleaning. Table 6.3 demonstrates the benefit of our MAD v2 annotations over v1, confirming the qualitative findings. Training the AD model with context on v2 outperforms training on v1 under all settings (both named and unnamed) by a significant margin. Since the v2 annotations are fewer in number than MAD-v1, this suggests they are indeed less noisy and result in AD

captioning models with improved performance.

Effect of Pretraining with Partial Data. In Table 6.2, we find that **visual-only pretraining** on open-domain vision-text data provides clear gains (CIDEr 8.4 vs 6.7 for CC3M, and 10.0 vs 6.7 for WebVid). But considering the size of visual samples, the improvement is not data-efficient. We attribute this to the large domain gap between movie AD and classical visual caption annotations like CC3M or WebVid2M. The **text-only pretraining** of our model also improves performance. For the recurrent AD context model, AudioVault-AD pretraining increases CIDEr from 12.6 to 14.1, which indicates the great importance of adapting to the text style and context. The combination of the visual module after visual-only pretraining (WebVid) and the textual modules after text-only pretraining (AV-AD) gives a further performance gain (C 21.9 vs 19.0 for the oracle setting, and 14.3 vs 14.1 for recurrent).

Length of Context. In Figure 6.4 we show the effect of varying the number of context ADs given to the model. Longer AD context improves performance almost consistently across all settings, but it brings extra computational cost due to the quadratic complexity of the attention operation in GPT-2. Note that we experiment with at most 6 contextual AD sentences, which is equivalent to about 70-word embeddings in Eq. 6.1. The trend for the recurrent setting flattens when the context ADs are longer than 3 sentences, which is probably due to the limited power of processing long context for the GPT2 model.

Qualitative Results

Fig. 6.5 shows qualitative examples of our model. Under the oracle setting, the model can use the character identities easily from the past ground-truth AD (e.g. “master-at-arms”). Whereas under the recurrent setting, the model can only learn names from the subtitles but names appear very sparsely in subtitles, therefore the model mostly predicts pronouns (e.g. “he”, “they”) but still gets the actions (“looks”) or objects (“necklace”) correct.

6.6.3 Comparison with Other Works

In Table 6.4, we compare our method with previous visual captioning methods. Note that since the MAD dataset only releases the CLIP visual features, rather than the movie frames, our comparison is limited to methods that build on frozen CLIP features. We show a clear performance improvement compared to ClipCap [Mokady et al. 2021] and CapDec [Nukrai et al. 2022], for the latter the language model is also adapted to the movie AD domain by text-only pretraining. The results highlight the importance of context for movie AD.

In Table 6.5, we adapt our method to the Multi-Sentence Description task on LSMDC, in which the model takes five consecutive clips and generates five corresponding descriptions. Since the task is performed on the *unnamed* annotations, we finetune our best model in Table 6.4 with varying 0-4 context ADs as input on MAD-v2-`Unnamed` dataset and test with the *recurrent* setting. To make minimal changes, our model still takes a single clip feature at each step, whereas previous methods take all five clips together for movie description. Despite this disadvantage, we obtain competitive results on this task even without using the *manually-cleaned* LSMDC training set (C 16.7 vs 15.4), effectively *zero-shot*. The performance of the model can be further improved by additionally training on LSMDC data.

6.7 Conclusion and Future Work

This paper focuses on the automatic generation of movie AD for a given time interval, and has made significant progress. We propose an AutoAD pipeline that incorporates contextual information. Additionally, we demonstrate the effectiveness of partial-data pretraining, a technique that could be widely applicable when full data is difficult to obtain. Further, we clean up the previous MAD dataset and collect a new text-only movie AD dataset as a pretraining resource. However, a clear limitation of this AutoAD pipeline is character naming – referencing *who* is doing *what*, a necessary ingredient for story-coherent movie AD. Additionally, future work could tackle the problem of *when* to generate AD, instead of relying on the annotated AD timestamps.

Acknowledgements. We thank Mattia Soldan for helping with the MAD dataset, Anna Rohrbach for the LSMDC dataset, and the AudioVault team for their priceless contribution to the visually impaired. This research is funded by EPSRC PG VisualAI EP/T028572/1, a Google-Deepmind Scholarship, and ANR-21-CE23-0003-01 CorVis.

6.8 Appendix

We first show the details of the AD collection pipeline (Sec. 6.8.1) with qualitative text examples (Sec. 6.8.2). Then we describe additional implementation details (Sec. 6.8.4) with extra qualitative movie AD examples (Sec. 6.8.5). Finally, we list the movie IDs used in our MAD-v2 split (Sec. **??**).

6.8.1 AD Collection Pipeline Additional Details

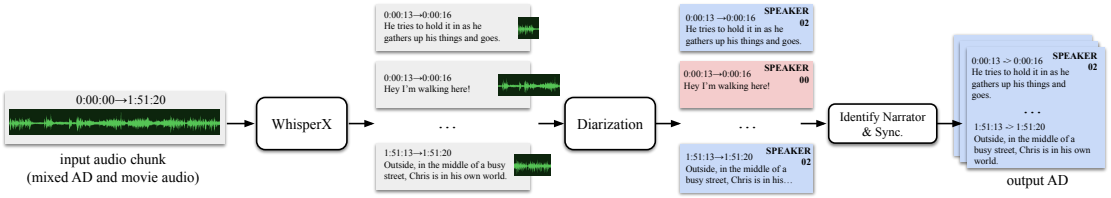


Figure 6.6: **A schematic of our AD collection pipeline.** The pipeline takes the audio file (with mixed AD and movie audio) as input, and automatically outputs the AD in text form with corresponding timestamps.

AD Collection Pipeline for MAD-v2

Collecting movie AD has two main challenges. First, in the audio files (e.g. from AudioVault) the movie AD is *fused* with the original movie audio, i.e. on the same audio track. The pipeline needs to identify the AD speaker among the movie characters accurately. Second, for the same movie, the audio files from AudioVault is usually not synchronised with the movie from which the MAD visual features were extracted, mainly due to the varied durations of intro and outro of different movie source. Since we rely on the MAD visual features, the synchronisation is an essential step.

The automated data collection pipeline is briefly introduced in Sect. 6.4 of the main paper. A schematic is shown in Fig. 8.1, in detail:

1. We transcribe the mixed audio file using *WhisperX* [Bain et al. 2023] which provides accurate punctuated transcriptions with word-level timestamps.

2. The transcript is tokenized into sentences using the nltk python toolbox [Bird 2006], resulting in transcription sentences and their corresponding temporal

segments (inferred the start and end time of the first and last word in the sentence respectively).

3. Each sentence segment is assigned a single speaker identity (e.g. `SPEAKER_00`, `SPEAKER_01`, *etc.*) by performing speaker diarization on the mixed audio, whereby each sentence timestamp is provided as oracle voice activity detection. Specifically, we use SpeechBrain ECAPA-TDNN voice embeddings [Desplanques et al. 2020] trained on VoxCeleb [Nagrani et al. 2017] and Agglomerative Clustering with a threshold of 0.95.

4. To automatically identify the cluster associated with the AD speaker, we exploit the third-person nature of AD narrations and select the cluster with the lowest proportional occurrence of first- & second-person pronouns, e.g. "I" and "you" with 95 or more speaker segments.

5. To synchronise the segment timestamps with the original audio track from which the MAD visual features were extracted, we follow [Soldan et al. 2022] and calculate the time delay τ between the original movie audio files and the mixed audio files via FFT cross-correlation. The timestamps of the identified AD segments are shifted according τ in order to synchronise them to the visual features and subtitles collected in MAD.

AD Collection Pipeline for AudioVault

The collection pipeline for AudioVault is introduced in Sect. 6.5 of the main paper, we provide more details here. To collect text-only AD annotations from AudioVault, the final synchronisation step is unnecessary. Therefore, we follow steps 1-4 of the MAD denoising pipeline as described above, which takes as input the mixed audio tracks and outputs the ASR with timestamps from the possible AD speaker.

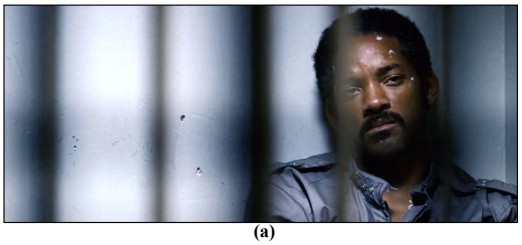
The large-scale collection from AudioVault audio files is noisy, e.g. some ADs are of lower-quality or are sourced from short movies. Therefore, we apply a stricter filtering step that removes movies containing fewer than 100 AD narrations or a word frequency of first- & second-person pronouns larger than 5%.

Comparison with MAD-v1

The key advantages of our pipeline are three-fold: (1) it relies on *audio-based* speaker separation to identify the AD speaker among the movie characters, whereas the pipeline in the original MAD work [Soldan et al. 2022] relies on *text-based* speaker separation by using the timestamps from the DVD subtitles and assumes any ASR transcription outside of these timestamps is AD. The error is propagated because the official subtitles are non-exhaustive (some dialogue is missed by the official subtitles). (2) It requires only the mixed audio as input, whereas MAD must also source the official DVD subtitles and align them – presenting additional scaling costs and challenges. (3) It uses an advanced ASR model Whisper [Radford et al. 2022b] which gives much more accurate transcriptions than previous methods, especially for punctuation and the spelling of names and other identities.

6.8.2 Qualitative Examples of MAD-v2 vs MAD-v1.

More qualitative examples of MAD-v2 and MAD-v1 are shown in Fig. 6.7 and 6.8. It is clear that our pipeline produces more accurate AD compared to the original MAD-v1, particularly in the spelling of names and the exclusion of dialogue.

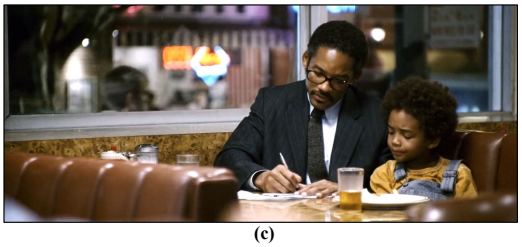


(a)

Manual Verification With a dead-eyed stare, Chris sits in a cell.
MAD-v1 Bring him back right. with a dead eyed stare, Chris sits in a cell.
MAD-v2 (ours) With a dead-eyed stare, Chris sits in a cell.

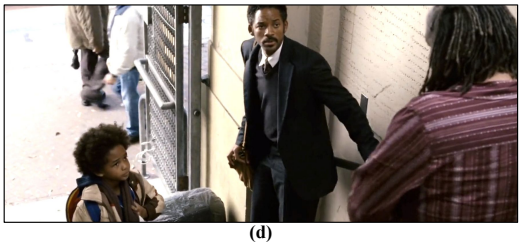
(b)

Manual Verification Chris puts the rucksack on the floor.
MAD-v1 Chris puts the rock psych on the floor.
MAD-v2 (ours) Chris puts the rucksack on the floor.



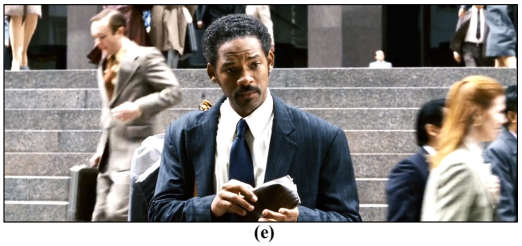
(c)

Manual Verification Later he sits in a diner with Christopher.
MAD-v1 Later he sits in a <?>.
MAD-v2 (ours) Later he sits in a diner with Christopher.



(d)

Manual Verification He comes up the steps.
MAD-v1 He comes up at steps. Can.
MAD-v2 (ours) He comes up with steps.



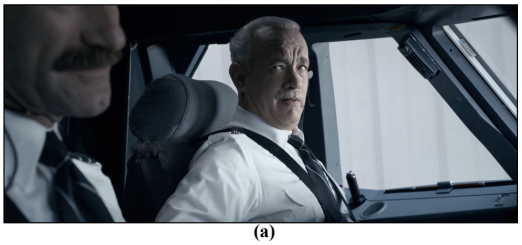
(e)

Manual Verification Chris looks ill as he watches Mr. Frohm's cab pull away.
MAD-v1 Chris looks sailors he watches Mr from cab pull away.
MAD-v2 (ours) Chris looks ill as he watches Mr. From's cab pull away.

(f)

Manual Verification An uneasy look flickers across Chris' face as Jay leaves the washroom.
MAD-v1 An uneasy look flickers across Chris's faces. Jail eats the washroom.
MAD-v2 (ours) An uneasy look flickers across Chris' face as Jay leaves the washroom.

Figure 6.7: **Comparison of the AD quality from MAD-v2 with MAD-v1.** The erroneous transcriptions are marked in red text. 'Manual Verification' means we manually transcribe the AD narration from the audio track. The sample is originally from *The Pursuit of Happyness* (2006). The failure mode of MAD-v1 in each example is **(a)** dialogue leakage, **(b)** incorrect ASR, **(c)** missing words, **(d)** dialogue leakage, **(e)** incorrect ASR and name spelling, **(f)** incorrect name spelling.



(a)

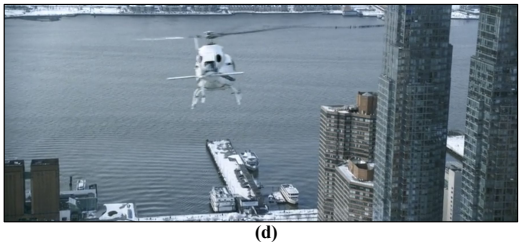
Manual Verification Sully adjusts his seat harness.
MAD-v1 Sully adjusts his seat harness, I.
MAD-v2 (ours) Sully adjusts his seat harness.

(b)

Manual Verification A male passenger looks up from his magazine.
MAD-v1 Oh yeah, a male passenger, looks up from his magazine.
MAD-v2 (ours) A male passenger looks up from his magazine.

(c)

Manual Verification Skiles turns to Sully in surprise.
MAD-v1 The hudson skiles turns to sully and surprise i.
MAD-v2 (ours) Skiles turns to Sully in surprise.



(d)

Manual Verification A sightseeing helicopter comes into view over the dark waters of the Hudson.
MAD-v1 <?> Sightseeing helicopter comes into view over <?>.
MAD-v2 (ours) A sightseeing helicopter comes into view over the dark waters of the Hudson.

(e)

Manual Verification The Manhattan skyline appears just under the wings of Flight 1549.
MAD-v1 The manhattan skyline appears just under the wings of flight fifteen. Forty nine.
MAD-v2 (ours) The Manhattan skyline appears just under the wings of Flight 1549.

(f)

Manual Verification Sully sticks out an arm as the jet bellies down onto the river.
MAD-v1 <?> The jet bellies down onto <?>.
MAD-v2 (ours) Sully sticks out an arm as the jet bellies down onto the river.

Figure 6.8: **(continue) Comparison of the AD quality from MAD-v2 with MAD-v1.** The erroneous transcriptions are marked in red text. 'Manual Verification' means we manually transcribe the AD narration from the audio track. The sample is originally from *Sully: Miracle on the Hudson* (2016). The failure mode of MAD-v1 in each example is **(a)** dialogue leakage, **(b)** dialogue leakage, **(c)** dialogue leakage and incorrect ASR, **(d)** missing words, **(e)** number spelling and sentence partitioning, **(f)** missing words.

6.8.3 Quantitative Comparison between MAD-v2 vs MAD-v1 on Grounding

We re-purpose the CLIP zero-shot video-language grounding (VLG) performance from [Soldan et al. 2022] as an indicator of dataset quality. In detail, for both MAD-v2 and MAD-v1, we randomly choose a set of 5 movies from the *training split*, and compute the VLG performance with frozen CLIP visual and textual encoders. The AD textual quality and timestamps are the only factors that differ in this comparison. We use the MAD training split because we did not modify the val/test splits, which are from LSMDC annotations. The code to compute VLG performance is from `https://github.com/Soldelli/MAD`. The result in Table 6.6 shows MAD-v2 annotations also benefit the VLG task.

R@50	IoU@0.1	IoU@0.3	IoU@0.5
MAD-v1-`Unnamed`	32.08	22.85	14.26
MAD-v2-`Unnamed`	**33.25**	**24.22**	**15.58**

Table 6.6: CLIP zero-shot VLG performance on MAD-v1 and MAD-v2.

6.8.4 Additional Implementation Details

Design Choices.

- Number of frames per movie clip N: We choose $N = 8$. Most AD annotations have a time duration of 1-3 seconds, equivalent to 5-15 frames (features) under 5 FPS – the sampling rate provided by the MAD dataset. Therefore $N = 8$ is a reasonable choice.
- Number of AD sentences as context K: We experiment $K \in \{1, 2, 3, 6\}$ in Figure 6.4.
- Number of subtitles L: As described in Sect. 6.6.1, for simplicity we take the most recent 4 dialogues within a 1-minute time window. Note that the time distribution of subtitles varies a lot – the most recent 4 dialogues could span just a few seconds or up to minutes before the current AD timestamp.

Evaluation Metrics. We use the `pycocoeval` package from `https://github.com/tylin/coco-caption` to compute the ROUGE-L, CIDEr, SPICE and ME-

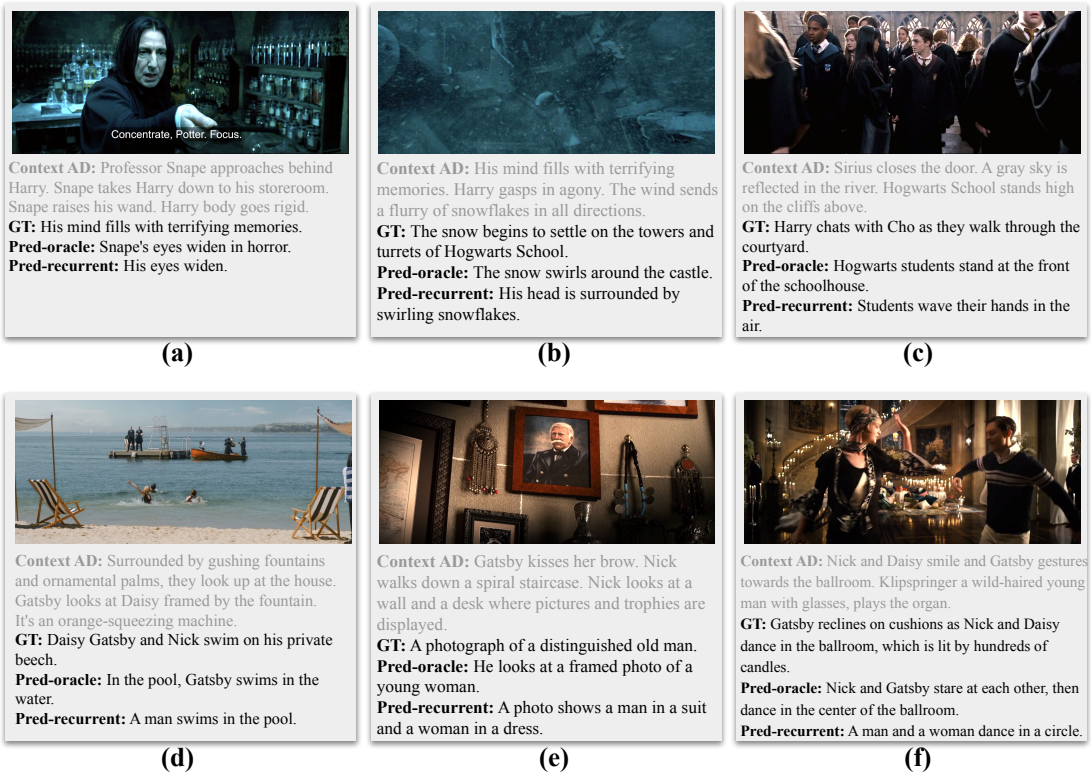


Figure 6.9: **Qualitative examples of AutoAD model.** We show the ground-truth AD and the AD predictions under both the oracle and recurrent settings. Previous AD context is shown in gray. Samples are taken from *Harry Potter and the Order of the Phoenix* (2007) and *The Great Gatsby* (2013).

TEOR. The package post-processes both the predicted text and ground-truth text internally to remove the punctuation and make them lowercase. To compute the BertScore, we use the package from https://github.com/Tiiiger/bert_score. Note that before computing the BertScore, both the predicted text and the ground-truth text are converted to lowercase without any punctuation, as these are factors that the BertScore is sensitive to.

Alternative approach for vision-language fusion. We investigate an alternative vision & language fusion mechanism whereby the context AD sentence prompts are fed as *language features* rather than *raw text tokens*. Empirically, we observe that raw text inputs outperform language features (e.g. 12.6 CIDEr in Table 6.2 vs. about 8.0 CIDEr when feeding language features).

6.8.5 Additional Qualitative Examples

More qualitative examples are shown in Fig. 6.9. It shows that the AutoAD model gives reasonable descriptions for the movie domain, like the actions (swim, dance), and face expression (eyes widen). Note that under the oracle setting, the model is capable of learning character names (sample **a, c, d, f**) mainly due to the extra information from the ground-truth context. The model is still limited in its ability to identify characters accurately, e.g. in sample **f**, the movie shows Nick and *Daisy* are dancing. Whereas the oracle prediction describes that Nick and *Gatsby* are dancing, and the recurrent model simply predicts that a man and woman are dancing. Also the pronouns often appear in the recurrent prediction, such as the word 'his' in sample **a** and **b**, which shows the model learns the bias of pronouns but cannot recognize characters correctly.

Chapter 7

AutoAD II: The Sequel – Who, When, and What in Movie Audio Description

The paper has been accepted for publication at the International Conference on Computer Vision (ICCV), 2023.

AutoAD II: The Sequel – Who, When, and What in Movie Audio Description

Tengda Han[1] Max Bain[1] Arsha Nagrani[1†]
Gül Varol[1,2] Weidi Xie[1,3] Andrew Zisserman[1]

[1]Visual Geometry Group, University of Oxford

[2]LIGM, École des Ponts, Univ Gustave Eiffel, CNRS

[3]CMIC, Shanghai Jiao Tong University

Abstract

Audio Description (AD) is the task of generating descriptions of visual content, at suitable time intervals, for the benefit of visually impaired audiences. For movies, this presents notable challenges – AD must occur only during existing pauses in dialogue, should refer to characters by name, and ought to aid understanding of the storyline as a whole.

To this end, we develop a new model for automatically generating movie AD, given CLIP visual features of the frames, the cast list, and the temporal locations of the speech; addressing all three of the 'who', 'when', and 'what' questions: (i) who – we introduce a *character bank* consisting of the character's name, the actor that played the part, and a CLIP feature of their face, for the principal cast of each movie, and demonstrate how this can be used to improve naming in the generated AD; (ii) when – we investigate several models for determining whether an AD should be generated for a time interval or not, based on the visual content of the interval and its neighbours; and (iii) what – we implement a new vision-language model for this task, that can ingest the proposals from the character bank, whilst conditioning on the visual features using cross-attention, and demonstrate how this improves over previous architectures for AD text generation in an apples-to-apples comparison.

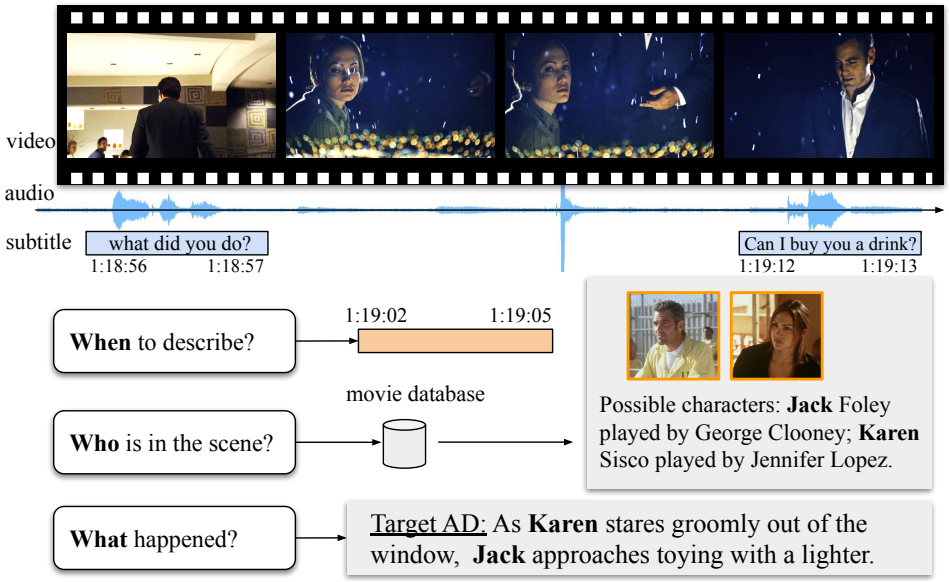


Figure 7.1: **AutoAD II:** We propose an automatic AD system that addresses key challenges - *when* to generate AD, *who* is in the scene and *what* is happening visually.

7.1 Introduction

> *For in acts we must take note of* ***who*** *did it, by what aids or instruments he did it (with),* ***what*** *he did, where he did it, why he did it, how and* ***when*** *he did it.*
>
> –Thomas Aquinas

Audio Description (AD) is the descriptive narration of visual elements in a video, that are not represented in the original audio track. While there has been a proliferation of online content with *closed captioning*[1] due to advancements in ASR, a vast majority of video online does not have AD, mostly due to the prohibitive cost of generating it ($30 per minute[2]). Generating AD automatically at scale has multiple benefits; not only does it improve access for the visually impaired – it may also enhance the visual experience for sighted users (sight-free multitasking such as driving, enhanced memory for visual details, language learning, and also aiding those with other cognitive disabilities) [Perego 2016]. Generating AD for movies is also an important research area in computer vision as it requires a system to perform multi-modal reasoning of long videos over time.

Despite these benefits, the progress in generating AD is still at a very nascent stage,

[1]Transcription of the speech
[2]https://www.3playmedia.com/blog/select-audio-description-vendor/

due to the following challenges: (a) An ideal AD generation system should perform two tasks simultaneously – first, determine when to generate AD by proposing temporal segments; second, generate AD for the proposed segments. Previous works ignore the *when* completely, operating on already trimmed video segments [Y. Yu et al. 2021]. (b) Secondly, given the strong relevance of characters to stories [Tapaswi et al. 2019; Kukleva et al. 2020b], AD typically includes references to a character's name (*who* is in the scene), their emotion, and their actions. This is particularly challenging as characters change from movie to movie. Due to anonymised test sets (LSMDC [A. Rohrbach et al. 2015b]), the relevance of character names in AD is often ignored [Y. Yu et al. 2021]. (c) Finally, AD also differs significantly from image or video captioning [K. Lin et al. 2022; H. Luo et al. 2020; Seo et al. 2022] in that it does not need to provide descriptions of events that can be understood from the sound track alone (such as dialogue and ambient sounds) and should incorporate previous context to create a pleasurable listening experience without being repetitive or redundant. Such aspects require reasoning over multi-modal inputs (i.e., vision, text, and speech) over time while determining *what* to generate. In this work we propose an AD system that focuses on all these three W's – *when*, *who* and *what* (Fig 7.1).

To address *when*, we introduce a module to first propose temporal segments for AD. The time intervals for possible AD are constrained in that they do not overlap with the dialogue, but whether an AD is provided or not in the permissible time intervals depends on a number of factors including: the importance of the visual content to the story line, ambiguity in the audio soundtrack, and new information relative to previous AD.

For *who*, we introduce an AD model that can incorporate character information *on-the-fly* by referring to a text-visual *character bank* for that movie. One of the challenges of AD is that each movie has a different set of characters (and the actors that play them) that ought to be referenced in the AD captions. We address this by training a visual-language model to refer both to the external character bank and to the visual content of the scene when generating AD. The model can then be applied to any movie, given its cast list, without requiring retraining. This significantly improves references to characters, both in actual naming and in pronouns, in the generated AD compared to previous methods [T. Han et al.

2023] that could only access names and pronouns present in the dialogue. Since character references appear in approximately 40% of AD, this is an important improvement.

The final challenge is *what* to generate, and involves reasoning over multimodal inputs – images, character bank and previous AD context. We do this via a novel multimodal cross-attention architecture, which ingests proposals from the character bank, and then conditions on visual features extracted from the movie frames.

Our contributions are the following. (1) We introduce a *Character Bank* to enable our AD generation model to label the characters appearing in the film. (2) We propose a Flamingo-style [Alayrac et al. 2022] architecture for the task, and compare this approach to the prompt style [Mokady et al. 2021] architecture used previously for AD [T. Han et al. 2023]. (3) We build a model for predicting *when* AD should be inserted, i.e. where on the timeline (using speech detection and visual cues). (4) Given the existing challenges with captioning based metrics [Fujita et al. 2020], we employ a new evaluation metric for the AD content performance based on retrieval compared to other AD sentences in the movie. (5) We significantly outperform the previous state-of-the-art on the MAD dataset [Soldan et al. 2022; T. Han et al. 2023].

7.2 Related Work

Dense Video Captioning. Dense video captioning is the task of temporally localising and captioning all events in an untrimmed video [Krishna et al. 2017a; T. Wang et al. 2021; L. Zhou et al. 2018c]. This differs from standard video captioning [Sharma et al. 2018b; K. Lin et al. 2022; H. Luo et al. 2020; Seo et al. 2022], where the goal is to produce a single caption for a given trimmed video clip. While most methods for dense video captioning [Krishna et al. 2017a; Iashin and Rahtu 2020a; Iashin and Rahtu 2020b; Jingwen Wang et al. 2018; T. Wang et al. 2020] consist of a 2-stage pipeline: a temporal localization stage followed by an event captioning stage; recent works [Chadha et al. 2021; Shaoxiang Chen and Jiang 2021; C. Deng et al. 2021; Y. Li et al. 2018; Mun et al. 2019; Rahman et al.

2019; Shen et al. 2017; Shi et al. 2019; Jingwen Wang et al. 2018; T. Wang et al. 2021; L. Zhou et al. 2018c; A. Yang et al. 2023] jointly train the captioning and localization modules in order to improve inter-event relationships. The datasets for this task are largely obtained from web videos (e.g. YouCook2 [L. Zhou et al. 2018b], ViTT [G. Huang et al. 2020] and ActivityNet Captions [Krishna et al. 2017a]). Unlike these works, AD captions must be complementary to the audio information, tell a coherent story, and must not overlap with dialogue.

Movie Understanding. Early pioneering works exploit movies to learn actions [Laptev et al. 2008]. The LSMDC [A. Rohrbach et al. 2015b] movie dataset sources its annotation from AD narrations and applies significant post-processing – character name anonymization and manual timestamp refinement – to ensure high correspondence between the short video clips and their captions. A series of short-form video tasks have since derived from LSMDC, including retrieval [Bain et al. 2021], person grounding [Y. Yu et al. 2020], and sequential video captioning. TPAM [Y. Yu et al. 2021] tackles the latter, prompting a frozen GPT-2 with local visual features. Later works propose tasks that require more long-form modelling, including aligning movies to books [Tapaswi et al. 2015a; Yukun Zhu et al. 2015] and synopses [Xiong et al. 2019]; long video retrieval with the Condensed Movies Dataset (CMD) [Bain et al. 2020b] and summarization [Papalampidi et al. 2021].

Characters in Movies. A distinctive characteristic of movie understanding, setting it apart from other video domains, is its *character-centric* nature. Thus, character recognition is a prerequisite for the task, and many works have proposed automatic identification pipelines using face, voice, and body information [Everingham et al. 2006; Tapaswi et al. 2012b; Tapaswi et al. 2019; Nagrani and Zisserman 2017; A. Brown et al. 2021c]. Similar to our work, [Nagrani and Zisserman 2017; Q. Huang et al. 2018; A. Brown et al. 2021b] initialize their character recognition pipeline with actor portraits, which can be further refined with noisy image captions [Q. Huang et al. 2020c]. Recently, CLIP has proved to be effective for zero-shot frame-level character labelling [Korbar and Zisserman 2022], alleviating the need for complex detection pipelines, which also inspires our character identification pipeline from CLIP features. Dense labelling of characters in movies and TV shows enables the modeling of interactions, relationships, and intentions – which can be formulated into classification [Kukleva et al. 2020b], question an-

swering [Lei et al. 2018], or captioning [Lei et al. 2020] tasks. Unlike these works, we use a character bank in a zero-shot practical setting for a real-world task: automatic AD generation.

Automated Audio Description. Visual captioning for assistive technologies is a growing area of computer vision research [Dognin et al. 2020; Gurari et al. 2020]. Yet, generating AD for video is still a relatively unexplored area of research. Initial work [Y. Wang et al. 2021] applies heuristic cost-based filtering to video captioning on ActivityNet to generate diverse and relevant captions more akin to AD.

Most similar to this work is AutoAD [T. Han et al. 2023], the first on AD generation for movies. They adopt the MAD [Soldan et al. 2022] dataset for training, and provide a cleaner set of AD annotations using their improved automated data collection pipeline based on WhisperX [Bain et al. 2023]. With this new scalable pipeline, they introduce AudioVault – a text-only AD corpus from over 7k movies – to enable in-domain LLM pretraining, resulting in substantial improvements to AD generation. They do not however tackle the problem of *when* to generate AD, assuming these segments are given a-priori, nor do they deal with the problem of *who* – with their model failing to generate coherent character names, a critical component to story-coherent AD generation for long-form video content such as movies and TV shows.

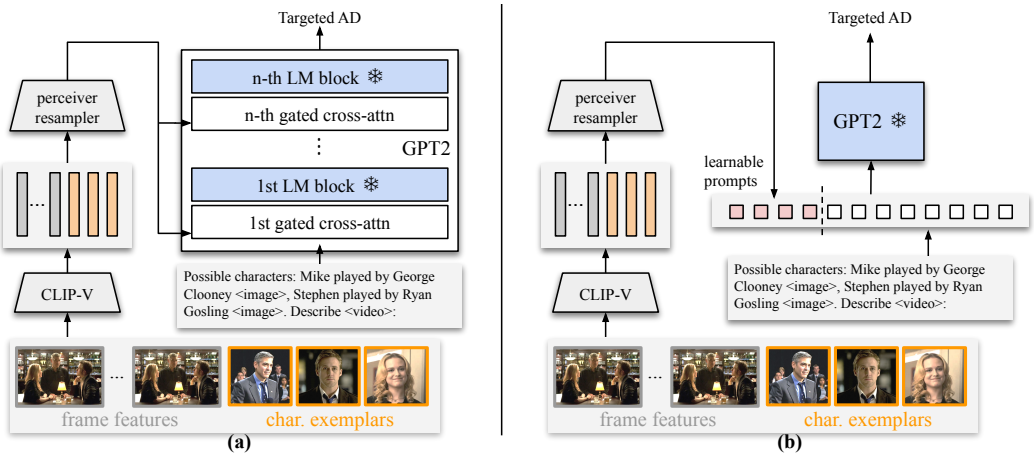


Figure 7.2: Architecture comparison: **(a) x-attn based method** vs. **(b) prompt learning based method**. We use architecture **(a)** in this paper. The character information is fed to the model in text form. Following Flamingo [Alayrac et al. 2022], we add text tags '⟨image⟩' to indicate the association between texts and character exemplar features. The model can also take context AD as additional text input, by simply appending more input text tokens.

7.3 New Models for Generating AD

Inspired by [T. Han et al. 2023], our method consists of adapting a large language model (LLM) for the task of generating AD. In the following sections, we describe three novel contributions: the first involves visual conditioning of multiple layers of the LLM (Sec. 7.3.1) in order to generate AD within a given time segment; the second describes a novel mechanism for incorporating character information *on-the-fly* that enables the model to infer a character's name in the scene (Sec. 7.3.2); and the third presents a simple approach for proposing temporal segments for where in time (when) the AD should be generated (Sec. 7.3.3).

7.3.1 A Visually Conditioned LM for Generating AD

Given a movie clip consisting of multiple frames $\mathbf{x}_i = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_N\}$, our aim is to produce AD text $\mathcal{T}_i$ that describes the visual elements in a way that helps the visually impaired follow the story line. To achieve this we build on the capabilities of a pre-trained and frozen generative language model (LM). Broadly, two types of architecture are currently used to condition a LM on visual inputs: (a) by introducing additional layers into the LM that cross-attend to the visual input (examples include Flamingo [Alayrac et al. 2022]); or (b) by mapping the visual input to tokens that act as prompts for the LM (examples include ClipCap [Mokady et al. 2021]). In both cases the LM is then able to generate descriptions of the visual inputs. In our case we have multiple video frames (represented by CLIP [Radford et al. 2021] vectors) and we use a Perceiver resampler to produce a fixed sized sequence of vectors for the visual input. The two types of architecture are illustrated in Figure 7.2.

In this paper, we develop a model based on type (a), with additional cross-attention layers in the LM. We describe this in more detail below, and demonstrate in the results that it has superior performance over type (b) in our case. We also briefly discuss the advantages and disadvantages of the two types of architecture below.

Architecture description. In detail, the architecture has three components: (i) a CLIP encoder that generates visual features from the input movie frames as $\mathbf{z} = f_{\text{CLIP}}(\mathcal{I}_1, \mathcal{I}_2, ...)$; (ii) a Perceiver resampler that models the contextual

information amongst these visual features and summarizes them into a sequence of fixed-length vectors: $\hat{\mathbf{x}} = \mathcal{P}([\mathbf{z}; \mathbf{x}])$, where $\mathbf{x}$ are learnable latent states of the Perceiver module $\mathcal{P}$; and (iii) trainable cross-attention blocks that are inserted into the frozen language model. Each cross-attention block is controlled by a `tanh` gating mechanism, which is initialized with zero values such that the language model maintains its original activation at the beginning of the training as $\mathbf{h}_{j+1} = \mathbf{h}_j + \texttt{tanh}\Big(\texttt{XAttn}(\mathbf{h}_j, \hat{\mathbf{x}}, \hat{\mathbf{x}})\Big)$, where $\mathbf{h}_j$ is the hidden vector of the j-th block of the language model and $\texttt{XAttn}(q, k, v)$ denotes the cross-attention module with its query, key and value inputs in order.

Flexibility for multimodal context. For our purposes the Flamingo-like architecture offers flexibility: the input can simply be the video frames (via the Perceiver resampler) and a text prompt to the LM, such as 'Describe ⟨video⟩:' to start the AD generation. However, in the case that additional image and text context is available (as in the additional character naming and image examples from the character bank, described below), then this can simply be prepended to the prompt, and the trainable cross-attention layers learn how to correctly attend to both the video frames and the image examples

In contrast, for the second type of architecture, where the visual input acts as a prompt to the LM, it is necessary to train new tokens, such as BOS [T. Han et al. 2023], in order to separate visual prompts from text prompts and start the AD generation.

In summary, both architectures build on frozen LMs (previous works [Alayrac et al. 2022] show that finetuning an LLM on the task-of-interest can harm their generalization) and have trainable parameters to allow the LM to condition on the visual input and adapt to the AD task. However, the cross-attention type of architecture offers greater flexibility and, as will be seen, superior performance.

7.3.2 Incorporating a Character Bank

Our goal is to recognize *active* characters – defined as those appearing on-screen – in a given movie clip by leveraging the movie cast list from an external movie database $\mathcal{M}$, and thereby provide the information about active characters to the AD generation. To this end, we (i) build visual character exemplar features by

exploiting actor portrait images from $\mathcal{M}$, further calibrated by comparing against the movie frames, and (ii) train a character recognition module that predicts the active characters given their exemplars and the movie clip.

Given a long-form movie $\mathcal{V}$, the corresponding cast list can be queried from the database $\mathcal{M}$. The character bank for this movie $\mathcal{V}$ can be written as $\mathcal{B}_\mathcal{V} = \{[\texttt{char}_j, \texttt{act}_j, \mathcal{A}_j]\}_{j=1}^{C}$, where C denotes the number of characters, $\texttt{char}_j$ is the character name in the movie, $\texttt{act}_j$ is the actor name, and $\mathcal{A}_j$ is the actor's portrait image from the movie database. Below are two example items in a character bank:

$$\{[\text{Jack Dawson}, \text{Leonardo DiCaprio}, \mathcal{A}_{\text{LD}}],$$
$$[\text{Rose DeWitt-Bukater}, \text{Kate Winslet}, \mathcal{A}_{\text{KW}}], ...\}$$

Calibrating the actor portrait feature. An actor's portrait image can differ considerably in appearance from the character in the movie due to various factors, such as hairstyle, makeup, dress, ageing, or camera viewpoint [Nagrani and Zisserman 2017]. In particular, for older movies with different dressing styles and fewer close-up shots, actor portraits might lie very far from the movie's frame in the feature space. To overcome this issue, we propose a calibration step. Instead of using the image features from the actor's portrait, we retrieve the top-k nearest frames within the same movie, and average the frame features to create an exemplar for that character. Specifically, let $\mathbf{z}_\mathcal{V} = f_{\text{CLIP}}(\mathcal{V})$ denote the sequence of visual features of the movie $\mathcal{V}$, and given a portrait image of actor j as $\mathcal{A}_j$, we first compute its visual feature $z_j = f_{\text{CLIP}}(\mathcal{A}_j)$, and compare it against $\mathbf{z}_\mathcal{V}$ via cosine similarity. The character exemplar feature of actor j in the movie $\mathcal{V}$ can then be computed by:

$$e_j = \frac{1}{k} \sum \mathbf{z}_\mathcal{V} \left[\text{top-}k\Big(\frac{z_j^\top \mathbf{z}_\mathcal{V}}{|z_j| \cdot |\mathbf{z}_\mathcal{V}|} \Big) \right],$$

where top-k finds the indices of the k most similar frames and [.] symbol means the indexing operation. In Appendix, we show this calibration procedure (i.e., replacing z_j with e_j) is essential for constructing reliable character banks.

Recognizing characters in the movie clip. Not all characters appear on-screen at the same time. With a character bank $\mathcal{B}_\mathcal{V}$ for the movie $\mathcal{V}$, our goal

is to recognize the *active* characters that appear between times t_1 and t_2 to enable naming them in the AD generation. This character recognition task may be achieved by face detectors [Jiankang Deng et al. 2020], or even speaker recognition from voice [W. Xie et al. 2019]. However, for the movie datasets used in this work, the absence of raw frames prohibits the use of face detection, and the characters mentioned in AD may not necessarily be speaking. Instead, we propose to use a character recognition module based purely on frame-level visual features and the character bank information $\mathcal{B}_\mathcal{V}$.

As shown in Fig. 7.3, both the exemplar features for each character $\{e_j\}_{j=1}^{C}$ and the movie frame features $\mathcal{V}_{[t_1,t_2]}$ are first fed to a linear projection layer, which aims to project general visual features onto a face feature space. Then a relatively shallow (2-block) transformer decoder takes both projected features and outputs a probability for each character on whether they appear between times $[t_1, t_2]$. This module is trained with a binary classification loss. The labels can be obtained from face annotations from datasets like MovieNet [Q. Huang et al. 2020a]. In Appendix, we also experiment with the labels obtained by running named entity recognition (NER) [Nadeau and Sekine 2007] on the AD annotation, which performs worse than MovieNet.

Unlike our proposed method, earlier work [T. Han et al. 2023] attempts to mine character names from subtitles using NER and provide these as prompts to the AD generation model – but this still fails to reference character names effectively. It may be because character names occur sparsely in subtitles[3] or names found may refer to off-screen characters. In Appendix, we show that compared to the names from the subtitle, the movie cast list (when narrowed down to those that appear on the scene with our character recognition module) provides a much higher precision and recall of active on-screen characters.

In Sec. 7.5.2, we also compare our model with another baseline method: thresholding the similarity between the character's exemplar and movie frame features with a scalar α. This experiment shows that a simple transformer decoder module outperforms the baseline.

Existing challenges. We notice that the NER-based label collection process is

[3]Based on the MAD-train movies, approximately 13% of subtitle sentences contain character names, compared to 41% of AD.

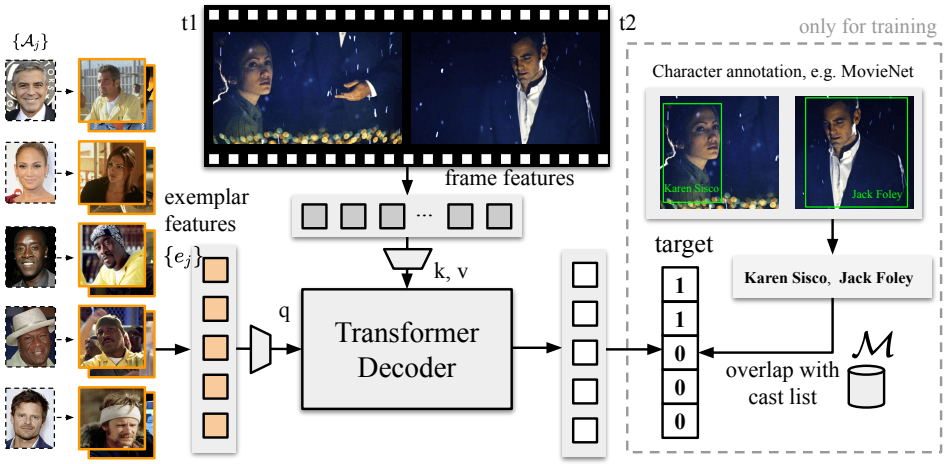


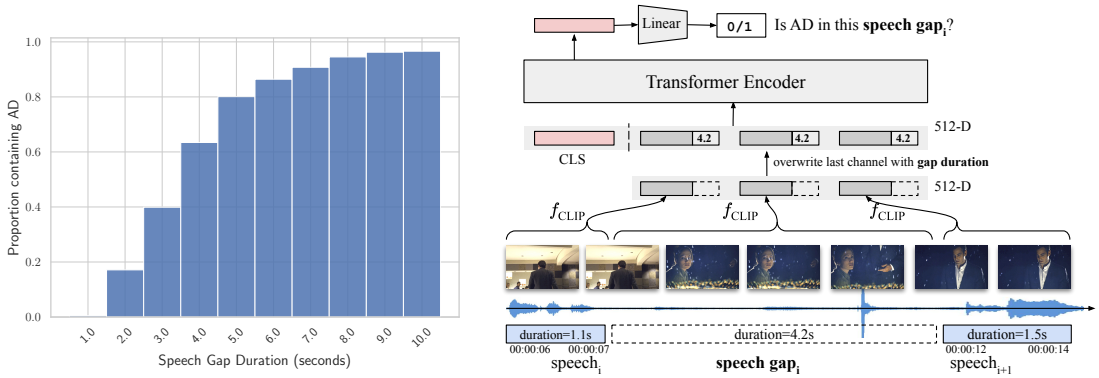
Figure 7.3: Character recognition module: Given character exemplar features for C characters $\{e_j\}_{j=1}^{C}$, and movie frame features for a given clip, we formulate a binary classification problem to determine whether each character is active in the scene or not. The label can be obtained from character annotations like MovieNet, and checking against the cast list in our movie database.

not exhaustive, e.g. character names referred to in AD may not be in the cast list, or equally, on-screen characters may not be referred to in the corresponding AD. Other than that, in AD sentences, Further, a character may be named in a multitude of ways, e.g. first name only – 'Albus', or last name with a prefix – 'Mr. Dumbledore', their professions, titles or pronouns – 'Professor', 'Prof. Dumbledore' or 'He', their relationships to other characters – 'Aberforth's brother', or other nicknames etc. Another challenge is that some ADs may contain references to off-screen characters – 'He went into Dumbledore's office'. We leave the mining of such references for future work.

Using the character bank for AD generation. A trained character recognition module can recognize the *active* characters in any video clip $\mathcal{V}_{[t_1,t_2]}$. Next, we feed this character information into our AD generation pipeline.

In Sec. 7.3.1, we introduce a versatile cross-attention-based architecture which supports textual and other multi-modal inputs. We feed in character information to the model mainly by *text prompting*. In more detail, given a character list for the movie clip $\mathcal{V}_{[t_1,t_2]}$, we explore three different ways of supplying the active characters in the scene. Let's assume [$\texttt{char}_1$, $\texttt{char}_2$] are recognized as active. The prompting templates are then:

1. "possible characters: $\texttt{char}_1$, $\texttt{char}_2$."
2. "possible characters: $\texttt{char}_1$ played by $\texttt{act}_1$; $\texttt{char}_2$ played by $\texttt{act}_2$."



(a) (b)

Figure 7.4: **(a) Proportion of speech gaps containing AD relative to their duration** – very short speech gaps rarely contain AD, and large speech gaps nearly always contain AD. The statistics are from the MAD training set. **(b) Architecture for AD temporal proposal classification.** Given a speech gap, the model classifies whether or not AD should be inserted in the gap, taking visual and duration cues as input.

3. "possible characters: $\texttt{char}_1$ played by $\texttt{act}_1$ ⟨image⟩; $\texttt{char}_2$ played by $\texttt{act}_2$ ⟨image⟩."

Note that in method (3), the ⟨image⟩ tag is purely in the text form; therefore, in this setting, we feed in the character exemplar features $[e_1, e_2]$ in the corresponding order to the perceiver resampler, such that it can learn the association between the character's identity and the movie clip.

7.3.3 Proposing AD Temporal Segments

An ideal AD system must not only generate high quality AD narrations (*what*), but must also decide *when* to generate AD. The Web Content Accessibility Guidelines 2.0 [Caldwell et al. 2008] outlines specific criteria for successful AD: (i) it must only be added during existing pauses in dialogue; and (ii) it need not be added when all of the video information is already provided in existing audio.

In practice, long pauses in dialogue and the subjectivity of the second guideline mean these provide rather weak constraints on the timing of AD, resulting in large variations between human-generated AD timestamps for the same movie[4]. Such variation makes it difficult to learn and evaluate fine-grained model predictions of

[4]An analysis on Audio Descriptions and inter-annotator agreement is provided in the Appendix.

proposed AD temporal segments. Therefore, we formulate the temporal proposal task into one of binary classification: *given an existing pause in dialogue, should AD be inserted in the pause?* This coarse-grained formulation has much higher inter-annotator agreement and inherently satisfies the first guideline for generating AD[4].

Given a long-form movie $\mathcal{V}$, our goal is to identify inactive speech regions and classify whether or not they should contain AD. First we apply voice activity detection (VAD) to the audio $\mathcal{U}$, resulting in a sequence of N non-overlapping segments each corresponding to active speech regions of the *existing audio* $\mathcal{S} = [S_1, S_2, ..., S_N]$, where $S_i = (t_0^i, t_1^i)$. The inactive speech regions $\mathcal{G}$ in the existing audio $\mathcal{U}$, or *speech gaps*, are simply the inverse of the active speech regions $\mathcal{S}$, $\mathcal{G} = [G_0, G_1, ..., G_{N-1}]$ where $G_i = (S_i[1], S_{i+1}[0])$. We then extract mean-pooled CLIP visual features for the speech gap in question and the two adjacent speech segments $\mathbf{z}_i = f_{\text{CLIP}}(S_i, G_i, S_{i+1}) \in \mathbb{R}^{3\times D}$. To inject duration information into the model while still maintaining the feature dimension for multi-head attention, we replace the last channel in the visual features with the duration of the queried speech gap, $\mathbf{z}[:, -1] = |G_i|$. The duration-injected visual features are then fed to a transformer encoder, and the output `[CLS]`token is linearly projected to a single logit for binary classification. We also investigate injecting audio features but this did not improve classification performance (Sect. 7.5.3).

By analysing the distribution of AD data (Figure 7.4), we find that whether or not AD is contained within a given speech gap is highly correlated with the duration of said gap. In fact, gaps of two seconds or less contain AD only 17% of the time. At the other extreme, gaps of 5 seconds or more contain AD 80% of the time. Due to such strong duration correlations , we restrict the prediction task to speech gaps between two and five seconds. The classification of whether to insert AD within shorter or longer speech gaps can be obtained via a hard-coded rule.

7.4 Implementation Details

7.4.1 Training Data

MAD [**Soldan et al. 2022**] is a movie audio description dataset consisting of movie frame features and timed AD in the text form. We follow [T. Han et al. 2023] and use 488 movies as the training set. Specifically for AD, we use the same preprocessing pipeline proposed in [T. Han et al. 2023] to obtain high-quality ASR outputs. We use the 'named' version of MAD dataset. **AudioVault-AD** [**T. Han et al. 2023**] is a text-only corpus of AD for 7057 movies downloaded from the AudioVault website. The movies are not included in MAD dataset. We use the AudioVault-AD for text-only pretraining. **WebVid** [**Bain et al. 2021**] is a dataset of 2.5M captioned short videos for visual-only pretraining. We find the NER from both LSMDC-train and MAD-train contain non-trivial noise, despite the one for LSMDC-train having been manually verified. **MovieNet** [Q. Huang et al. 2020a] is a movie dataset providing movie keyframes and various annotations including character names for each keyframe. We choose an overlap of MovieNet movies with MAD training movies to train the character recognition module.

7.4.2 Testing Data

MAD-eval [**T. Han et al. 2023**] consists of 10 movies for evaluating AD captioning from the LSMDC validation and testing set. The timestamps from LSMDC are manually edited to ensure high visual correspondence with the caption. We treat this as our standard evaluation for measuring AD caption quality.

MAD-t-eval is our proposed benchmark for evaluating AD time point prediction. The edited timestamps in *MAD-eval* are not appropriate for measuring temporal proposals because they are expanded and often overlap with speech segments. Therefore we evaluate time prediction models on *MAD-t-eval*, consisting of three movies (from MAD-eval) where the AD and their original timestamps are sourced from Audiovault and manually verified. We restrict the evaluation to speech gaps with a duration between two and five seconds, resulting in 530 gaps across the three movies.

7.4.3 Collecting Character Banks

The character information for movies can be collected from online databases or review websites like IMDb[5]. In detail, for each movie in Audiovault, MAD-train and the MAD-eval datasets, we download the top 10 cast information from IMDb including the actor names, their character role name, and the actor portrait image. Full details are provided in Appendix.

7.4.4 Training & Inference Recipe

In this section, we first outline the architectures used for each module in the AD captioning system; we then describe how each module individually is pretrained; and finally we describe the finetuning and inference details for the full AD captioning system.

Architectural components

AD generation model (Section 7.3.1) is built on top of *GPT2-small*, specifically the open-source version from HuggingFace. We insert an X-attn block after *each* of the transformer block of GPT-2. The perceiver resampler has two transformer decoder blocks with 10 latent vectors. For the visual encoder, we use CLIP [Radford et al. 2021] ViT-B/32 model which extracts 512-d features for each movie frame. These features are provided by the MAD dataset [Soldan et al. 2022].

Character recognition module (Section 7.3.2) consists of a linear layer and a 2-block transformer decoder. It takes the movie character exemplar features $\{e_j\}$ and movie clip features as input, and outputs a probability for each exemplar feature.

AD temporal proposals (Section 7.3.3). For VAD, we use the pyannote model [Bredin et al. 2020a]. For the temporal proposal classification model, we use a 3-layer transformer encoder with sin-cos positional embeddings. For the visual and audio features we use CLIP ViT-B/32 [Radford et al. 2021] and VGGish [Gemmeke et al. 2017], respectively.

[5]https://www.imdb.com/

Pretraining recipe

To overcome the limited amount of paired AD training data, we follow [T. Han et al. 2023] and perform partial data pretraining for each component in our modular architecture.

GPT-2 (Section 7.3.1). We follow [T. Han et al. 2023] and perform secondary in-domain pretraining of GPT-2 on the Audiovault text-only corpus to match the text distribution for AD generation.

Video captioning (Section 7.3.1). We pretrain the cross-attention visual captioning blocks on 2.5M video-text pairs from WebVid [Bain et al. 2021], while keeping the GPT-2 LM block weights frozen.

Character recognition module (Section 7.3.2). The module is trained on character name labels from MAD-L-char.

Finetuning & Inference

AD captioning (Section 7.3.1). With the recognized active character list as an additional input and model parameters partially pretrained, the AD generation model is finetuned on MAD-train with an AdamW [Loshchilov and Hutter 2017] optimizer and 10^{-4} learning rate. For output text sampling, we use beam search with the beam size of 5 and report results by the top-1 beam-searched outputs, since it performs slightly better than greedy search on multiple scenarios. The full training details are in Appendix.

AD temporal proposals (Section 7.3.3). The transformer encoder is trained on the MAD dataset for three epochs, with a BCE loss and an AdamW [Loshchilov and Hutter 2017] optimizer of learning rate 10^{-4}. The classification task at training and inference is restricted to speech gaps with durations between 2-5 seconds.

7.5 Experiments

The experimental section is organised as follows: we start by describing the evaluation metrics in Sect. 7.5.1; then in Sect. 7.5.2, we demonstrate the effectiveness

Methods	ROC AUC	Average Precision
Cosine-Sim	0.72	0.55
TFM Decoder	**0.93**	**0.87**

Table 7.1: We compare different methods for recognising characters in a clip, reported on ten MAD eval movies.

of our proposed architecture and training strategy, based on the groundtruth AD time segments, for example, visually conditioned LM, effect of character bank, and partial-data pretraining; in Sect. 7.5.3, we evaluate on the temporal proposal, and present qualitative results in Sect. 7.5.4.

7.5.1 Evaluation Metrics

Classic metrics for text generation. We adopt classic captioning metrics to compare the generated AD to the ground-truth AD, namely, ROUGE-L [C.-Y. Lin 2004] (**R-L**) and CIDEr [Vedantam et al. 2015] (**C**).

Retrieval-based metric for text sequence generation. We propose a new recall-based metric: 'Recall@k within N neighbours' (**R@k-N**). In detail, given two sequences of generated texts and ground-truth (GT) texts in their temporal order, for each generated text at time point $[t_1, t_2]$, we compute the Recall@k with N adjacent GT texts, then average the score. To compute recall, we use the BertScore [T. Zhang et al. 2020] as the text similarity measure. Classic captioning metrics like CIDEr or ROUGE-L are mainly based on n-gram accuracy, which tends to over-penalise the system on linguistic text variations, i.e. ecause there are multiple ways to express the same meaning. The retrieval-based **R@k-N** metric is less affected by these low level variations in the text.

Metrics for character recognition and time proposal. The character recognition (described in Sect. 7.3.2) and the time segment proposal tasks (described in Sect. 7.3.3) are formulated as multi-label and binary classification problems respectively We report ROC-AUC and Average Precision for the classifiers, with class macro-averaging for the multi-label case.

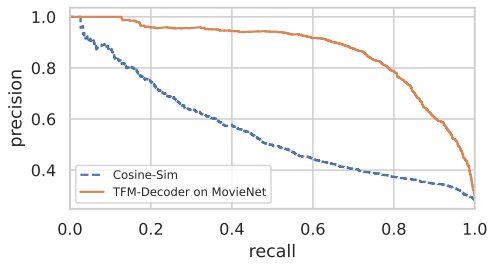


Figure 7.5: The precision-Recall curve for the character recognition methods, computed on 4 MAD-eval movies that have character annotations from MovieNet. We compare two methods: thresholding actor-movie cosine similarity, and learned transformer decoder on MovieNet. The precision/recall is calculated on a per-character basis, i.e. the precision/recall of the cosine thresholds to correctly find a character name mentioned in the AD. More baselines are described in the Appendix.

Exp.	AD Context	PT	Arch	CharBank Settings Source	Char.	Act.	Exem.	R-L	C
A1	✗	✗	Prompt	-	✗	✗	✗	9.3	6.7
A2	✗	✗	Prompt	recog.	✓	✓	✓	10.4	11.0
B1	✗	✗	X-Attn	-	✗	✗	✗	9.7	10.0
B2	✗	✗	X-Attn	recog.	✓	✗	✗	10.8	14.2
B3	✗	✗	X-Attn	recog.	✓	✓	✗	11.1	15.0
B4	✗	✗	X-Attn	recog.	✓	✓	✓	12.7	18.3
B5	✗	✗	X-Attn	full-cast	✓	✓	✓	10.9	14.9
C1	✗	AV&WV	X-Attn	recog.	✓	✓	✓	13.1	19.2
C2	✓(recurrent)	AV&WV	X-Attn	recog.	✓	✓	✓	13.4	19.5

Table 7.2: **Ablations for AD generation.** We ablate the effect of the cross-attention module and character bank, and show the effect of partial-data pretraining. All models are trained on MAD-train-`named` and evaluated on MAD-eval-`named`. Performance is reported in terms of ROUGE-L (R-L) and CIDEr (C).

7.5.2 Audio Description on GT segments

This section focuses on the effectiveness of each proposed component in the AD generation pipeline, based on the *ground-truth* AD time segments, as shown in Table 7.2.

Architecture comparison

We investigate two ways for conditioning a pre-trained and frozen generative language model (LM) with visual inputs, that is, (a) by introducing additional layers into the LM that cross-attend to the visual input, or (b) by mapping the visual input to tokens that act as prompts for the LM. Comparing rows 'B1 vs A1' and 'B4 vs A2' in Table 7.2, the architecture with newly introduced cross-attention

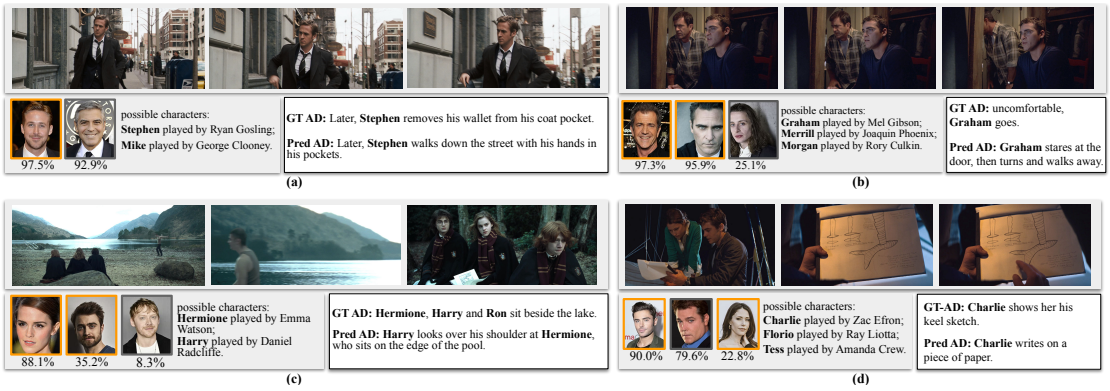


Figure 7.6: Qualitative results of our results with character bank. For a given movie clip, the character recognition module can recognize active characters on-the-fly and its results are fed into the AD generation pipeline. Note that for character recognition, we threshold the 'active' characters in the scene with a probability of 20% to encourage a higher recall. The probability shown below characters' portraits is the output of our recognition module, with correctly recognized characters marked using an orange border. For visualization purpose, we show the character's IMDb portrait image but the model actually takes in exemplar features as input. Three frames are shown for each movie clip. To illustrate the effect of character bank, this model is trained *without* context AD – that is, naming information is only from the character bank. The movies are from: **(a)**: Ides of March (2011), **(b)**: Signs (2002), **(c)**: Harry Potter and the Goblet of Fire (2005), **(d)**: Charlie St. Cloud (2010).

outperforms the prompting-based architecture both with or without the character bank inputs. The performance gain comes from greater interaction between visual and textual features by its interleaved design.

Effect of character bank

Here, we start by investigating the effect of incorporating the character bank in three different ways as discussed in Sect. 7.3.2, followed by comparing our proposed character recognition module with a naïve baseline.

Choices for exploiting character bank. By default, the model takes the predicted 'active' characters in the scene from the character recognition module. From the comparison rows 'B1-4', we can draw the following observations: (i) injecting character names gives a clear performance gain ('B2 vs B1'), highlighting the dependency of the AD task on character names; (ii) inputting additional actor names only brings marginal improvements ('B3 vs B2'), we conjecture this is because CLIP have seen large number of celebrities' picture-names pairs at pre-training stage, the visual features have thus already encoded such information; (iii) feeding in characters' exemplar features improves the performance ('B4 vs B3'), showing

Methods	ROC AUC	Average Precision
Baseline (D)	0.70	0.53
TFM (V)	0.71	0.52
TFM (V+D)	**0.78**	**0.61**
TFM (V+A+D)	0.73	0.55

Table 7.3: Results on the binary AD temporal proposal task on the `MAD-t-eval` benchmark. TFM refers to the transformer encoder architecture where V, D, and A refer to visual, duration and audio features respectively.

the complementary nature of visual-textual features; (iv) presenting the full cast list (i.e. ll 10 characters downloaded from IMDb) as character bank leads to inferior performance, as shown by comparison between B4 and B5. This is because the full cast list introduces irrelevant characters to the AD generation pipeline and harms the training, especially with a limited size of training data. This comparison also shows the necessity and effectiveness of our character recogition module, which provides a higher-quality character list to aid AD generation.

Character recognition module. We compare the proposed character recognition module (described in Fig. 7.3) with the baseline method that simply thresholds CLIP similarities between exemplar features and frame features. Specifically, we train the character recognition module on LSMDC-train movies and report results on MAD-eval set consisting of 10 movies. Since the task of recognizing active characters is a binary classification, we report ROC-AUC and Average Precision, as shown in Table 7.1, our proposed character recognition module clearly outperforms the baseline by a large margin. More details and discussion on a PR curve are provided in supplementary.

Partial-data pretraining and context

Following [T. Han et al. 2023], we also pre-train our model with partial-data, for example, AudioVault and WebVid, as well as incorporate previous AD as context for the model. The results show that AD generation can be further improved by combining these methods, showing that our newly introduced cross-attention module and character bank are orthogonal to the contributions in [T. Han et al. 2023].

7.5.3 Temporal proposal results

In Table 7.3 we compare the different time proposal classification models with the baseline threshold method. We see that CLIP visual features combined with the gap-duration feature (V+D) significantly outperforms the baseline and other modality combinations. Surprisingly we find that VGGish audio features worsen classification performance, which could be attributed to the weaker correlation compared to visual and duration.

7.5.4 Qualitative Results

Fig. 7.6 shows four qualitative examples. It shows that the character recognition module is able to recognize active characters reasonably well, and the AD generation module can associate the active characters with the descriptions. Note that even if the character recognition module proposes incorrect characters, the AD generation pipeline has learned to *ignore* such irrelevant characters for AD generation, such as 'Mike' in sample **(a)** and Morgan in sample **(b)**. Given that a large portion of AD sentences (41%) contains human identity like those samples, recognizing characters is an essential capability for high-quality AD generation. More examples in the Appendix.

7.5.5 Comparison with state-of-the-art

In Table 7.4, we report AD captioning results on the MAD-eval benchmark and achieve state-of-the-art performance by considerable margins across both the local and recurrent settings. Note that our method *without* context AD or AV/WebVid pretraining already surpasses AutoAD-I (CIDEr 18.3 vs 14.3). Adding partial data pretraining on AV/WebVid and context AD further increases the performance (CIDEr 19.5 vs 14.3).

Methods	Time window	Pretrain Data	R-L	C	R@5-16
ClipCap [Mokady et al. 2021]	local	CC3M	8.5	4.4	36.5*
AutoAD-I [T. Han et al. 2023]	local	WebVid	9.9	10.0	38.2*
AutoAD-I [T. Han et al. 2023]	local	AV & WebVid	10.3	12.1	39.8*
Ours	local	None	12.7	18.3	45.6
Ours	local	AV & WebVid	**13.1**	**19.2**	**51.3**
AutoAD-I [T. Han et al. 2023]	recurrent	AV & WebVid	11.9	14.3	42.1*
Ours	recurrent	AV & WebVid	**13.4**	**19.5**	**50.8**

Table 7.4: Comparison with other methods on `MAD-eval` benchmark under both the local (without AD context) and recurrent (with previously predicted AD as context) settings. *Denotes results re-implemented by us using the same evaluation setting.

7.6 Discussion and Future Work

Taken together this paper has proposed all the elements needed for a fully automated AD system: when to produce AD, what it should contain, and who it should describe (naming). Note these sub-tasks can probably be done jointly by using a transformer decoder with special time tokens, such as Whisper [Radford et al. 2022a] or Vid2Seq [A. Yang et al. 2023]. Predicting accurate timestamps for such architectures [Bain et al. 2023], modelling long-term dependency and leveraging multi-modal information are exciting challenges towards human-level movie understanding.

7.7 Appendix

7.7.1 Downloading cast information

As briefly described in the main paper Section 4.3, We download cast information from IMDb[6]. Specifically, we first query the movie based on its IMDb ID, e.g. tt0120780, which is provided by the datasets like AudioVault-AD [T. Han et al. 2023] or MAD [Soldan et al. 2022]. Next, we download the cast list under the HTML element '`<span>`Top Cast`</span>`', where each item in the list contains the actor name, the character name and a portrait picture of the actor. For each movie, we download such information for up to 10 characters.

[6] https://www.imdb.com/

Special Cases. Some characters in the cast list do not have corresponding portrait pictures. Among 488 movies from MAD-train, we find 293 movies have missing portrait pictures in their top-10 cast list. By manual verification, we find it is typically because the actors are less known and therefore do not have an IMDb profile page – since most of the IMDb data source is contributed by volunteers, there exists an inevitable bias towards celebrities or well-known movies. In such cases, we remove the characters in our data collection pipeline. Overall, among 488 movies from MAD-train, there are 17 movies with less than 5 characters downloaded, and one movie has an empty character list, which is *Human Flow (2017)*[7], a documentary.

7.7.2 Statistics of movie AD and subtitles

Frequency of names and pronouns. Table 7.5 and 7.6 show the frequency of names and pronouns on AD and subtitles respectively. The frequency is calculated on a per-sentence basis, that is, if any name (from Named-Entity Recognition (NER) outputs) or pronoun exists in the AD/subtitle sentence, the count is accumulated by one. The tables show that a substantial 39.1% of AD sentences contain character names, compared to only 13.3% for subtitles. Generating sentences with correct names is an important aspect of AD quality. Note that in this analysis, we discard the intro and outro of the movie for more reliable frequencies. The AD during those periods mainly performs an OCR task – introducing the producers, the name of the studio or reading movie credits at the end, which includes a large number of '[PER]' tags from the NER outputs.

Unique names within each movie. From the NER output of AD sentences, we aggregate the unique words with '[PER]' tags for each movie. For 488 movies in MAD-train, we found on average there are 69 unique names for each movie, with a maximum of 176 unique names and a minimum of 3 unique names. The number is much higher than the length of a typical cast list because (i) characters could be mentioned in different ways, e.g. by their first-name, last-name or titles, (ii) the names mentioned in AD do not correspond to characters, e.g. Gryffindor

[7] https://www.imdb.com/title/tt6573444/

from 488 **MAD-train** movies	quantity	ratio
all AD sentences	310,494	100%
AD with [PER] tag	121,557	39.1% (40.7%†)
AD with pronouns*	111,974	36.1%
AD with ([PER] tag *or* pronouns)	202,256	65.1%

Table 7.5: Frequency of names or pronouns in the **AD sentences**. The numbers are based on MAD-train movies *after removing the intro and outro* of the movies. The '[PER]' is the entity category for 'person' from NER outputs. '†': If including AD from intro and outro, the percentage of AD with [PER] tag is 40.7%, which is reported in the main paper page-4 and 8. '*': We count the occurrence of any one of six pronouns {she, her, he, him, they, them}.

from 488 **MAD-train** movies	quantity	ratio
all subtitle sentences	628,613	100%
subtitles with [PER] tag	83,904	13.3%
subtitles with pronouns*	150,564	24.0%
subtitles with ([PER] tag *or* pronouns)	216,410	34.4%

Table 7.6: Frequency of names or pronouns in the **subtitles**. The numbers are based on MAD-train movies. The '[PER]' is the entity category for 'person' from NER outputs. '*': We count the occurrence of any one of eight pronouns {she, her, he, him, they, them, i, me}.

for the college name, (iii) errors or noises of the NER pipeline that the words are partitioned incorrectly.

Visualization of AD and subtitles on the time axis. Following The Web Content Accessibility Guidelines 2.0 [Caldwell et al. 2008] (also introduced in the main paper Section 3.3), successful AD should be added during existing pauses in movie dialogues. In Figure 7.7, we visualize both ground-truth AD and movie subtitles on the timeline for 15-second and 10-minute movie clips to illustrate this interleaved property of ground-truth AD and subtitles.

Stats of inter-annotator agreement. As briefly described in Section 3.3, the timestamps of human-generated AD vary for the same movie, especially during long pauses in dialogue. On the AudioVault website, a small portion (less than 20%) of movies have more than one AD versions or multi-lingual AD versions. Figure 7.8 shows an example movie clip with its two AD versions on AudioVault-AD. Those two versions describe the same movie but are provided by annotators

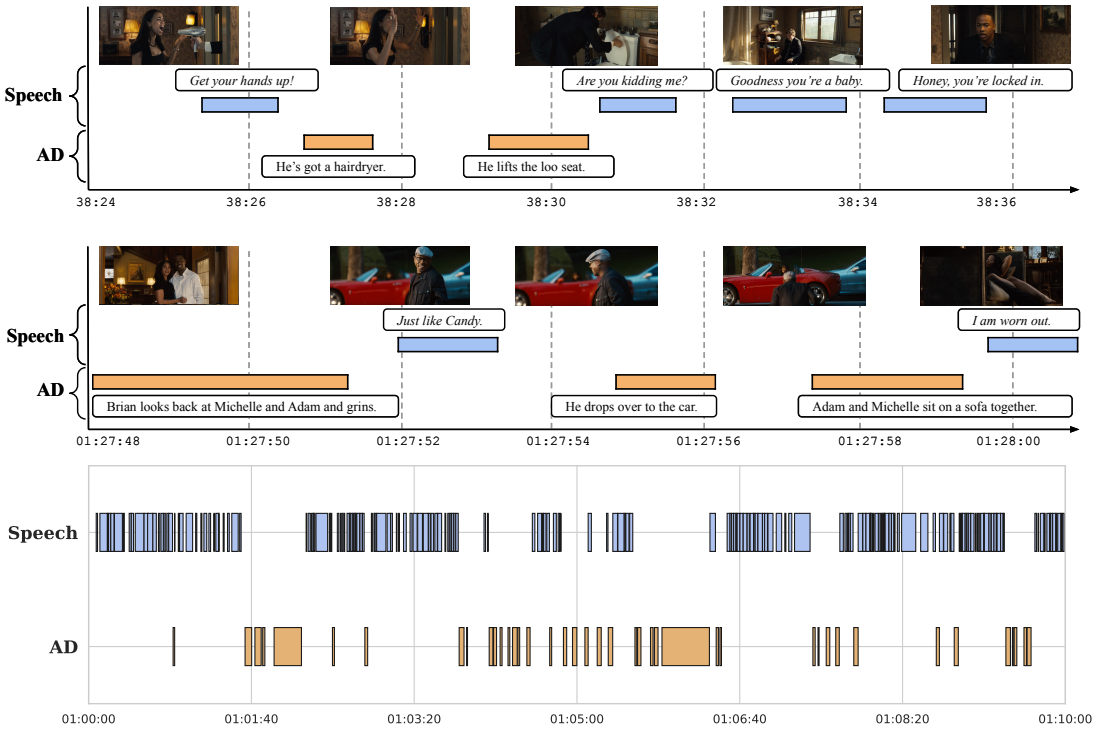


Figure 7.7: Timeline visualization of a movie with its original dialogue (speech) and human-generated Audio Description (AD). AD is inserted at appropriate times between speech, describing relevant visual elements in the frames. The top and mid figures show movie clips spanning 15 seconds with corresponding frames and texts, the bottom figure shows a movie clip spanning 10 minutes with only timestamps. The movie shown here is *Death at a Funeral (2010)* with IMDb ID tt1321509. The corresponding AD is sourced from AudioVault-AD (ID 17295).

from the US and UK respectively. Comparing the middle blocks with the lower blocks in Fig. 7.8, it can be seen that AD sentences from the two versions have different start/end timestamps (both shown in orange blocks). We also notice that character names are referred to differently in both AD versions, e.g. the AD at 20:40. Incorporating multiple versions of AD of the same movie would be an interesting research direction. In this paper, we only consider one AD version for each movie by choosing the version with a lower AudioVault ID.

7.7.3 Training details

Character recognition module

Architecture details. See Table 7.7 for the details of character recognition module.

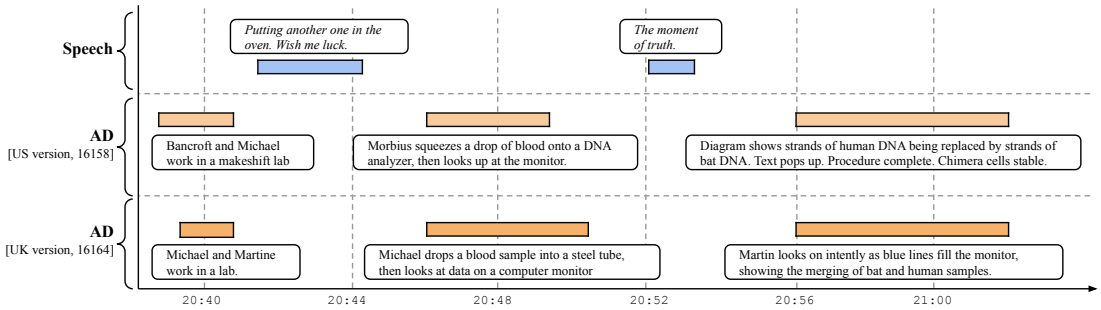


Figure 7.8: Timeline visualization of the **same movie clip** with its original dialogue (speech) and **two versions** of human-generated Audio Description (AD). Note that disagreements of timestamps exist between different versions of AD for the same movie clip. The movie clip is from *Morbius (2022)* with IMDb ID tt5108870. The two versions of AD are from AudioVault-AD with ID 16158 (US annotator) and 16164 (UK annotator). The characters who appeared in the scene are *Dr. Michael Morbius* and *Martine Bancroft*.

num blocks	2
channel	512
num head	8
ff dimension	2048

Table 7.7: The architecture details of the character recognition module, which consists of a 2-layer transformer decoder.

Training recipe. The character recognition module is trained with noisy labels acquired from AD sentences with NER outputs, as described in the main paper Section 4.1. The model is trained with AdamW optimizer with a learning rate of10^{-4} for 10 epochs with a batch size of 512 movie clips. The loss is binary cross-entropy with label balancing.

Other pretraining with partial data.

We follow [T. Han et al. 2023] for the pretraining with partial data. Specifically, we use the text-only AudioVault-AD dataset to finetune the last 6 blocks of a WebText pretrained GPT2 for 5 epochs. We also use the video-text data from WebVid to pretrain the perceiver resampler and X-Attn blocks for 5 epochs, but with GPT2 weights frozen. Both pretraining procedures can be achieved in parallel, and the trained weights from both settings can be combined as an initialization for the AD generation finetuning.

The final finetuning.

Architecture details. See Table 7.8 for the details of the perceiver resampler and X-Attn blocks.

Perceiver Resampler	project layer†	512–768
	num latent	10
	num blocks	2
	channel	768
	num head	12
	ff dimension	3072
X-Attn	num blocks	12*
	channel	768
	num head	12
	ff dimension	3072

Table 7.8: The architecture details of perceiver resampler and X-Attn blocks. †: The perceiver resampler takes 512-d CLIP visual features as input. Those features are first projected to 768-d for further computation. *: We insert 12 X-Attn blocks into 12-block GPT2-small model, that is one X-Attn block for each GPT2 block.

Training recipe. The AD generation pipeline is trained (or finetuned) on MAD-train data with a batch size of 64 movie clips for 10 epochs. We use the AdamW optimizer with a cosine-decayed learning rate schedule with a linear warm-up. The default learning rate is 10^{-4}. The GPT2 weights are frozen when training for AD generation. The trainable parameters are the perceiver resampler and the X-Attn blocks. For the textual character information (e.g. Jack played by Leonardo DiCaprio ...), we right-pad the sequences of text tokens for up to 64 tokens. For the contextual AD information, we right-pad the sequences for up to 32 tokens. For the character's exemplar features, we pad with zero values for up to 10 characters.

Temporal Proposal Classification

Architecture Details. See Table 7.9 for the details of the temporal proposal module.

Training recipe. The temporal proposal module needs a triplet of visual/audio features to make a binary decision (as described in the main paper Section 3.3) – the features before/during/after the dialogue gap. The model is trained with

project layer†	512 – 128
num blocks	6
channel	128
num head	4
ff dimension	256

Table 7.9: The architecture details of the temporal proposal classification module. †: The module takes 512-d CLIP visual features as input. Those features are first projected to 128-d for further computation.

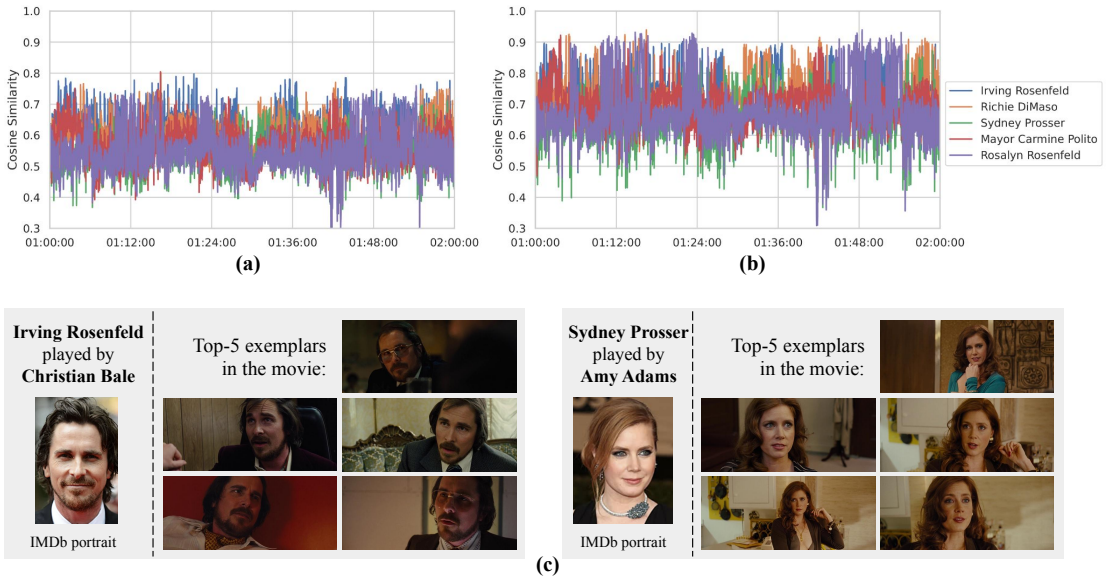


Figure 7.9: Details of calibrating cosine distance and leveraging IMDb portrait images. **(a)** Cosine similarity between actors' IMDb portrait images and the movie features *before* calibration (only a one-hour clip is shown for clarity). **(b)** Cosine similarity between characters' in-movie exemplar features with the movie features, i.e. *after* calibration. The same one-hour clip is shown. **(c)** Visualization of top-5 exemplars for two characters, which are simply obtained by taking the top-5 peaks from Fig.(a) for each actor. The movie samples are from *American Hustle (2013)* with IMDb ID tt1800241.

a batch size of 64 feature triplets for 3 epochs on MAD-train movies. We use AdamW optimizer with a learning rate of 10^{-4}.

7.7.4 Analysis

Tanh gating during training

Following Flamingo [Alayrac et al. 2022], we visualize the absolute value of tanh gating for each X-Attn block during training, which could be a rough indicator showing how much visual information is conditioned by the GPT-2 model. In contrast to Flamingo Fig. 6 that their tanh gating values are much closer to 1, our

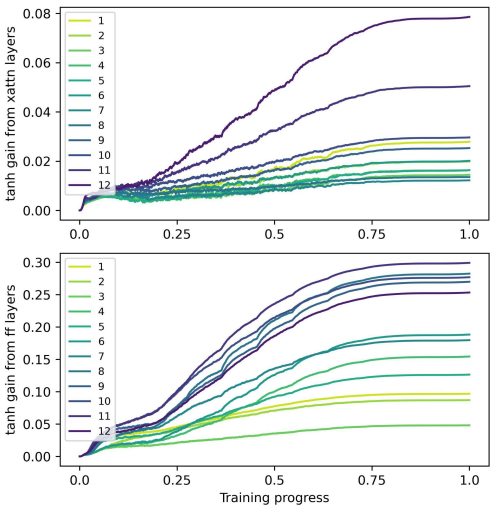


Figure 7.10: Monitoring Tanh gating during the training process. There are two Tanh gates for each X-Attn block: one for X-Attn operation and the other for feed-forward operation. Please refer to [Alayrac et al. 2022] for details. In this figure, the X-Attn blocks are trained from randomly initialized weights, thus the gating value starts from zero.

Figure 7.10 shows the tanh values have a similar increasing trend during training but the final value is much lower. It indicates a longer training schedule with a larger dataset would further benefit our model.

Character recognition module

Cosine distance and calibration. As shown in Figure 7.9-(a), the cosine similarity between actors' IMDb portrait images and the movie features is not a good indicator of in-screen or off-screen actors. For example, the peaks of the blue curve (Irving Rosenfeld) are always higher than that of the purple curve (Rosalyn Rosenfeld). As introduced in the main paper page 4, in order to compensate for the variance of appearance from IMDb portrait images, we find exemplars of the actors in the same movie as a calibration process. Figure 7.9-(c) shows two examples of exemplar searching, which is achieved by simply taking the top-5 peaks for each actor in Fig. 7.9-(a). Next, we use the averaged exemplar features to replace the original IMDb portrait features and re-compute the cosine similarity. As shown in Figure 7.9-(b), the calibration process normalizes the cosine similarity and makes the comparison between actors more meaningful.

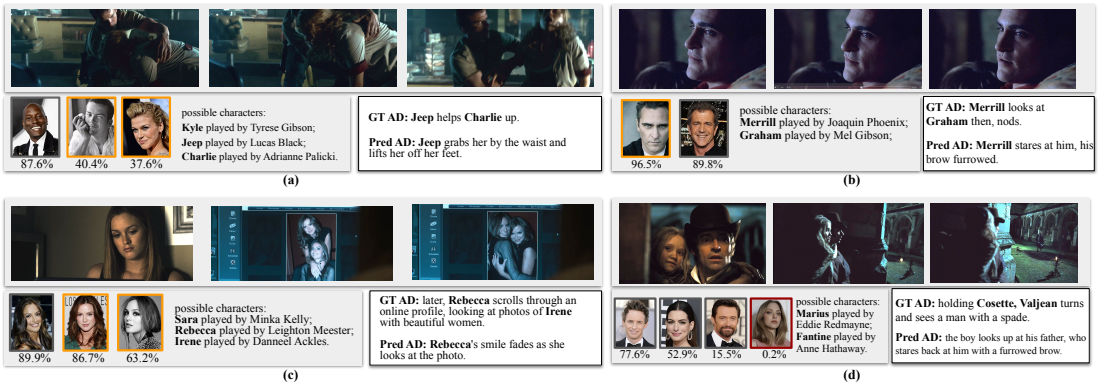


Figure 7.11: Following the same style as the main paper Figure 5, we show qualitative results with the character bank. The probability shown below the characters' portraits is the output of our character recognition module, with correctly recognized characters marked using an orange border. We use 20% as the decision boundary for active characters. The movies are from **(a)**: Legion (2010), **(b)**: Signs (2002), **(c)**: The Roommate (2011), **(d)**: Les Misérables (2012).

Other character annotation dataset. In the main paper, we use the manually annotated character annotation from the MovieNet dataset. But the character labels can also be obtained with weakly annotated data, such as the AD annotation.

We propose a dataset named **MAD-L-char** for movie character recognition, which is sourced from MAD-train and LSMDC-train. The character names in **MAD-L-char** are automatically mined in two steps: (1) running named entity recognition (NER) [Nadeau and Sekine 2007] on the AD annotation, and (2) computing the intersection with the movie's cast list. Specifically, the NER on MAD-train is sourced by running an open-sourced model [8], and the NER from 139 LSMDC-train movies can be obtained from the LSMDC annotations.

P-R curve for character recognition. In addition to the main paper Table 1 and Fig. 5, here in Fig. 7.12, we compare three PR curves: thresholding actor-movie cosine similarity, learned transformer decoder on MAD-L-char, and learned transformer decoder on MovieNet. The PR curve shows that the model trained on manually annotated MovieNet dataset clearly outperforms the same model trained on the automatically mined MAD-L-char dataset. Note that for some movies, even the top 10 characters downloaded from IMDb may not cover the main characters, such as the Harry Potter series which has a very large cast list.

[8] https://huggingface.co/Jean-Baptiste/camembert-ner

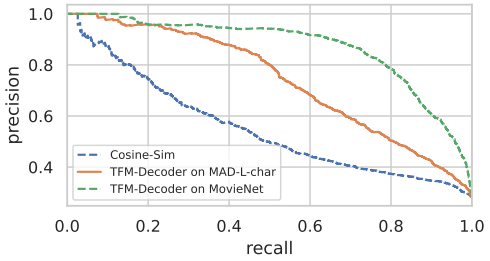


Figure 7.12: More Precision-Recall curves for the character recognition methods. We show three methods: thresholding actor-movie cosine similarity, learned transformer decoder on MAD-L-char, and learned transformer decoder on MovieNet. The precision/recall is calculated on a per-character basis, i.e. the precision/recall of the cosine thresholds to correctly find a character name mentioned in the AD. More baselines are described in the Appendix

Statistics of recognized active characters. After the character recognition module is trained, we simply choose the standard probability of 0.5 as the threshold for the decision boundary. With a threshold of 0.5, the character recognition module achieves 0.83 recall and 0.75 precision on MAD-eval movies (read from Figure 7.12). Next, this module can be used to recognize active characters in any public movie, either offline or on-the-fly. Among more than 300k AD sentences in MAD-train, the character recognition module predicts 1.3 active characters on average per AD sentence, with 94.8% AD sentences having no more than 5 predicted active characters and 14.6% AD sentences having zero active characters.

Temporal proposal classification module

For the binary temporal proposal classification task described in Section 3.3 of the main paper, we propose a simple decision-based baseline whereby any speech gap with a duration greater than a fixed threshold is classified to have AD inserted, and not AD inserted otherwise. In Table 3 of the main paper, the Average Precision and ROC AUC is calculated by varying the fixed threshold at 100 values equally spaced between 2.5 and 7.5 seconds.

Learning with subtitles

In addition to the character bank, we find feeding in subtitles as model inputs does not further improve performance. There are two possible reasons: (i) usually the subtitles do not describe the scene or characters, and (ii) the character names are

already supplied by the character bank. Leveraging movie subtitles effectively is a promising future direction.

7.7.5 More qualitative results

More qualitative results are shown in Figure 7.11. Note that in **(d)**, the girl in the scene (*young* Cosette played by Isabelle Allen) is not in the top cast whereas our top cast contains the *adult* Cosette played by Amanda Seyfried, shown in red border. Recognizing characters in such cases is challenging but it indicates the character recognition module has a large space for improvement.

Part III

Audiovisual Animal Behaviour Recognition

Chapter 8

Automated Audiovisual Behavior Recognition in Wild Primates

The paper was accepted as a journal publication at Science Advances, 2020.

Automated Audiovisual Behaviour Recognition in Wild Primates

Max Bain Arsha Nagrani Daniel Schofield

Sophie Berdugo Joana Bessa Jake Owen

Kimberley J. Hockings Tetsuro Matsuzawa Misato Hayashi

Dora Biro Susana Carvalho Andrew Zisserman

University of Oxford University of Exeter

California Institute of Technology University of Rochester

Coimbra University Algarve University

maxbain@robots.ox.ac.uk

Abstract

Large video datasets of wild animal behavior are crucial to produce longitudinal research and accelerate conservation efforts; however, large-scale behavior analyses continue to be severely constrained by time and resources. We present a deep convolutional neural network approach and fully automated pipeline to detect and track two audiovisually distinctive actions in wild chimpanzees: buttress drumming and nut cracking. Using camera trap and direct video recordings, we train action recognition models using audio and visual signatures of both behaviors, attaining high average precision (buttress drumming: 0.87 and nut cracking: 0.85), and demonstrate the potential for behavioral analysis using the automatically parsed video. Our approach produces the first automated audiovisual action recognition of wild primate behavior, setting a milestone for exploiting large datasets in ethology and conservation.

8.1 Introduction

The field of ethology seeks to understand animal behavior from both mechanistic and functional perspectives and to identify the various genetic, developmental, ecological, and social drivers of behavioral variation in the wild [Tinbergen 1963]. It is increasingly becoming a data-rich science: Technological advances in data collection, including biologgers, camera traps, and audio recorders, now allow us to capture animal behavior in an unprecedented level of detail [Jens Krause et al. 2013]. In particular, large data archives including both audio and visual information have immense potential to measure individual- and population-level variation as well as ontogenetic and cultural changes in behavior that may span large temporal and spatial scales. However, this potential often goes untapped: The training and human effort required to process large volumes of video data continue to limit the scale and depth at which behavior can be analyzed. Automating the measurement of behavior can transform ethological research, open up large-scale video archives for detailed interrogation, and be a powerful tool to monitor and protect threatened species in the wild. With rapid advances in deep learning, the novel field of computational ethology is quickly emerging at the intersection of computer science, engineering, and biology, using computer vision algorithms to process large volumes of data [D. J. Anderson and Perona 2014].

The aim of this paper is to automate animal behavior recognition in wild footage. Deep learning-based behavior recognition has thus far been shown in constrained laboratory settings [Sturman et al. 2020; van Dam et al. 2020] or using still images [Swarup et al. 2021] and has yet to be effectively demonstrated on unconstrained video footage recorded in the wild. Measuring animal behavior from wild footage presents substantial challenges—often, behaviors are hard to detect, obscured by motion blur, occlusion, vegetation, poor resolution, or lighting. If successful, then the tools would enable exploration of a multitude of research questions in ethology and conservation. Increasingly, research is revealing fine-scale variation between individuals and populations of wild animals [Kaufhold and Van Leeuwen 2019]; however, capturing this variation is often laborious and not feasible on the large scale through manual annotation. Auto-mated approaches allow us to examine in more detail the variation, through cross comparison of animal groups in a

wide variety of contexts. Detailed time series data of individual behavior enables integration of time depth perspectives into field research to more comprehensively reconstruct how behavior develops across the life span (ontogenetically) as well as examine how other processes such as social transmission, demography, and ecology interact to drive behavior change over time [Cantor et al. 2021]. These detailed behavioral data are also a crucial component of conservation research: They enable us to investigate how anthropogenic pressures such as climate change and habitat fragmentation disrupt animal behavior [Dominoni et al. 2020] (migratory patterns, foraging, reproduction, etc.) and to develop novel behavioral metrics to monitor the risks to and viability of threatened populations [Christiansen et al. 2013; Caravaggi et al. 2017]. Here, we demonstrate the potential of such an approach by developing a system for the automated classification of two distinct wild chimpanzee behaviors with idiosyncratic audiovisual features: nut cracking and buttress drumming. We also analyze pilot data of sex and age differences in percussive behaviors (nut cracking and drumming) from longitudinal archive and camera trap datasets. Chimpanzees are an ideal species for testing behavioral recognition; owing to their large fission-fusion societies, complex sociality, and behavioral flexibility, they exhibit exceptionally rich behavioral repertoires [McGrew et al. 2001]. Our target behaviors, nut cracking and buttress drumming, differ in their function—extractive tool use versus long-distance communication, respectively—but both involve percussive actions that produce distinctive sounds, i.e., the pounding of a hammer stone against a nut balanced on an anvil stone and the pounding of hands or feet against large buttress roots. Whereas nut cracking is limited to some West African and Cameroon chimpanzees (Pan troglodytes verus and Pan troglodytes ellioti), buttress drumming is a universal behavior across all chimpanzee communities [McGrew et al. 2001].In relation to previous works using deep learning, individual reidentification has been a critical first step toward full automation [Bain et al. 2019; Schofield et al. 2019], but this alone cannot capture the full complexity of behaviors that animals perform in the wild across space and time. Existing methods have used deep learning for markerless pose estimation to track the movement of animal body parts [Mathis et al. 2018], but pose estimation models perform poorly at recognizing actions using posture and limb movements alone [Shao et al. 2020]. Other approaches have used single-image analysis to identify basic activities of wild animals using tagged information from camera traps,

but these fail to capture the dynamic sequences of behavior required for detailed analysis [Norouzzadeh et al. 2018]. Recent advances in human action recognition in the field of computer vision have used three-dimensional (3D) convolutional neural net-works (CNNs) [Ji et al. 2012], which incorporate spatiotemporal information across video frames [Carreira and Zisserman 2017], but thus far have only been applied to animal species to produce broad behavioral classification limited to the visual domain [Sakib and Burghardt 2021].Given that both behaviors have strong audio and visual signatures, we recognize actions using both audio and visual streams. Our automatic framework consists of two stages: (i) body detection and tracking of individuals through the video (localization in space and time) and (ii) audiovisual action recognition (Fig. 8.1). Audio allows us to determine temporal segments where the nut cracking and buttress drumming occur ("scene level") but does not pinpoint the individual responsible. By visually detecting and tracking all chimpanzees that appear in the video, frame by frame, we are able to determine the spatial position of each individual present.

The next stage of our framework uses both the scene level audio and the visual content of each track to specify which individual is per-forming the behavior ("individual level"). Both stages in our pipe-line use a deep CNN model (see Materials and Methods).The audio stream can also be used to provide a preview mechanism to filter out behavioral sequences for human annotators to label [R. Gao et al. 2020], substantially reducing the time required to collect annotations. This is achieved using an audio-only action recognition model (which operates at the scene level) and can identify "proposals" or short video sequences where the action is likely to occur. A human annotator then only verifies whether the sequence contains the action or not. This allows us to efficiently create a labeled action recognition dataset that can be used to train the second stage of our automatic pipeline.Our method is able to identify where fine-grained movements such as striking and drumming are occurring in time and space automatically. It consists of a deep CNN model, which predicts actions using audio only, visual only, and both audio and visual modalities together. We demonstrate the use of our pipeline on two different data sources: For nut cracking, we use part of a longitudinal video archive recorded by human-operated camcorders at an "outdoor laboratory" in Bossou, Guinea [Schofield et al. 2019; Matsuzawa 1994]; while for buttress drumming, data were collected be-

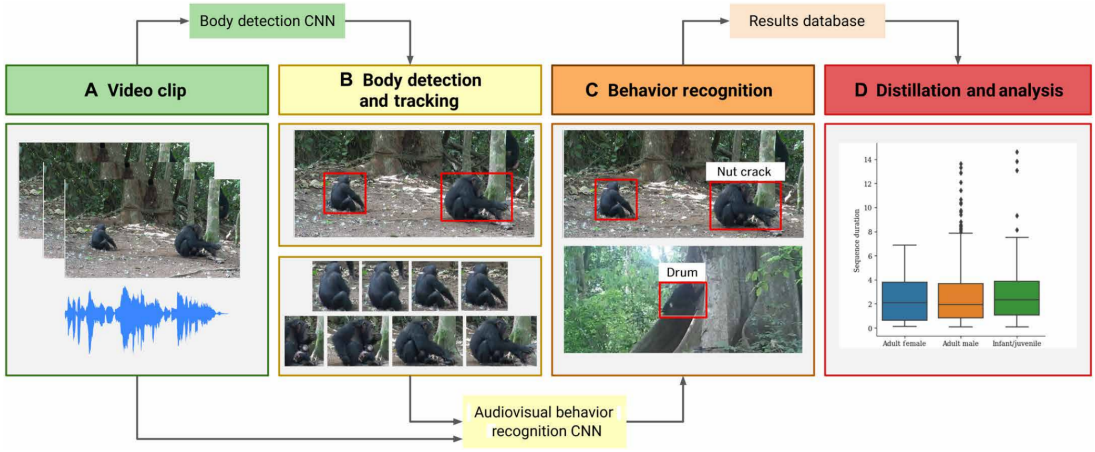


Figure 8.1: **Fully unified pipeline for wild chimpanzee behavior recognition and analysis from raw video footage.** The pipeline consists of the following stages: (**A**) Frames and audio are extracted from raw video. (**B**) Body detection is performed over the video frames using a deep CNN single-shot detector (SSD) model, and the detections are tracked using a Siamese tracker. (**C**) The body tracks are classified (e.g., is this individual cracking nuts?) using the audio data and spatiotemporal visual information for the track by a deep CNN audiovisual behavior model. The system only requires the raw video as input and produces labeled body tracks and metadata as temporal and spatial information. This automated system can be used to perform large-scale analysis (D) of behavior.

tween 2017 and 2019 by 25 motion triggered cameras in Cantanhez National Park, Guinea-Bissau [Bessa et al. 2021]. Last, we also demonstrate possible next steps in behavioral analysis enabled by the automatically parsed video. This approach represents the first automated audiovisual action recognition of species in the wild.

Table 8.1: **Recognition results for both nut cracking and buttress drumming.** We provide a baseline (random), which shows the chance performance of a random classifier. Bold indicates the highest performing method for the task.

Task	**Method**	**Average Precision**	
		Nut cracking	**Buttress Drumming**
I. Scene level	Random	0.09	0.11
	Audio	**0.85**	**0.87**
II. Individual level	Random	0.12	0.13
	Audio	0.30	0.81
	Visual	0.76	0.64
	Audiovisual	**0.77**	**0.86**

8.2 Results

For nut cracking, we apply our pipeline to 40.2 hours of video containing 2,448 nut cracking sequences (see Materials and Methods for definition of a sequence), resulting in a total of 24,700 individual body tracks (linked detections through video frames of the same 11individual; Fig. 8.1C). The training set for our model consists of data taken from three years (2004, 2008, and 2012), while we test the performance of our model entirely on data from a different year (2013) to demonstrate generalizability over time. Our audio-only nut cracking recognition CNN model obtains high average precision (85%; Table 8.1) at the scene level. Results at a scene level only detect time periods where nut cracking is being performed in the video, but they do not isolate the nut cracker, given that multiple individuals may be nut-cracking in the video at the same time (Fig. 8.2). We also predict results at an individual level, identifying whether a particular individual is nut-cracking or not. Our chimpanzee body detector achieved an average precision of 92%, and our nut cracking recognition model performed well on different poses and lighting conditions typical of videos recorded in the wild (Fig. 8.2), achieving an overall average precision of 77% at an individual level (Table 8.1 and Fig. 8.3).

For buttress drumming, 10.8 hours of camera trap footage are analyzed, resulting in a total of 1251 drumming sequences. We trained our model on data from two chimpanzee communities (Cabante and Caiquene-Cadique) and evaluated our model on manually labeled held-out test data from a third community (Lautchandé). Data from an additional community (Cambeque) are included in the analysis. Our drumming recognition CNN model achieved 87% average precision at a scene level (using audio only) and 86% average precision at an individual level (Table 8.1 and Fig. 8.3). To demonstrate the potential applications of this framework, we used the output of our automatic pipeline to further characterize nut cracking and buttress drumming behaviors.

For nut cracking, we trained a visual classifier to identify eating events: This model followed the protocols of the visual-only drumming and nut cracking classifiers and sought to identify instances when food was passed from hand to mouth (an indication of successful nut cracking). Given that an individual typically eats in conjunction with nut cracking, our audio pre-screening narrowed down the search

Figure 8.2: **Behavior recognition results demonstrate the CNN model's robustness to variations in pose, lighting, scale, and speed of action.** Example of correctly labeled body tracks from unseen and unheard videos (nut cracking and drumming for the top two and bottom two rows, respectively). Middle two rows: Multiple individuals nut-cracking and buttress-drumming showing variations in lighting, pose, background, and number of chimpanzees.

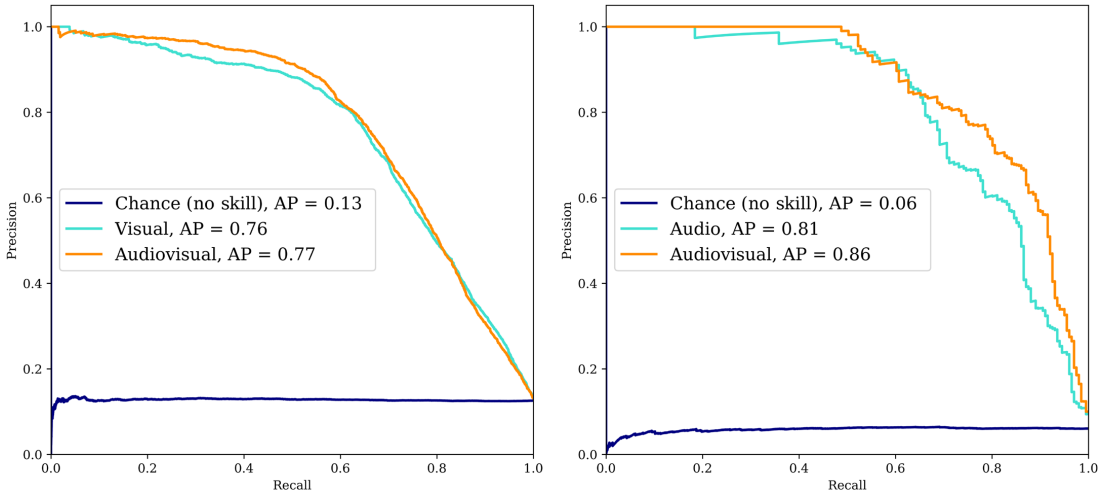


Figure 8.3: **Performance of the audio, visual, and audiovisual models for individual-level behavior classification.** The curves for nut-cracking (left) and buttress-drumming (right) demonstrate that audiovisual outperforms single-modality methods. Instances where the behavior is either visually or audibly occluded can be compensated by using the other modality (AP: Average Precision).

space, allowing us to efficiently label 896 body tracks of individuals consuming nuts. This enabled us to analyze, as a function of age/sex class, the average time spent nut cracking per eating event (a proxy for the number of nuts successfully cracked and consumed) (Fig. 8.4). For buttress drumming, we automatically detect the first and last beats of each drumming bout to precisely measure drumming bout length as a function of age/sex class, allowing us to map the distribution of drumming events throughout the day (Fig. 8.5; details of the automatic beat detection method are found in the "Analysis" section in Materials and Methods). For Bossou chimpanzees, nut cracking bouts were predominantly performed by adult males (n = 4665 bouts) followed by adult females (n = 5485 bouts) and juveniles (n = 2134 bouts), while infants (n = 1) were not observed nut-cracking. The mean time spent nut- cracking and the proportion of time spent nut-cracking differed between age/sex groups. Adult males spent a greater proportion of their time nut-cracking than adult females (males, mean ± SD = 9.21 ± 9.49%; and females, mean ± SD = 7.97 ± 9.19%), while juveniles required longer nut cracking sequences per nut consumed than adult males and females (males, mean ± SD =16.8 ± 6.46 s; females, mean ± SD =15.7 ± 10.41 s; and juveniles/infants, mean ± SD = 43.4 ± 39.0 s) (Fig. 8.5), confirming previous reports on the ontogeny of nut cracking [Biro et al. 2003]. This suggests that adult males consumed the greatest number of nuts.For buttress drumming, we analyzed 992 drumming bouts; the

majority of bouts were performed by adult males (n = 845), con-firming previous observations that this is a predominantly male activity [Arcadi et al. 1998], and occurred throughout the day, following a bimodal distribution with peaks in the morning and in the afternoon (Fig. 8.5B). When analyzing bout duration, adult males had, on average, shorter bouts (mean ± SD = 2.21 ± 1.80 s) than immature individuals (mean ± SD = 2.72 ± 2.39 s) and adult females (mean ± SD = 2.75 ± 1.79 s). There is a marked variation within each age/sex group, especially in adult males (min = 0.21 s and max = 19.48 s). In addition, drumming context (travel, feeding, and agonistic display) was analyzed for both adult males and adult females. In both groups drumming, during "travel" was the most common (509 bouts for males and 34 bouts for females), followed by "agonistic display" (225 bouts for males and 25 bouts for females), and lastly "feeding" (111 bouts for males and nine bouts for females). The proportion of drumming events for different contexts was approximately equal for both adult males and females. All drumming performed by immature individuals was done in a "play" context. Drumming bout duration varied between contexts in both adult males and adult females as well as between sexes. Feeding drumming bouts were, on average, shorter (males, mean ± SD = 1.74 ± 1.13 s; and females, mean ± SD = 2.17 ± 1.09 s) than agonistic display (males, mean ± SD = 2.31 ± 2.61 s; and females, mean ± SD = 2.78 ± 1.78 s) and travel drumming bouts (males, mean ± SD = 2.27 ± 1.35 s; and females, mean ± SD = 2.89 ± 1.97 s).

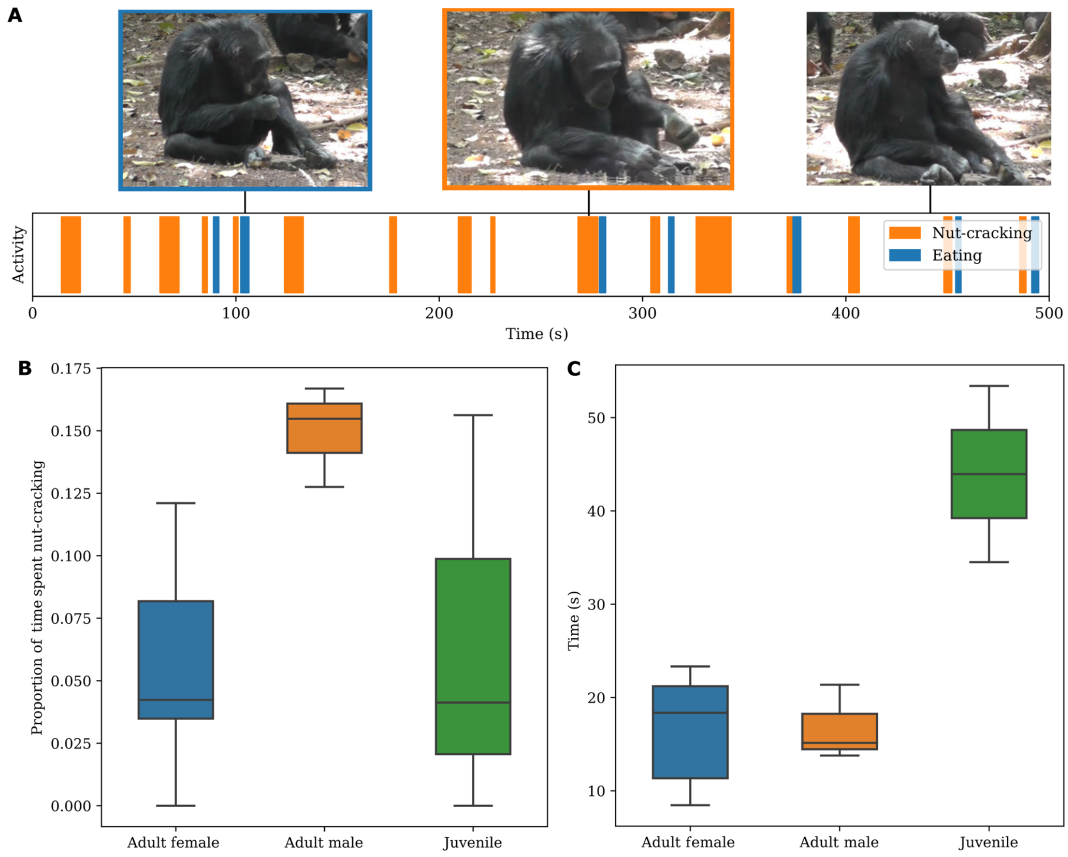


Figure 8.4: **Nut cracking analysis.** (**A**) An example activity sequence following a single individual over the course of a video. The blank white spaces are any activities that are not nut cracking or eating. Note that eating typically follows nut cracking events. (**B**) Proportion of time spent nut cracking as a fraction of total time visible. (**C**) Average time spent nut cracking per eating event. Computed by dividing the cumulative time spent nut-cracking over the total number of eating events as a function of age and sex.

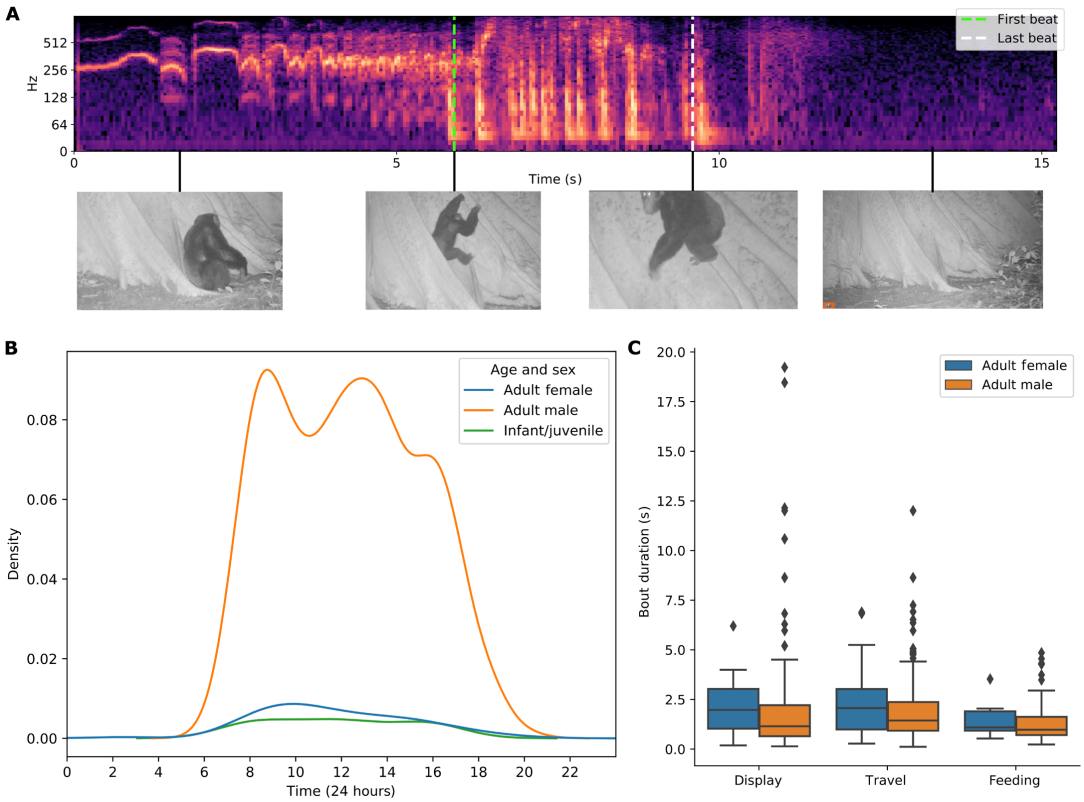


Figure 8.5: **Buttress drumming analysis.** (**A**) Spectrogram showing a detected drumming bout delineated by the first and last beats, with video frames visualized. (**B**) Kernel density estimation plot showing the diel distribution of buttress drumming bouts, based on hh:mm:ss data captured by camera traps. (**C**) Duration (in seconds) of buttress drumming bouts by context (agonistic display, travel, and feeding) and by age/sex (adult females and adult males).

8.3 Discussion

Overall, our model demonstrates the efficacy of using deep neural network architectures for a biological application: the automated recognition of percussive behaviors in a wild primate. Unlike older, rule-based automation methods, our method is entirely based on deep learning and is data-driven. It also improves on previous single-frame methods by being video-based: It uses 3D convolutions in time [Ji et al. 2012] to reason about temporal information, which is important for action detections, and exploits the multimodality of video to use the audio and visual streams jointly to classify behaviors.Often, it is challenging to curate datasets large enough to train action recognition models without sifting through a significant amount of footage (20 and 4.1% of footage yielded our behaviors of interest for the nut cracking and buttress drumming data, respectively). A key aspect of our approach is the use of audio as a prescreening mechanism, which substantially cuts down the large search space of video for annotation.Furthermore, we do not constrain the video data in any way, as is done commonly for deep learning methods applied to primate recognition and analysis by aligning individual detections or selecting for age, resolution, or lighting [Deb et al. 2018]. Instead, we are able to perform the task "in the wild" and ensure an end-to-end pipeline that will work on raw video with minimum preprocessing (Fig. 8.2). We also demonstrate that our method is applicable to both long-term targeted field video recordings (including in a field experimental setting) and to remote monitoring camera trap datasets, demonstrating its usefulness across different data collection protocols.The pipeline can be applied to data where only audio or only visual information is available (e.g., camera trap recordings where the behavior occurs off-screen, video recordings with noisy or corrupted audio, or microphone-only recordings). The benefits of our audio method are that it is not affected by visual distractors such as lighting, pose, size, and occlusion and is also computationally cheaper to run. Certain actions (such as buttress drumming) are also more discriminative in the audio space than the visual space (Table 8.1) and hence require less training data. Audio also allows greater coverage, by detecting actions beyond the field of view of the camera. Our visual-only method, on the other hand, provides the added benefit of allowing localization at an individual level, predicting which individuals are performing particular actions. This is a key advantage for

potential future applications in, for example, the monitoring of individual behavior and welfare, both in the wild and in captive set-tings. Our audiovisual model combines the benefits of both modalities. For drumming, we demonstrate that our model works well even on camera trap data from locations unseen by our model during training (which therefore might contain different tree species) and communities of chimpanzees. Our model's performance demonstrates the effectiveness of using multimodal deep learning for behavioral recognition of individual animals in longitudinal video archives and camera trap datasets in the wild. Using a novel combination of data collection methods (automated classifiers and manual annotations) and video datasets (archival footage and camera traps), we validate our approach by reproducing known findings on the ontogeny of nut cracking from the existing literature [Biro et al. 2003] and go further to gain preliminary insights into drumming behavior in unhabituated communities as well as revealing potential sex and age differences in different contexts (previously neglected in published work). Ultimately, the integration of computer vision and ethology using automated behavior recognition can aid behavioral research and conservation, moving beyond inferences of social structure and demographics that can be inferred using individual identification [e.g. [Schofield et al. 2019]] to capturing the full complexity and dynamics of social interactions and behaviors. Typically, the time and resources required for manual data collection of multiple behaviors (either through in situ observation or retrospective video coding) prohibits analysis of large scale datasets. Adopting automated behavioral recognition is scalable, increasing the speed, quantity, and detail of data that can be collected and analyzed. Once classifiers have been trained, such work can move beyond broad classification of general behavioral states (eating, resting, etc.) to include fine-grained analysis at multiple layers/dimensions of behavior [K. Huang et al. 2021] —for example, using pose estimation to quantify postural kinematics or detect the number and order of elements in a behavioral sequence (e.g., nut cracking strikes) or investigating temporal co-occurrences between the behavior of individuals in the same group. We also envisage that our method could have a large impact in conservation science. Anthropogenic pressures are increasingly affecting animal behavior, with habitat fragmentation and population loss posing an imminent threat to "cultural species" through the erosion of behavioral diversity [Kühl et al. 2019]. Automating the measurement of behavioral diversity and activity budgets could

be crucial for developing more sophisticated metrics to monitor the health and stability of wild populations [Christiansen et al. 2013].There are some limitations to our study, notably that the audio preview step is limited to actions that contain a distinctive sound (such as percussion). Nonetheless, none of the pipeline steps are specific to primate behavior, and the method can be readily applied to other animal species and behaviors. Furthermore, behaviors that are audio distinctive exist in multiple domains, and we envisage possible applications for our pipeline in, for example, marine and terrestrial animal communication (vocalizations), movement (wing flapping and stepping), self-maintenance (scratching), aggression (hitting, slapping, and screaming), and foraging (tearing, smashing, and chewing). These analyses could be performed on data not only from remote sensors but also from animal-borne audio-only biologgers. Another limitation concerns the fact that, for individual-level recognition, our method is heavily reliant on the performance of the body detector: Individuals that are not detected or tracked can-not have their behavior classified. For example, the detector often fails to detect infants on their mother's backs, although for our present analyses, this poses no problem, because young chimpanzees do not nut-crack or buttress-drum while being carried. For behaviors that specifically require a visual classifier (such as successful nut cracking being identified through the hand-to-mouth motion of eating), visual occlusion or motion blur poses challenges. However, we note that the body detector has far fewer missed individual detections than other methods that are reliant on face detection [Schofield et al. 2019; Matsuzawa 1994]. Future directions to improve our pipeline include adopting active learning, which minimizes annotator effort by automatically selecting informative samples from a pool of unannotated data for a human to an-notate to retrain the network [Norouzzadeh et al. 2021]. In addition, self-supervised learning enables label-free pretraining, initializing the model in such a way that reduces the annotation requirement for training [T. Han et al. 2019]. Our pipeline provides a critical first step in large-volume auto-mated behavioral coding and represents a breakthrough in measuring behavior. It will permit detailed intraindividual, interindividual, and cross-site comparisons, automated collection of activity budgets, and longitudinal studies of behavior at individual and population levels, enabling detailed investigation into ontogeny, cultural evolution, and the persistence/decline of behavioral variation over time and how these relate to environmental change. It has transformative potential to science, setting

a milestone for exploiting large datasets in ethology and conservation.

8.4 Materials and Methods

8.4.1 Video Archive

Description of Actions

Nut cracking has been described as the most complex tool-use behavior in wild chimpanzees, with the nut cracker typically combining three objects [Matsuzawa et al. 2011; Carvalho et al. 2008]. It involves placing a hard-shelled nut on an anvil and then using a hammer to pound the nut until the edible kernel is exposed—sometimes one or two wedges are used to stabilize the anvil. We defined nut cracking "sequences" as beginning when the hammer is raised before the initial strike of a nut and ending when the hammer makes contact with the nut or anvil for the final time before the nut is consumed or abandoned or the camera is moved away. Sequences often included multiple strikes for a single nut.Buttress drumming is a universal and frequent behavior across all chimpanzee communities, but there is much left to understand about its functions and potential cross-community variation. Drumming occurs when a chimpanzee slaps or stamps rhythmically on the buttress of a tree, often accompanied by a distinct vocalization called a pant hoot. Multiple functions of drumming have been proposed, including long-distance communication [Arcadi et al. 1998; Boesch 1991]and intimidation accompanying agonistic displays [Nishida 2011]. Distinctive individual drumming patterns and pant hoot vocalizations are thought to act as signals that coordinate group movement and distribution when traveling, as well as containing information about the individual's identity [Babiszewska et al. 2015]. These distinct drumming patterns have been described for both males and females [V. Reynolds 2005], but male chimpanzees appear to drum more frequently when traveling [Babiszewska et al. 2015]. We defined buttress drumming sequences as beginning when the first beat was detected and ending with the last beat; any behavior, such as pant hoots, occurring immediately before the first beat or immediately after the last beat were not included. Beats were detected visually, when at least one hand or foot was in

contact with the buttress, and auditorily, when the distinct beat sound was heard.

Structure of the data

Nut cracking at Bossou, Guinea. Data used were collected in the Bossou forest, southeastern Guinea, West Africa, a long-term chimpanzee field site established by Kyoto University in 1976 [Schofield et al. 2019]. Bossou is home to an outdoor laboratory: A natural forest clearing (7 m by 20 m) located in the core of the Bossou chimpanzees' home range (07°39'N and 008°30'W) where raw materials for tool use—stones and nuts—are provisioned, and the same group has been recorded since 1988 [Schofield et al. 2019; Matsuzawa 1994]. The use of standardized video recording over many field seasons has led to the accumulation of more than 30 years of video data, providing unique opportunities to analyze chimpanzee behavior over multiple generations. In total, we analyzed 43.1 hours of video footage.

Buttress drumming in Cantanhez National Park, Guinea-Bissau. Data used were collected by camera traps (n = 25) deployed in the home ranges of four different communities (Caiquene-Cadique, Lautchandé, Cambeque, and Cabante) in Cantanhez National Park, Southern Guinea-Bissau, West Africa (11°14'17.2"N and 15°02'16.9"W) be-tween February 2017 and December 2018. Chimpanzees in Cantanhez National Park inhabit an agroforest landscape and are not habituated to researchers. The camera traps were set up in areas that chimpanzees' frequented and pointed to trees with large buttress roots with clear signs of wear from chimpanzee buttress drumming. Some cameras were moved during the study period to account for seasonal changes in chimpanzee ranging patterns and when a new area of interest was located. Cameras were motion sensitive and were set to record 1-min video clips when triggered. Approximately 41,000 video clips were collected over the study period, of which 4745 contained footage of chimpanzees, spanning a total of 47.2 hours of video footage.

Dataset splits: Training, testing, and analysis. We divide the data-set into different sections. Part of the data is manually annotated by human annotators, which provides data for training and testing our automated framework (described in the "Methods" section). The remaining data are unlabeled by humans. Our framework is applied to these unlabeled data automatically (this stage is referred

to as inference) for analysis (described in the "Analysis" section). Dataset statistics are provided in the original journal publication.

8.4.2 Methods

Our pipeline for the detection of audio-discriminative percussive behaviors consists of the following two stages: (i) chimpanzee detection and tracking and (ii) audiovisual action recognition (Fig. 8.2). To efficiently collect annotations for the second stage (ii) audio-visual action recognition, we also use an additional "audio preview stage" (described below in the "Audio action recognition" section) only when collecting training data. This optional audio preview stage uses audio only to determine temporal segments where the behaviors (nut cracking and drumming) occur at a scene level. This markedly reduces the total search space of the video, allowing for efficient annotations, used to train a model on both scene-level audio and the visual content of each track to determine which individual is carrying out the behavior.With this trained model, our pipeline can then be applied directly to previously unseen videos without any human input. At this point, we do not require the audio preview stage and only use (i) detection and tracking and (ii) audiovisual action recognition.All stages in the method are implemented using deep CNNs. For audio previewing, we train a CNN on the spectrogram image of the audio. For detection, we use a single-shot detector (SSD) object category detector [W. Liu et al. 2016] to detect individuals. The detections for an individual are then grouped across frames (time) using a pretrained tracker. The final audiovisual action recognition stage involves a spatiotemporal CNN for the visual features and a spectrogram CNN for the audio. The training data were obtained by using the VGG Image Annotator (VIA) annotation tool [Dutta and Zisserman 2019]. We provide a detailed description for each stage of the pipeline in the following sections and then describe how the analysis is carried out given the detected behaviors.

Audio Action Recognition

With the audio data alone, our framework is able to classify actions at the scene level. The nut cracking and buttress drumming audio classifier achieved 85 and

87% average precision, respectively, on unseen test data (Table 8.1).Network architecture. For the audio model, we use a 2D CNN (ResNet-18), pretrained on VGGSound [Honglie Chen et al. 2020]. The output is passed through two linear layers and then a final predictive layer with two neurons and a softmax activation function, resulting in a binary classifier for each target action.Inputs. We use short-term magnitude spectrograms as input to a ResNet-18 model. All audio is first converted to single-channel, 16-bit streams at a 16-kHz sampling rate for consistency. Spectrograms are then generated in a sliding window fashion using a hamming window with a width of 32 ms and a hop of 10 ms, with a 512-point fast Fourier transform. This gives spectrograms a size of 257 $\times$ 201 for 3 s of audio. The resulting spectrogram is integrated into 64 mel-spaced frequency bins with a minimum frequency of 125 Hz and a maximum frequency of 7.5 kHz, and the magnitude of each bin is log-transformed. This gives log mel spectrogram patches of 64 $\times$ 201 bins, used as input to the CNN.Augmentations. Temporal jittering of 0.5 s is used as well as augmentation to positive samples by randomly adding background audio samples (audio that does not contain nut cracking and buttress drumming).Training. Binary cross-entropy is used as the training objective, along with an Adam optimizer with a learning rate of 5×10^{-3}.Audio preview for manual annotation. Videos in the wild (including from camera traps) contain a lot of dead footage, where the actions of interest may be captured rarely. Manually searching through all this footage is a labor-intensive task. Hence, we use an inexpensive and computationally efficient prescreening method to automatically sift through many hours of footage, proposing short videos that contain the action and discarding the rest. This is done using the audio alone, because our actions of interest are all percussive and make a distinct sound.The audio model is applied using a sliding window of size 3 s, with a stride of 0.5 s over the raw video footage. This produces a probability score P(action) of the action of interest being present within each temporal window. We then use the most confident 7% of windows (using the probability score as the confidence) for discrete video labeling, resulting in 2418 discrete, 3-second long video proposals to be annotated. The more expensive body detection and tracking is performed only on these "audio proposals." The body tracks are visualized on the proposals, allowing the annotators to label each actor in the proposal with a binary label denoting whether or not they are performing the action. Given that the drumming video footage is

already segmented into short clips and annotated, the audio preview step was not required for the buttress drumming data at training time, so it was only used for nut cracking here.At inference time, the audio preview can be used as a filtering step first before the full framework, providing computation savings. Because audio is much cheaper computationally than the full framework (detection, tracking, and audiovisual classification), this can be useful in resource-constrained environments such as running the framework on the camera traps themselves. Because this work was not constrained in terms of compute, we did not use the audio preview step at inference. For buttress drumming, the trade-off is minimal; a computation saving of 64% still captures 97% of drumming events. For nut cracking, the trade-off is greater; a computation saving of 64% captures 70% of nut cracking events. As there are many off-screen nut cracking events, the sound of nut cracking is not definitively on-screen.

8.4.3 Visual detection and tracking

A prerequisite for our method of automated detection of primate behavior is the detection and tracking of the target animal, producing spatio-temporal tracks following individuals through time. Deep learning has proved to be highly successful at object detection and tracking, and previous works describe the protocol and results of this applied to footage of wild animals [Bain et al. 2019; Schofield et al. 2019; Jiwen Yu et al. 2019; P. Chen et al. 2020]. In more detail, we follow the same protocol as in [Bain et al. 2019], which involves fine-tuning an SSD object detector [W. Liu et al. 2016] on bounding box annotations of chimpanzee bodies. Because the two datasets contain very different sources of footage, including camera traps for drumming and direct longitudinal recordings for nut cracking (the former containing night vision, varied lighting, and out of focus blur; with the latter having higher quality video but consisting of close-ups as well as medium shots), we separately fine-tune the two object detectors, one for each dataset. For the nut cracking dataset, we fine-tune on 16,000 bounding box annotations across 5513 video frames. For the buttress drumming dataset, we fine-tune on 2200 bounding box annotations across 2137 video frames. All video frames were sampled every 10 s.

Tracking. The object tracker used to link the resulting detections through time is a pretrained Siamese network. Pairs of detections in consecutive frames with a Jaccard overlap greater than 0.5 are given as input to the network. Detection pairs with a similarity score greater than 0.5 are deemed to be from the same track.

Evaluation for the detectors. Evaluation is performed on a held-out test set using the standard protocol outlined in [Everingham et al. 2010]. The precision-recall curve is computed from a method's ranked output. Recall is defined as the proportion of all positive examples above a given rank, while precision is the proportion of all examples above that rank, which are from the positive class. For the purpose of our task, high recall is more important than high precision (i.e., false positives are less dangerous than false negatives) to ensure that no chimpanzees are missed. The Bossou and Cantanhez detectors achieved average precision scores of 0.92 and 0.91 on their respective test sets.

Programming implementation details. The detector was implemented using the machine learning library PyTorch and trained on two Titan X Graphical Processing Units (GPUs) for 20 epochs (where 1 epoch consists of an entire pass through the training set) using a batch size of 32 and two sub-batches. Flip, zoom, path, and distort augmentation was used during preprocessing with a zoom factor of 4. The ratio of negatives to positives while training was 3, and the overlap threshold was 0.5. The detector was trained without batch normalization. The tracker was also implemented in PyTorch.

8.4.4 Audio-visual action recognition

Network architecture. For the visual stream, we use a 3D ResNet-18, with 3D convolutions (30). The output is passed through two linear layers and then a final predictive layer, with two neurons and a soft-max activation function. For the audiovisual fusion model, 512 dimensional embeddings from the ResNet backbone in each stream are concatenated and then passed to the final predictive layer, with two neurons and a softmax activation function.

Inputs. For the audio stream, the preprocessing is identical to the "Audio action recognition" stage. Video frames are sampled at 25 frames per second, and all detections are resized to 128 × 128—we feed in 40 frames over 2.5 s, with three

red-green-blue channels each, sampled randomly during training and uniformly during inference. This gives final inputs of size $40 \times 128 \times 128 \times 3$.

Augmentations. Standard augmentation techniques are applied to the visual inputs: color jittering, random cropping, and horizontal flipping. For the audio, we repeat the augmentations in the "Audio action recognition" section.

Training. All models are trained with a binary cross-entropy loss. In this stage, we use the annotations obtained from the "Audio action recognition" stage of the pipeline to train the model.

Evaluation. Evaluation for the action recognition models is per-formed on a held-out test set, the statistics of which are supplied in Table 1. The audiovisual fusion model performed the best at the individual level for both nut cracking and buttress drumming (77 and 86%, respectively), demonstrating its robustness across domains and actions and demonstrating its efficacy over audio or vision alone.

The models are evaluated on their precision recall at either the scene level or individual level. For the scene level, we evaluate the audio-only model with a stride of 0.5 s and a forgiveness collar of 0.5 s. For the individual level, we evaluate the audio, visual, and audiovisual models with a stride of 0.5 s per track and a forgiveness collar of 0.5 s.

Implementation details. The networks for action recognition were trained on four Titan X GPUs for 20 epochs using a batch size of 16. We trained both models end to end via stochastic gradient descent with momentum (0.9) weight decay (5×10^{-4}) and a logarithmically decaying learning rate (initialized to 10^{-2} and decaying to 10^{-8}). The visual stream is initialized with weights from [T. Han et al. 2019], and the audio model is initialized with weights pretrained on VGG-Sound [Honglie Chen et al. 2020].

Action-specific implementation details

Nut cracking analysis: Success detection. To further analyze nut cracking behaviors, we additionally measure another action: passing food from hand to mouth, which is an indication of successful nut cracking. Here, the shell has been successfully cracked and the individual passes the kernel to their mouth using their

hand; hence-forth, this action is referred to as "eating." Because this behavior has a strong visual signature, we train a visual classifier to determine this. This model follows the protocol of the visual-only drumming and nut cracking classifiers. The training labels for eating events were gathered from the audio preview proposals, totaling 896 track-lets of individuals eating. While the audio preview searches for nut cracking, eating is often found shortly after successful nut cracking events, so the short audio proposals often contain this action as well. Furthermore, individuals often nut-crack together, resulting in multiple individuals in a video proposal. Training the model on data from 2004 and 2008 results in 89% accuracy in classifying eating on unseen tracks from 2012.

Buttress drumming duration analysis. We investigate the duration of drumming bouts by determining the start and the end beat in a drumming bout using audio-based beat detection. Beat detection is performed in an automated fashion by using low-pass filtering and onset detection to the audio signal of the drumming bout. The audio sequence is first low pass–filtered using a Butterworth filter with a cutoff frequency of 800 Hz. Onset detection is then per-formed on the filtered audio waveform. We use the onset detection method provided by the Librosa Python toolbox. The hyperparameters were chosen to achieve the best beat counting accuracy on 30 drumming bouts hand-labeled with the number of beats. During evaluation, we apply a forgiveness collar of 0.25 s on either side of the drumming event boundaries to be more lenient toward imprecise boundary annotation. From the beat detections, we define the duration of a drumming bouts to be the interval between the first and last beats. This beat detection method predicts drumming duration with a mean and median error of 0.205 and 0.131 s, respectively.

Acknowledgements: Acknowledgments: We are grateful to Kyoto University's Primate Research Institute for leading the Bossou Archive Project and supporting the research presented here and to the IREB and DNRSIT of Guinea. This study is dedicated to all the researchers and field assistants who have collected data in Bossou since 1988. We thank the Instituto da Biodiversidade e das Áreas Protegidas (IBAP) for their permission to conduct research in Guinea-Bissau and for logistical support, research assistants and local guides for assisting with data collection, and local leaders for granting us permission to conduct research. We thank M. Ramon for collecting camera trap data in Cabante, Guinea-Bissau. Funding: This

study was supported by EPSRC Programme Grants Seebibyte EP/M013774/1 and Visual AI EP/T028572/1; Google PhD Fellowship (to A.N.); Clarendon Fund (to D.S. and S.B.); Boise Trust Fund (to D.S., S.B., and J.B.); Wolfson College, University of Oxford (to D.S.); Keble College Sloane-Robinson Clarendon Scholarship, University of Oxford (to S.B.); Fundação para a Ciência e a Tecnologia, Portugal SFRH/BD/108185/2015 (to J.B.); Templeton World Charity Foundation grant no. TWCF0316 (to D.B.); National Geographic Society (to S.C.); St Hugh's College, University of Oxford (to S.C.); Kyoto University Primate Research Institute for Cooperative Research Program (to M.H. and D.S.); MEXT-JSPS (no. 16H06283), LGP-U04, the Japan Society for the Promotion of Science (to T.M.); and Darwin Initiative funding grant number 26-018 (to K.J.H.). Author contributions:Conceptualization: D.S. Methodology: M.B., A.N., and A.Z. Data curation: M.B., D.S., J.B., S.B., and J.O. Data collection: D.B., S.C., T.M., M.H., K.J.H., and J.B. Software, formal analysis, and visualization: M.B. Supervision: A.Z., D.B., and S.C. Writing (original draft): M.B., A.N., D.S., and J.B. Writing (review and editing): A.Z., D.B., S.C., and K.J.H. Competing interests: The authors declare that they have no competing interests.

Chapter 9

Discussion

This chapter begins with a summary of the principal accomplishments and an emphasis on the significance of the research delineated in this thesis (Section 9.1). Following this, potential directions for future exploration and study are proposed (Section 9.2).

9.1 Achievements and Impact

Joint Video-Text Representations for Retrieval. In Chapter 2, text-to-video retrieval is explored using the proposed Condensed Movies Dataset (CMD). The study finds character recognition and longer context derived from the full movie improves retrieval performance of captioned clips. The dataset, which is publicly available, is the largest movie dataset in terms of number of hours and movies. It has also been released as a challenge at an ICCV 2021 vision and language workshop[1] to evaluate long-form text-to-video retrieval. As of July 2023, the workshop challenge has recieved submissinos from ten teams, and multiple papers on long-form video-text understanding have cited its result in the leaderboard [Croitoru et al. 2021; Yuchong Sun et al. 2022]. CMD has stimulated a number of derivate works, including re-purposing the dataset for classification tasks such as scene recognition [Bose et al. 2023], genre classification [C.-Y. Wu and Krahenbuhl 2021b] as well as additional classification of metadata labels [C.-Y. Wu and Krahenbuhl 2021b]. A benchmark suite of these long-form video classification

[1] https://www.robots.ox.ac.uk/ vgg/data/condensed-movies/challenge.html

tasks has been established by [C.-Y. Wu and Krahenbuhl 2021b].

In Chapter 3, a dual-stream visual-text encoder is proposed that can be jointly trained on both captioned images and videos. The paper posits that an image, essentially a single-frame video 'frozen in time', can be utilised for video learning via a curriculum based on the number of frames. It demonstrates that more than 85% of pretraining iterations can be performed on single frame videos while achieving performance comparable to standard training on multiple frames for all iterations. The model resulting from this approach attained state-of-the-art performance across a range of benchmarks: MSR-VTT, MSVD, DiDeMo and LSMDC.

Further the proposed WebVid dataset of 2.5 million captioned videos, and its extension WebVid10M, represents the largest publicly available cleanly-captioned video dataset of their kind is the largest public video captioned dataset of its kind. The WebVid dataset has become the standard dataset for text-to-video generation training, including Meta's Make-A-Video [Singer et al. 2022] and others [Hong et al. 2022; Fu et al. 2023; Z. Luo et al. 2023]. Text-to-video generation, a long-standing key goal of computer vision, is closely linked with video understanding due to the requirement of temporal structure, partircularly for long-form content. The paper and its accompanying dataset have garnered more than 330 citations since their publication, as of June 2023.

In Chapter 4, a technical report is introduced that presents a simple but effective baseline for achieving state-of-the-art in text-to-video retrieval. This work reveals value in using the text query to guide the frame aggregation to compute a text-video similarity score – a strategy that has subsequently inspired numerous studies [Kahatapitiya et al. 2023; L. D. Tran et al. 2023; Jin et al. 2023]. Given that this temporal aggregation approach is parameter-less, these findings highlight the lack of necessary temporal modelling in the long video-text retrieval benchmarks, thereby indicating an insufficient capture of "video understanding". Consequently, this report has motivated further investigation within this thesis into video captioning as a superior evaluation task.

Automated Movie Audio Description. To expedite the development of learning from audio descriptions (AD), Chapter 5 introduces *WhisperX*. This method aims to (i) speed-up auto-regressive speech recognition models, in this case Whis-

per, via batched inference; and (ii) provide accurate word-level timestamps. As a result, speech transcription experiences a 12-fold speedup. The open-source GitHub repository[2], which has accumulated a total of 3,700 stars as of July 2023, is employed across a broad range of industrial applications and research areas including speech-based health sensing methods [Favaro et al. 2023; Gómez-Zaragozá et al. 2023].

In Chapter 6, AD data from over 8,000 movies is collected using the *WhisperX* pipeline in conjunction with speaker diarization. The pipeline also facilitates substantial enhancements to the annotations of the existing MAD dataset [Soldan et al. 2022]. The observations underscore the value of extensive contextual information for the AD captioning task and the advantages of partial pretraining on text-only corpora. Building on these, Chapter 7 addresses some of the limitations of the preliminary AD work, including character naming and time point prediction. These pioneering studies mark the first steps toward the automatic generation of movie audio descriptions – a process that is extremely costly to annotate manually and is currently available for only a fraction of online video data.

Audiovisual Action Recognition in Wild Primates. Chapter 8 addresses the training of an audiovisual action classifier designed to predict behavioural activities in wild primates. This challenging application of video understanding is demonstrated to benefit from both RGB and audio modalities. The resultant predictions offer valuable behavioural analysis insights, which can be instrumental in wildlife monitoring research.

9.2 Future Works

Lastly, the thesis concludes by outlining promising future research directions.

Audiovisual transcription, diarization, and identification. For comprehensive understanding of human-centric video, a system must be capable of identifying each human and understanding their communication (speech). This goes beyond simple audio-based transcription and voice clustering. Voice and face identity banks could be utilized for low-shot identification of individuals across video and

[2] https://github.com/m-bain/whisperX

audio streams. Such a system could be generalized to any type of fine-grained object identification that benefits video understanding. This forms a critical foundation for reasoning about human-centric video data, such as movies.

Modelling everything as language. Inspired by the recent success of LLM-based reasoning systems, such as ViperGPT [Gupta and Kembhavi 2023] and VisProg [Surıs et al. 2023] for visual question-answering, a promising direction for video understanding and automated audio descriptions lies in representing each constituent modality with language. This approach might involve the dense description of each frame using (i) image captioning models; (ii) labels from action classification models; and (iii) face recognition models. Likewise, the audio can be densely described with speech recognition outputs, and labels from audio captioning or classification models. An LLM can then process these dense linguistic descriptors for cross-modal and long-form reasoning. Furthermore, video metadata, such as a plot synopsis, can be seamlessly integrated as supplementary information. This approach capitalizes on the proven success and reasoning capabilities of LLMs, employing vision and audio modalities as perceptual descriptors for language.

9.3 Conclusion

In this thesis, innovative methods are developed to harness language as a supervisory signal for enhancing video understanding. The bottom-up learning of a video's constituent modalities is proven to be an effective and cost-effective strategy for tackling longer-form and multimodal video tasks – particularly in scenarios where supervised data is scarce, and compute is costly. Further investigations reveal that video learning from complementary language sources, such as narrations in movies, serve as a valuable resource. As a consequence, automated video systems can be readily applied to high-impact applications such as audio descriptions, and animal wildlife analysis. Beyond these immediate applications, the advancements in language-based video systems propel us closer to achieving human-level intelligence.

References

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan (2016). "Youtube-8m: A large-scale video classification benchmark". In: *arXiv preprint arXiv:1609.08675*.

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*.

Sandhini Agarwal, Gretchen Krueger, Jack Clark, Alec Radford, Jong Wook Kim, and Miles Brundage (2021). "Evaluating CLIP: towards characterization of broader capabilities and downstream implications". In: *arXiv preprint arXiv:2108.02818*.

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien (2016). "Unsupervised learning from narrated instruction videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

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

Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman (2020). "Self-supervised multimodal versatile networks". In: *NeurIPS*.

Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran (2020). "Self-supervised learning by cross-modal audio-video clustering". In: *Advances in Neural Information Processing Systems*.

Elad Amrani, Rami Ben Ari, Daniel Rotman, and Alex Bronstein (2020). “Noise estimation using density estimation for self-supervised multimodal learning”. In: *arXiv preprint arXiv:2003.03186*.

David J Anderson and Pietro Perona (2014). “Toward a science of computational ethology”. In: *Neuron*.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould (2016). “Spice: Semantic propositional image caption evaluation”. In: *Proc. ECCV*. Springer.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang (2018). “Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering”. In: *CVPR*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell (2017). “Localizing moments in video with natural language”. In: *ICCV*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh (2015). “Vqa: Visual question answering”. In: *ICCV*.

Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic (2016). “NetVLAD: CNN architecture for weakly supervised place recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

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

Adam Clark Arcadi, Daniel Robert, and Christophe Boesch (1998). “Buttress drumming by wild chimpanzees: Temporal patterning, phrase integration into loud calls, and preliminary evidence for individual distinctiveness”. In: *Primates*.

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi (2019). “Self-labelling via simultaneous clustering and representation learning”. In: *arXiv preprint arXiv:1911.05371*.

Yuki Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi (2020). “Labelling unlabelled videos from scratch with multi-modal self-supervision”. In: *Advances in Neural Information Processing Systems*.

Magdalena Babiszewska, Anne Marijke Schel, Claudia Wilke, and Katie E Slocombe (2015). “Social, contextual, and individual factors affecting the occurrence and acoustic structure of drumming bouts in wild chimpanzees (Pan troglodytes)”. In: *American journal of physical anthropology*.

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

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola (2022). "Exploring Visual Prompts for Adapting Large-Scale Models". In: *arXiv:2203.17274*.

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

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

Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman (2020b). "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: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman (2021). "Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval". In: *Proc. ICCV*.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman (2022). "A CLIP-Hitchhiker's Guide to Long Video Retrieval". In: *arXiv preprint arXiv:2205.08508*.

Kobus Barnard, Pinar Duygulu, David Forsyth, Nando De Freitas, David M Blei, and Michael I Jordan (2003). "Matching words and pictures". In: *The Journal of Machine Learning Research*.

Kobus Barnard and David Forsyth (2001). "Learning the semantics of words and pictures". In: *Proc. ICCV*. IEEE.

Hugo Berg, Siobhan Hall, Yash Bhalgat, Hannah Kirk, Aleksandar Shtedritski, and Max Bain (2022). "A Prompt Array Keeps the Bias Away: Debiasing Vision-Language Models with Adversarial Learning". In: *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*.

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

Joana Bessa, Kimberley Hockings, and Dora Biro (2021). "First evidence of chimpanzee extractive tool use in Cantanhez, Guinea-Bissau: Cross-community variation in honey dipping". In: *Frontiers in Ecology and Evolution*.

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

Dora Biro, Noriko Inoue-Nakamura, Rikako Tonooka, Gen Yamakoshi, Claudia Sousa, and Tetsuro Matsuzawa (2003). "Cultural innovation and transmission of tool use in wild chimpanzees: evidence from field experiments". In: *Animal cognition*.

Christophe Boesch (1991). "Symbolic communication in wild chimpanzees?" In: *Human Evolution*.

Simion-Vlad Bogolin, Ioana Croitoru, Hailin Jin, Yang Liu, and Samuel Albanie (2022). *Cross modal retrieval with querybank normalisation*.

Piotr Bojanowski, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic (2013). "Finding actors and actions in movies". In: *Proc. ICCV*.

Piotr Bojanowski, Remi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid (Dec. 2015). "Weakly-Supervised Alignment of Video With Text". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Digbalay Bose, Rajat Hebbar, Krishna Somandepalli, Haoyang Zhang, Yin Cui, Kree Cole-McLaughlin, Huisheng Wang, and Shrikanth Narayanan (2023). "MovieCLIP: Visual Scene Recognition in Movies". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

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

Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill (2020a). "Pyannote. audio: neural building blocks for speaker diarization". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill (2020b). "pyannote.audio: neural building blocks for speaker diarization". In: *Proc. ICASSP*.

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, Ernesto Coto, and Andrew Zisserman (2021b). "Automated video labelling: Identifying faces by corroborative evidence". In: *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE.

Andrew Brown, Vicky Kalogeiton, and Andrew Zisserman (2021c). "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, et al. (2020). "Language models are few-shot learners". In: *NeurIPS*.

Fabio Brugnara, Daniele Falavigna, and Maurizio Omologo (1993). "Automatic segmentation and labeling of speech based on Hidden Markov Models". In: *Speech Communication*.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles (2015). "ActivityNet: A large-scale video benchmark for human activity understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ben Caldwell, Michael Cooper, Loretta Guarino Reid, Gregg Vanderheiden, Wendy Chisholm, John Slatin, and Jason White (2008). "Web content accessibility guidelines (WCAG) 2.0". In: *WWW Consortium (W3C)*.

James Cameron (1997). *Titanic*. Paramount Pictures.

Mauricio Cantor, Adriana A Maldonado-Chaparro, Kristina B Beck, Hanja B Brandl, Gerald G Carter, Peng He, Friederike Hillemann, James A Klarevas-Irby, Mina Ogino, Danai Papageorgiou, et al. (2021). "The importance of individual-to-society feedbacks in animal ecology and evolution". In: *Journal of Animal Ecology*.

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.*

Anthony Caravaggi, Peter B Banks, A Cole Burton, Caroline MV Finlay, Peter M Haswell, Matt W Hayward, Marcus J Rowcliffe, and Mike D Wood (2017).

"A review of camera trapping for conservation behaviour research". In: *Remote Sensing in Ecology and Conservation*.

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

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. (2006). "The AMI meeting corpus: A pre-announcement". In: *Machine Learning for Multimodal Interaction: Second International Workshop, MLMI 2005, Edinburgh, UK, July 11-13, 2005, Revised Selected Papers 2*. Springer.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin (2021). "Emerging properties in self-supervised vision transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

João Carreira and Andrew Zisserman (2017). "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset". In: *CVPR*.

Susana Carvalho, Eugénia Cunha, Cláudia Sousa, and Tetsuro Matsuzawa (2008). "Chaınes opératoires and resource-exploitation strategies in chimpanzee (Pan troglodytes) nut cracking". In: *Journal of Human Evolution*.

Santiago Castro and Fabian Caba Heilbron (2022). "FitCLIP: Refining Large-Scale Pretrained Image-Text Models for Zero-Shot Video Understanding Tasks". In: *arXiv preprint arXiv:2203.13371*.

Aman Chadha, Gurneet Arora, and Navpreet Kaloty (2021). "iPerceive: Applying Common-Sense Reasoning to Multi-Modal Dense Video Captioning and Video Question Answering". In: *Proc. WACV*.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut (2021). "Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts". In: *CVPR*.

David Chen and William B Dolan (2011). "Collecting highly parallel data for paraphrase evaluation". In:

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

Hui Chen, Guiguang Ding, Xudong Liu, Zijia Lin, Ji Liu, and Jungong Han (2020). *IMRAM: Iterative Matching with Recurrent Attention Memory for Cross-Modal Image-Text Retrieval*. arXiv: 2003.03772 [cs.CV].

Hui Chen, Hanyi Zhang, Longbiao Wang, Kong Aik Lee, Meng Liu, and Jianwu Dang (2023). "Self-Supervised Audio-Visual Speaker Representation with Co-Meta Learning". In: *ICASSP*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. (2021). "Evaluating large language models trained on code". In: *arXiv preprint arXiv:2107.03374*.

Peng Chen, Pranjal Swarup, Wojciech Michal Matkowski, Adams Wai Kin Kong, Su Han, Zhihe Zhang, and Hou Rong (2020). "A study on giant panda recognition based on images of a large proportion of captive pandas". In: *Ecology and Evolution*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. (2022). "Wavlm: Large-scale self-supervised pre-training for full stack speech processing". In: *IEEE Journal of Selected Topics in Signal Processing*.

Shaoxiang Chen and Yu-Gang Jiang (2021). "Towards bridging event captioner and sentence localizer for weakly supervised dense event captioning". In: *Proc. CVPR*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton (2020). "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick (2015). "Microsoft coco captions: Data collection and evaluation server". In: *arXiv preprint arXiv:1504.00325*.

Xinlei Chen and C. Lawrence Zitnick (2014). "Learning a Recurrent Visual Representation for Image Caption Generation". In: *arXiv*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu (2020). "Uniter: Universal image-text representation learning". In: *European conference on computer vision*. Springer.

Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng (2018). "A2-Nets: Double Attention Networks". In: *arXiv preprint arXiv:1810.11579*.

Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen (2021). *Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever (2019). "Generating long sequences with sparse transformers". In: *arXiv preprint arXiv:1904.10509*.

Chung-Cheng Chiu, Wei Han, Yu Zhang, Ruoming Pang, Sergey Kishchenko, Patrick Nguyen, Arun Narayanan, Hank Liao, Shuyuan Zhang, Anjuli Kannan, et al. (2019). "A comparison of end-to-end models for long-form speech recognition". In: *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*. IEEE.

Fredrik Christiansen, Marianne H Rasmussen, and David Lusseau (2013). "Inferring activity budgets in wild animals to estimate the consequences of disturbances". In: *Behavioral Ecology*.

Condensed Movies Challenge (n.d.). https://www.robots.ox.ac.uk/~vgg/research/condensed-movies/challenge.html. Accessed: 2022-03-06.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli (2020). "Unsupervised cross-lingual representation learning for speech recognition". In: *arXiv preprint arXiv:2006.13979*.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi (2019). "On the relationship between self-attention and convolutional layers". In: *arXiv preprint arXiv:1911.03584*.

Timothee Cour, Benjamin Sapp, Chris Jordan, and Ben Taskar (2009). "Learning from ambiguously labeled images". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Ioana Croitoru, Simion-Vlad Bogolin, Marius Leordeanu, Hailin Jin, Andrew Zisserman, Samuel Albanie, and Yang Liu (2021). "TeachText: CrossModal Generalized Distillation for Text-Video Retrieval". In: *Proc. ICCV*. IEEE.

James E Cutting (2016). "Narrative theory and the dynamics of popular movies". In: *Psychonomic bulletin & review*.

Debayan Deb, Susan Wiper, Sixue Gong, Yichun Shi, Cori Tymoszek, Alison Fletcher, and Anil K Jain (2018). "Face recognition: Primates in the wild". In: *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE.

Chaorui Deng, Shizhe Chen, Da Chen, Yuan He, and Qi Wu (2021). "Sketch, ground, and refine: Top-down dense video captioning". In: *CVPR*.

Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou (2020). "RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild". 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 2020*.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *NAACL-HLT*.

Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu (2021). *Similarity Reasoning and Filtration for Image-Text Matching*. arXiv: 2101.01368 [cs.CV].

Pierre Dognin, Igor Melnyk, Youssef Mroueh, Inkit Padhi, Mattia Rigotti, Jarret Ross, Yair Schiff, Richard A Young, and Brian Belgodere (2020). "Image captioning as an assistive technology: Lessons learned from vizwiz 2020 challenge". In: *arXiv preprint arXiv:2012.11696*.

Davide M Dominoni, Wouter Halfwerk, Emily Baird, Rachel T Buxton, Esteban Fernández-Juricic, Kurt M Fristrup, Megan F McKenna, Daniel J Mennitt, Elizabeth K Perkin, Brett M Seymoure, et al. (2020). "Why conservation biology can benefit from sensory ecology". In: *Nature Ecology & Evolution*.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell (2015). "Long-term recurrent convolutional networks for visual recognition and description". In: *Proc. CVPR*.

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: *ICLR*.

Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach, and Jean Ponce (2009). "Automatic annotation of human actions in video". In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE.

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*.

Philippe Ercolessi, Hervé Bredin, and Christine Sénac (2012). "StoViz: story visualization of TV series". In: *Proceedings of the 20th ACM international conference on Multimedia*.

Sepideh Esmaeilpour, Bing Liu, Eric Robertson, and Lei Shu (2021). "Zero-Shot Open Set Detection by Extending CLIP". In: *arXiv preprint arXiv:2109.02748*.

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, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman (2010). "The pascal visual object classes (voc) challenge". In: *International journal of computer vision*.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler (2017). "Vse++: Improving visual-semantic embeddings with hard negatives". In: *arXiv preprint arXiv:1707.05612*.

Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen (2021). "CLIP2Video: Mastering Video-Text Retrieval via Image CLIP". In: *arXiv preprint arXiv:2106.11097*.

Anna Favaro, Laureano Moro-Velázquez, Ankur Butala, Chelsie Motley, Tianyu Cao, Robert David Stevens, Jesús Villalba, and Najim Dehak (2023). "Multilingual evaluation of interpretable biomarkers to represent language and speech patterns in Parkinson's disease". In: *Frontiers in Neurology*.

L. Feng, X. Zhao, Y. Liu, Y. Yao, and B. Jin (2010). "A similarity measure of Jumping Dynamic Time Warping". In: *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*.

David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik (2018). "From lifestyle vlogs to everyday interactions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu (2021). "VIOLET: End-to-End Video-Language Transformers with Masked Visual-token Modeling". In: *arXiv preprint arXiv:2111.12681*.

Tsu-Jui Fu, Licheng Yu, Ning Zhang, Cheng-Yang Fu, Jong-Chyi Su, William Yang Wang, and Sean Bell (2023). "Tell me what happened: Unifying text-guided video completion via multimodal masked video generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Soichiro Fujita, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata (2020). "SODA: Story oriented dense video captioning evaluation framework". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer.

Valentin Gabeur, Arsha Nagrani, Chen Sun, Karteek Alahari, and Cordelia Schmid (Jan. 2022). "Masking Modalities for Cross-Modal Video Retrieval". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid (2020). "Multi-modal transformer for video retrieval". In: *ECCV*.

Adrien Gaidon, Zaïd Harchaoui, and Cordelia Schmid (2013). "Temporal Localization of Actions with Actoms". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao (2021). "Clip-adapter: Better vision-language models with feature adapters". In: *arXiv:2110.04544*.

Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani (2020). "Listen to look: Action recognition by previewing audio". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo (2022). "BridgeFormer: Bridging Video-text Retrieval with Multiple Choice Questions". In: *arXiv preprint arXiv:2201.04850*.

Gregory Gelly and Jean-Luc Gauvain (2018). "Optimization of RNN-Based Speech Activity Detection". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

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: *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE.

Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell (2017). "Actionvlad: Learning spatio-temporal aggregation for action classification". In: *CVPR*.

John Godfrey and Edward Holliman (1993). "Switchboard-1 Release 2 LDC97S62". In: *Linguistic Data Consortium*.

Lucıa Gómez-Zaragozá, Simone Wills, Cristian Tejedor-Garcia, Javier Marın-Morales, Mariano Alcañiz, and Helmer Strik (2023). "Alzheimer Disease Classification through ASR-based Transcriptions: Exploring the Impact of Punctuation and Pauses". In: *arXiv preprint arXiv:2306.03443*.

Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass (2022). "Ssast: Self-supervised audio spectrogram transformer". In: *Proc. AAAI*.

Walter Goodwin, Sagar Vaze, Ioannis Havoutis, and Ingmar Posner (2022). "Semantically Grounded Object Matching for Robust Robotic Scene Rearrangement". In: *ICRA*.

Kyle Gorman, Jonathan Howell, and Michael Wagner (2011). “Prosodylab-aligner: A tool for forced alignment of laboratory speech”. In: *Canadian Acoustics*.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. (2020). “Bootstrap your own latent-a new approach to self-supervised learning”. In: *Advances in neural information processing systems*.

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”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui (2021). “Open-Vocabulary Detection via Vision and Language Knowledge Distillation”. In: *arXiv preprint arXiv:2104.13921*.

Tanmay Gupta and Aniruddha Kembhavi (2023). “Visual programming: Compositional visual reasoning without training”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya (2020). “Captioning images taken by people who are blind”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*. Springer.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith (2020). “Don’t stop pretraining: adapt language models to domains and tasks”. In: *arXiv preprint arXiv:2004.10964*.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman (2023). “AutoAD: Movie Description in Context”. In: *Proc. CVPR*.

Tengda Han, Weidi Xie, and Andrew Zisserman (2019). “Video Representation Learning by Dense Predictive Coding”. In: *Workshop on Large Scale Holistic Video Understanding, ICCV*.

Tengda Han, Weidi Xie, and Andrew Zisserman (2022). “Temporal Alignment Networks for Long-term Video”. In: *CVPR*.

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh (2018). “Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?” In: *CVPR*.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick (2022). "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Esteve (2018). "TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation". In: *Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20*. Springer.

Geoffrey E Hinton and Ruslan R Salakhutdinov (2006). "Reducing the dimensionality of data with neural networks". In: *science*.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang (2022). "Cogvideo: Large-scale pretraining for text-to-video generation via transformers". In: *arXiv preprint arXiv:2205.15868*.

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei (2018). "Relation Networks for Object Detection". In: *CVPR*.

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu (2019). "Squeeze-and-Excitation Networks". In: *IEEE PAMI*.

Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut (2020). "Multimodal pretraining for dense video captioning". In: *arXiv preprint arXiv:2011.11760*.

Kang Huang, Yaning Han, Ke Chen, Hongli Pan, Gaoyang Zhao, Wenling Yi, Xiaoxi Li, Siyuan Liu, Pengfei Wei, and Liping Wang (2021). "A hierarchical 3D-motion learning framework for animal spontaneous behavior mapping". In: *Nature communications*.

Qingqiu Huang et al. (2020a). "MovieNet: A Holistic Dataset for Movie Understanding". In: *ECCV*.

Qingqiu Huang, Wentao Liu, and Dahua Lin (2018). "Person Search in Videos with One Portrait Through Visual and Temporal Links". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin (2020b). "MovieNet: A Holistic Dataset for Movie Understanding". In: *The European Conference on Computer Vision (ECCV)*.

Qingqiu Huang, Lei Yang, Huaiyi Huang, Tong Wu, and Dahua Lin (2020c). "Caption-Supervised Face Recognition: Training a State-of-the-Art Face Model without Manual Annotation". In: *The European Conference on Computer Vision (ECCV)*.

Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. (2016). "Visual storytelling". In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*.

Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer (2014). "Densenet: Implementing efficient convnet descriptor pyramids". In: *arXiv preprint arXiv:1404.1869*.

Vladimir Iashin and Esa Rahtu (2020a). "A better use of audio-visual cues: Dense video captioning with bi-modal transformer". In: *BMVC*.

Vladimir Iashin and Esa Rahtu (2020b). "Multi-modal dense video captioning". In: *CVPR Workshops*.

Oana Ignat, Laura Burdick, Jia Deng, and Rada Mihalcea (2019). "Identifying Visible Actions in Lifestyle Vlogs". In: *arXiv preprint arXiv:1906.04236*.

Anil K Jain and Richard C Dubes (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. (2020). "Computer vision for autonomous vehicles: Problems, datasets and state of the art". In: *Foundations and Trends® in Computer Graphics and Vision*.

Bhavan Jasani, Rohit Girdhar, and Deva Ramanan (2019). "Are we asking the right questions in MovieQA?" In: *ICCVW*.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu (2012). "3D convolutional neural networks for human action recognition". In: *IEEE transactions on pattern analysis and machine intelligence*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig (2021). "Scaling up visual and vision-language representation learning with noisy text supervision". In: *International Conference on Machine Learning*. PMLR.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim (2022). "Visual Prompt Tuning". In: *Proc. ECCV*.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu (2019). "Leveraging weakly supervised data to improve end-to-end speech-to-text translation". In: *ICASSP*. IEEE.

Peng Jin, Hao Li, Zesen Cheng, Jinfa Huang, Zhennan Wang, Li Yuan, Chang Liu, and Jie Chen (2023). “Text-Video Retrieval with Disentangled Conceptualization and Set-to-Set Alignment”. In: *arXiv preprint arXiv:2305.12218*.

Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie (2022). “Prompting Visual-Language Models for Efficient Video Understanding”. In: *Proc. ECCV*.

Kumara Kahatapitiya, Anurag Arnab, Arsha Nagrani, and Michael S Ryoo (2023). “VicTR: Video-conditioned Text Representations for Activity Recognition”. In: *arXiv preprint arXiv:2304.02560*.

Jingu Kang, Jaesung Huh, Hee Soo Heo, and Joon Son Chung (2022). “Augmentation adversarial training for self-supervised speaker representation learning”. In: *IEEE Journal of Selected Topics in Signal Processing*.

Andrej Karpathy and Li Fei-Fei (2015). “Deep visual-semantic alignments for generating image descriptions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Stephan P Kaufhold and Edwin JC Van Leeuwen (2019). “Why intergroup variation matters for understanding behaviour”. In: *Biology Letters*.

W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman (2017). “The Kinetics Human Action Video Dataset”. In: *CoRR*.

Yeon-Jun Kim and Alistair Conkie (2002). “Automatic segmentation combining an HMM-based approach and spectral boundary correction”. In: *Seventh International conference on spoken language processing*.

Jason Kincaid (2018). *Which Automatic Transcription Service is the Most Accurate?* https://medium.com/descript/which-automatic-transcription-service-is-the-most-accurate-2018-2e859b23ed19. Accessed: 2023-04-27.

Diederik P. Kingma and Jimmy Ba (2014). “Adam: A Method for Stochastic Optimization”. In: *CoRR*.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel (2014). “Unifying visual-semantic embeddings with multimodal neural language models”. In: *arXiv preprint arXiv:1411.2539*.

Bruno Korbar, Fabio Petroni, Rohit Girdhar, and Lorenzo Torresani (2020). “Video Understanding as Machine Translation”. In: *arXiv preprint arXiv:2006.07203*.

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, Du Tran, and Lorenzo Torresani (Oct. 2019). "SCSampler: Sampling Salient Clips From Video for Efficient Action Recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Bruno Korbar and Andrew Zisserman (2022). "Personalised CLIP or: how to find your vacation videos". In: *British Machine Vision Conference*.

Jens Krause, Stefan Krause, Robert Arlinghaus, Ioannis Psorakis, Stephen Roberts, and Christian Rutz (2013). "Reality mining of animal social systems". In: *Trends in ecology & evolution*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles (2017a). "Dense-Captioning Events in Videos". In: *Proc. ICCV*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. (2017b). "Visual genome: Connecting language and vision using crowdsourced dense image annotations". In: *International Journal of Computer Vision*.

Hjalmar S Kühl, Christophe Boesch, Lars Kulik, Fabian Haas, Mimi Arandjelovic, Paula Dieguez, Gaëlle Bocksberger, Mary Brooke McElreath, Anthony Agbor, Samuel Angedakin, et al. (2019). "Human impact erodes chimpanzee behavioral diversity". In: *Science*.

Anna Kukleva, Makarand Tapaswi, and Ivan Laptev (June 2020a). "Learning Interactions and Relationships Between Movie Characters". In: *Proc. CVPR*.

Anna Kukleva, Makarand Tapaswi, and Ivan Laptev (2020b). "Learning interactions and relationships between movie characters". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll (2020). "CTC-segmentation of large corpora for german end-to-end speech recognition". In: *Speech and Computer (SPECOM 2020)*. Springer.

Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld (2008). "Learning realistic human actions from movies". In: *Proc. CVPR*.

Victor Lavrenko, Raghavan Manmatha, and Jiwoon Jeon (2003). "A model for learning the semantics of pictures". In: *NeurIPS*.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He (2018). *Stacked Cross Attention for Image-Text Matching*. arXiv: 1803.08024 [cs.CV].

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu (2021). "Less is more: Clipbert for video-and-language learning via sparse sampling". In: *arXiv preprint arXiv:2102.06183*.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg (2018). "Tvqa: Localized, compositional video question answering". In: *arXiv preprint arXiv:1809.01696*.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal (2019). "TVQA+: Spatio-Temporal Grounding for Video Question Answering". In: *Tech Report, arXiv*.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal (2020). "Tvr: A large-scale dataset for video-subtitle moment retrieval". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*. Springer.

Brian Lester, Rami Al-Rfou, and Noah Constant (2021). "The Power of Scale for Parameter-Efficient Prompt Tuning". In: *EMNLP*.

Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang (2019). "DSFD: dual shot face detector". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Jingbei Li, Yi Meng, Zhiyong Wu, Helen Meng, Qiao Tian, Yuping Wang, and Yuxuan Wang (2022). "Neufa: Neural network based end-to-end forced alignment with bidirectional attention mechanism". In: *ICASSP*. IEEE.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi (2022). "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation". In: *ICML*.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi (2021). "Align before Fuse: Vision and Language Representation Learning with Momentum Distillation". In: *NeurIPS*.

Junnan Li, Yongkang Wong, Qi Zhao, and M. Kankanhalli (2020). "Video Storytelling: Textual Summaries for Events". In: *IEEE Transactions on Multimedia*.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu (2020). "Hero: Hierarchical encoder for video+ language omni-representation pre-training". In: *EMNLP*.

Xiang Lisa Li and Percy Liang (2021). "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *ACL*.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. (2020). "Oscar: Object-semantics aligned pre-training for vision-language tasks". In: *ECCV*.

Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei (2018). "Jointly localizing and describing events for dense video captioning". In: *CVPR*.

Chin-Yew Lin (2004). “Rouge: A package for automatic evaluation of summaries”. In: *Text summarization branches out*.

Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang (2022). “SwinBERT: End-to-end transformers with sparse attention for video captioning”. In: *CVPR*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014). “Microsoft COCO: Common objects in context”. In: *ECCV*.

Yan-Bo Lin, Jie Lei, Mohit Bansal, and Gedas Bertasius (2022). “ECLIPSE: Efficient Long-range Video Retrieval using Sight and Sound”. In: *arXiv preprint arXiv:2204.02874*.

Jialin Liu, Sam Snodgrass, Ahmed Khalifa, Sebastian Risi, Georgios N Yannakakis, and Julian Togelius (2021). “Deep learning for procedural content generation”. In: *Neural Computing and Applications*.

Song Liu, Haoqi Fan, Shengsheng Qian, Yiru Chen, Wenkui Ding, and Zhongyuan Wang (2021). *HiT: Hierarchical Transformer with Momentum Contrast for Video-Text Retrieval*. arXiv: 2103.15049 [cs.CV].

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg (2016). “Ssd: Single shot multibox detector”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer.

Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman (2019). “Use What You Have: Video Retrieval Using Representations From Collaborative Experts”. In: *Proc. BMVC*.

Ilya Loshchilov and Frank Hutter (2017). “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101*.

Jérôme Louradour (2023). *whisper-timestamped*. https://github.com/linto-ai/whisper-timestamped/tree/f861b2b19d158f3cbf4ce524f22c78cb471d6131.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh (2018). “Neural Baby Talk”. In: *CVPR*.

Chenxu Luo and Alan Yuille (2019). “Grouped Spatial-Temporal Aggretation for Efficient Action Recognition”. In: *ICCV*.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou (2020). “UniVL: A unified video and language pre-training model

for multimodal understanding and generation". In: *arXiv preprint arXiv:2002.06353*.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li (2021). "CLIP4Clip: An Empirical Study of CLIP for End to End Video Clip Retrieval". In: *arXiv preprint arXiv:2104.08860*.

Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan (2023). "VideoFusion: Decomposed Diffusion Models for High-Quality Video Generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Richard Marquand (1983). *Star Wars: Episode VI – Return of the Jedi*. 20th Century Fox.

Marcin Marszałek, Ivan Laptev, and Cordelia Schmid (2009). "Actions in Context". In: *Proc. CVPR*.

Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge (2018). "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning". In: *Nature neuroscience*.

Tetsuro Matsuzawa (1994). "Field experiments on use of stone tools by chimpanzees in the wild." In:

Tetsuro Matsuzawa, Tatyana Humle, and Yukimaru Sugiyama (2011). *The chimpanzees of Bossou and Nimba*. Springer.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger (2017). "Montreal forced aligner: Trainable text-speech alignment using kaldi." In: *Interspeech*.

Philip M McCarthy and Scott Jarvis (2010). "MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment". In: *Behavior research methods*.

Philip Mccarthy and Scott Jarvis (May 2010). "MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment". In: *Behavior research methods*.

WC McGrew, CEG Tutin, RW Wrangham, et al. (2001). "Charting cultural variation in chimpanzees". In: *Behaviour*.

Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman (2021). "Thinking Fast and Slow: Efficient Text-to-Visual Retrieval with Transformers". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman (2020). "End-to-End Learning of Visual Representations from Uncurated Instructional Videos". In: *CVPR*.

Antoine Miech, Ivan Laptev, and Josef Sivic (2017). "Learnable pooling with Context Gating for video classification". In: *arXiv:1706.06905*.

Antoine Miech, Ivan Laptev, and Josef Sivic (2018). "Learning a text-video embedding from incomplete and heterogeneous data". In: *arXiv*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic (2019). "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury (2018). "Learning joint embedding with multimodal cues for cross-modal video-text retrieval". In: *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*.

Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury (2019). "Weakly supervised video moment retrieval from text queries". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ron Mokady, Amir Hertz, and Amit H Bermano (2021). "ClipCap: CLIP prefix for image captioning". In: *arXiv preprint arXiv:2111.09734*.

Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. (2019). "Moments in time dataset: one million videos for event understanding". In: *IEEE transactions on pattern analysis and machine intelligence*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen (2016). "A corpus and cloze evaluation for deeper understanding of commonsense stories". In:

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie (2021). "SLIP: Self-supervision meets Language-Image Pre-training". In: *arXiv preprint arXiv:2112.12750*.

Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han (2019). "Streamlined dense video captioning". In: *Proc. CVPR*.

David Nadeau and Satoshi Sekine (2007). "A survey of named entity recognition and classification". In: *Lingvisticae Investigationes*.

Arsha Nagrani (2020). “Video understanding using multimodal deep learning”. PhD thesis. University of Oxford.

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: *Proc. Interspeech 2017*.

Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, and Cordelia Schmid (2022). “Learning Audio-Video Modalities from Image Captions”. In: *Proc. ECCV*.

Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman (June 2020). “Speech2Action: Cross-Modal Supervision for Action Recognition”. In: *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Arsha Nagrani and Andrew Zisserman (2017). “From Benedict Cumberbatch to Sherlock Holmes: Character Identification in TV series without a Script”. In: *Proc. BMVC*.

Iftekhar Naim, Abdullah Al Mamun, Young Chol Song, Jiebo Luo, Henry Kautz, and Daniel Gildea (2016). “Aligning movies with scripts by exploiting temporal ordering constraints”. In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE.

Toshisada Nishida (2011). *Chimpanzees of the lakeshore: natural history and culture at Mahale*. Cambridge University Press.

Mohammad Sadegh Norouzzadeh, Dan Morris, Sara Beery, Neel Joshi, Nebojsa Jojic, and Jeff Clune (2021). “A deep active learning system for species identification and counting in camera trap images”. In: *Methods in ecology and evolution*.

Mohammad Sadegh Norouzzadeh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S Palmer, Craig Packer, and Jeff Clune (2018). “Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning”. In: *Proceedings of the National Academy of Sciences*.

David Nukrai, Ron Mokady, and Amir Globerson (2022). “Text-Only Training for Image Captioning using Noise-Injected CLIP”. In: *arXiv preprint arXiv:2211.00575*.

Andreea-Maria Oncescu, Joao F Henriques, Yang Liu, Andrew Zisserman, and Samuel Albanie (2021). “Queryd: A video dataset with high-quality text and audio narrations”. In: *Proc. ICASSP*. IEEE.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa,

Alaaeldin El-Nouby, et al. (2023). “Dinov2: Learning robust visual features without supervision”. In: *arXiv preprint arXiv:2304.07193*.

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)*. IEEE.

Pinelopi Papalampidi, Frank Keller, and Mirella Lapata (2019). “Movie plot analysis via turning point identification”. In: *arXiv preprint arXiv:1908.10328*.

Pinelopi Papalampidi, Frank Keller, and Mirella Lapata (2021). “Movie summarization via sparse graph construction”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Jae Sung Park, Marcus Rohrbach, Trevor Darrell, and Anna Rohrbach (2019). “Adversarial Inference for Multi-Sentence Video Description”. In: *Proc. CVPR*.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran (2018). “Image transformer”. In: *ICML*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer (2017). “Automatic differentiation in pytorch”. In:

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *NeurIPS*.

Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, João Henriques, and Andrea Vedaldi (2020). “Support-set bottlenecks for video-text representation learning”. In: *arXiv preprint arXiv:2010.02824*.

Alonso Patron-Perez, M. Marszałek, Andrew Zisserman, and Ian D. Reid (2010). “High Five: Recognising Human Interactions in TV Shows”. In: *Proc. BMVC*.

Georgios Pavlakos, Ethan Weber, Matthew Tancik, and Angjoo Kanazawa (2022). “The One Where They Reconstructed 3D Humans and Environments in TV Shows”. In: *Proc. ECCV*. Ed. by Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner.

Elisa Perego (2016). “Gains and losses of watching audio described films for sighted viewers”. In: *Target*.

Hamed Pirsiavash and Deva Ramanan (2014). "Parsing videos of actions with segmental grammars". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Jean Baptiste Polle (n.d.). *Camembert-NER: model fine-tuned from camemBERT for NER task*. https://huggingface.co/Jean-Baptiste/camembert-ner. Accessed: 2022-11-01.

KR Prajwal, Liliane Momeni, Triantafyllos Afouras, and Andrew Zisserman (2021). "Visual keyword spotting with attention". In: *arXiv preprint arXiv:2110.15957*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever (2021). "Learning Transferable Visual Models From Natural Language Supervision". In: *Proc. ICML*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever (2022a). "Robust Speech Recognition via Large-Scale Weak Supervision". In: *OpenAI blog*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever (2022b). "Robust speech recognition via large-scale weak supervision". In: *arXiv preprint arXiv:2212.04356*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). "Language Models are Unsupervised Multitask Learners". In: *OpenAI blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *The Journal of Machine Learning Research*.

Tanzila Rahman, Bicheng Xu, and Leonid Sigal (2019). "Watch, listen and tell: Multi-modal weakly supervised dense event captioning". In: *Proc. ICCV*.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens (2019). "Stand-alone self-attention in vision models". In: *arXiv preprint arXiv:1906.05909*.

Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin (2020). "A Local-to-Global Approach to Multi-modal Movie Scene Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Hareesh Ravi, Kushal Kafle, Scott Cohen, Jonathan Brandt, and Mubbasir Kapadia (2021). "AESOP: Abstract Encoding of Stories, Objects, and Pictures". In: *Proc. ICCV*.

Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal (2013). "Grounding action descriptions in videos". In: *Transactions of the Association for Computational Linguistics*.

Video Description Research and Development Center (2013). *YouDescribe*. URL: https://youdescribe.org/.

Vernon Reynolds (2005). *The chimpanzees of the Budongo forest: Ecology, behaviour and conservation*. OUP Oxford.

Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele (2015a). "A Dataset for Movie Description". In: *CVPR*.

Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele (2015b). "A dataset for movie description". In: *Proc. CVPR*.

Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Chris Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele (2017a). "Movie Description". In: *International Journal of Computer Vision*.

Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele (2017b). "Movie description". In: *International Journal of Computer Vision*.

Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele (2012). "A database for fine grained activity detection of cooking activities". In: *CVPR*.

Andrew Rouditchenko, Angie Boggust, David Harwath, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, et al. (2020). "AVLnet: Learning audio-visual language representations from instructional videos". In: *arXiv preprint arXiv:2006.09199*.

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*.

Faizaan Sakib and Tilo Burghardt (2021). "Visual Recognition of Great Ape Behaviours in the Wild". In: *Proc. ICPR Workshop on VAIB*.

H. Sakoe and Seibi Chiba (1978). "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *CoRR*. arXiv: 1910.01108. URL: http://arxiv.org/abs/1910.01108.

Daniel Schofield, Arsha Nagrani, Andrew Zisserman, Misato Hayashi, Tetsuro Matsuzawa, Dora Biro, and Susana Carvalho (2019). "Chimpanzee face recognition from videos in the wild using deep learning". In: *Science advances*.

Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena (2015). "Unsupervised semantic parsing of video collections". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid (2022). "End-to-end generative pretraining for multimodal video captioning". In: *Proc. CVPR*.

Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid (2021). "Look Before you Speak: Visually Contextualized Utterances". In: *CVPR*.

Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani (2021). "Only time can tell: Discovering temporal data for temporal modeling". In: *WACV*.

Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin (2020). "Finegym: A hierarchical video dataset for fine-grained action understanding". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut (2018a). "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut (2018b). "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning". In:

Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, and Xiangyang Xue (2017). "Weakly supervised dense video captioning". In: *Proc. CVPR*.

Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou (2019). "Dense procedure captioning in narrated instructional videos". In:

Nina Shvetsova, Brian Chen, Andrew Rouditchenko, Samuel Thomas, Brian Kingsbury, Rogerio Feris, David Harwath, James Glass, and Hilde Kuehne (2021). "Everything at Once–Multi-modal Fusion Transformer for Video Retrieval". In: *arXiv preprint arXiv:2112.04446*.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta (2016a). "Hollywood in homes: Crowdsourcing data collection for activity understanding". In: *Proc. ECCV*. Springer.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta (2016b). "Hollywood in homes: Crowdsourcing data collection for activity understanding". In: *Proc. ECCV*. Springer.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. (2022). "Make-a-video: Text-to-video generation without text-video data". In: *arXiv preprint arXiv:2209.14792*.

Josef Sivic, Mark Everingham, and Andrew Zisserman (2009). ""Who are you?" – Learning Person Specific Classifiers from Video". In: *Proc. CVPR*.

Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng (2014). "Grounded compositional semantics for finding and describing images with sentences". In: *Transactions of the Association for Computational Linguistics*.

Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem (2022). "Mad: A scalable dataset for language grounding in videos from movie audio descriptions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Andreas Stolcke, Neville Ryant, Vikramjit Mitra, Jiahong Yuan, Wen Wang, and Mark Liberman (2014). "Highly accurate phonetic segmentation using boundary correction models and system fusion". In: *ICASSP*. IEEE.

Oliver Sturman, Lukas von Ziegler, Christa Schläppi, Furkan Akyol, Mattia Privitera, Daria Slominski, Christina Grimm, Laetitia Thieren, Valerio Zerbi, Benjamin Grewe, et al. (2020). "Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions". In: *Neuropsychopharmacology*.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid (2019). "Videobert: A joint model for video and language representation learning". In: *arXiv preprint arXiv:1904.01766*.

Yidan Sun, Qin Chao, and Boyang Li (2022). "Synopses of Movie Narratives: a Video-Language Dataset for Story Understanding". In: *arXiv preprint arXiv:2203.05711*.

Yuchong Sun, Hongwei Xue, Ruihua Song, Bei Liu, Huan Yang, and Jianlong Fu (2022). "Long-Form Video-Language Pre-Training with Multimodal Temporal Contrastive Learning". In: *arXiv preprint arXiv:2210.06031*.

Dıdac Surıs, Sachit Menon, and Carl Vondrick (2023). "Vipergpt: Visual inference via python execution for reasoning". In: *arXiv preprint arXiv:2303.08128*.

Pranjal Swarup, Peng Chen, Rong Hou, Pinjia Que, Peng Liu, and Adams Wai Kin Kong (2021). “Giant panda behaviour recognition using images”. In: *Global Ecology and Conservation*.

Mingkang Tang, Zhanyu Wang, Zhenhua Liu, Fengyun Rao, Dian Li, and Xiu Li (2021). “Clip4caption: Clip for video caption”. In: *Proceedings of the 29th ACM International Conference on Multimedia*.

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou (2019). “Coin: A large-scale dataset for comprehensive instructional video analysis”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen (2014). “Storygraphs: visualizing character interactions as a timeline”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen (2015a). “Book2movie: Aligning video scenes with book chapters”. In: *Proc. CVPR*.

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen (2012a). ““Knock! Knock! Who is it?” probabilistic person identification in TV-series”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen (2012b). ““Knock! Knock! Who is it?” Probabilistic Person Identification in TV Series”. In: *Proc. CVPR*.

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen (2015b). “Aligning plot synopses to videos for story-based retrieval”. In: *International Journal of Multimedia Information Retrieval*.

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen (June 2015c). “Book2Movie: Aligning Video scenes with Book chapters”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Makarand Tapaswi, Marc T. Law, and Sanja Fidler (2019). “Video Face Clustering with Unknown Number of Clusters”. In: *Proc. ICCV*.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler (2016). “MovieQA: Understanding Stories in Movies through Question-Answering”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

N Tinbergen (1963). *On aims and methods of ethology. Zeit fur Tier 20: 410–33*.

Atousa Torabi, Christopher Pal, Hugo Larochelle, and Aaron Courville (2015). “Using descriptive video services to create a large data source for video annotation research”. In: *arXiv preprint arXiv:1503.01070*.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou (2020). "Training data-efficient image transformers & distillation through attention". In: *arXiv preprint arXiv:2012.12877*.

Du Tran, Heng Wang, L. Torresani, Jamie Ray, Y. LeCun, and Manohar Paluri (2018). "A Closer Look at Spatiotemporal Convolutions for Action Recognition". In: *CVPR*.

Ly Duyen Tran, Binh Nguyen, Liting Zhou, and Cathal Gurrin (2023). "MyEachtra: Event-Based Interactive Lifelog Retrieval System for LSC'23". In: *Proceedings of the 6th Annual ACM Lifelog Search Challenge*.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill (2021). "Multimodal few-shot learning with frozen language models". In: *NeurIPS*.

Elsbeth A van Dam, Lucas PJJ Noldus, and Marcel AJ van Gerven (2020). "Deep learning improves automated rodent behavior recognition within a specific experimental setup". In: *Journal of neuroscience methods*.

Gül Varol, Ivan Laptev, and Cordelia Schmid (2018). "Long-term Temporal Convolutions for Action Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh (2015). "Cider: Consensus-based image description evaluation". In: *Proc. CVPR*.

Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko (2015). "Sequence to Sequence – Video to Text". In: *ICCV*.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko (2014). "Translating videos to natural language using deep recurrent neural networks". In: *arXiv preprint arXiv:1412.4729*.

Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler (2018). "MovieGraphs: Towards Understanding Human-Centric Situations from Videos". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan (2016). "Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge". In: *IEEE transactions on pattern analysis and machine intelligence*.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba (2016). "Anticipating visual representations from unlabeled video". In: *Proc. CVPR*.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux (2021). "Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation". In: *arXiv preprint arXiv:2101.00390*.

Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu (2018). "Bidirectional attentive fusion with context gating for dense video captioning". In: *Proc. CVPR*.

Jue Wang and Anoop Cherian (2018). "Learning Discriminative Video Representations Using Adversarial Perturbations". In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Cham: Springer International Publishing.

L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool (2019). "Temporal Segment Networks for Action Recognition in Videos". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

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: *European conference on computer vision*. Springer.

Liwei Wang, Yin Li, and Svetlana Lazebnik (2016). "Learning deep structure-preserving image-text embeddings". In: *CVPR*.

Mengmeng Wang, Jiazheng Xing, and Yong Liu (2021). "Actionclip: A new paradigm for video action recognition". In: *arXiv preprint arXiv:2109.08472*.

Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo (2021). "End-to-end dense video captioning with parallel decoding". In: *Proc. ICCV*.

Teng Wang, Huicheng Zheng, Mingjing Yu, Qian Tian, and Haifeng Hu (2020). "Event-centric hierarchical representation for dense video captioning". In: *IEEE Transactions on Circuits and Systems for Video Technology*.

Xiaohan Wang, Linchao Zhu, and Yi Yang (2021a). *T2VLAD: Global-Local Sequence Alignment for Text-Video Retrieval*. arXiv: 2104.10054 [cs.CV].

Xiaohan Wang, Linchao Zhu, and Yi Yang (June 2021b). "T2VLAD: Global-Local Sequence Alignment for Text-Video Retrieval". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He (2018). "Non-local neural networks". In: *CVPR*.

Yujia Wang, Wei Liang, Haikun Huang, Yongqi Zhang, Dingzeyu Li, and Lap-Fai Yu (2021). "Toward automatic audio description generation for accessible videos". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.

Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman (2018). "Learning and using the arrow of time". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron Weiss, Kevin Wilson, and John Hershey (2020). "Unsupervised sound separation using mixture invariant training". In: *NeurIPS*.

Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick (June 2019a). "Long-Term Feature Banks for Detailed Video Understanding". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick (2019b). "Long-term feature banks for detailed video understanding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Chao-Yuan Wu, Ross B. Girshick, Kaiming He, Christoph Feichtenhofer, and Philipp Krahenbuhl (2020). "A Multigrid Method for Efficiently Training Video Models". In: *CVPR*.

Chao-Yuan Wu and Philipp Krahenbuhl (June 2021a). "Towards Long-Form Video Understanding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chao-Yuan Wu and Philipp Krahenbuhl (2021b). "Towards long-form video understanding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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: *arXiv preprint arXiv:2201.08383*.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy (2018). "Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification". In: *ECCV*.

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*.

Yu Xiong, Qingqiu Huang, Lingfeng Guo, Hang Zhou, Bolei Zhou, and Dahua Lin (2019). "A graph-based framework to bridge movies and synopses". In: *Proc. ICCV*.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer (2021). "Videoclip: Contrastive pre-training for zero-shot video-text understanding". In: *arXiv preprint arXiv:2109.14084*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui (2016). "Msr-vtt: A large video description dataset for bridging video and language". In: *CVPR*.

Rui Yan, Mike Zheng Shou, Yixiao Ge, Alex Wang, Xudong Lin, Guanyu Cai, and Jinhui Tang (2021). "Video-Text Pre-training with Learned Regions". In: *ArXiv*.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid (Oct. 2021). "Just Ask: Learning To Answer Questions From Millions of Narrated Videos". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid (2022). "Zero-Shot Video Question Answering via Frozen Bidirectional Language Models". In: *arXiv preprint arXiv:2206.08155*.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid (2023). "Vid2Seq: Large-Scale Pretraining of a Visual Language Model for Dense Video Captioning". In: *arXiv preprint arXiv:2302.14115*.

Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang (2016). "WIDER FACE: A Face Detection Benchmark". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yao-Yuan Yang, Moto Hira, Zhaoheng Ni, Artyom Astafurov, Caroline Chen, Christian Puhrsch, David Pollack, Dmitriy Genzel, Donny Greenberg, Edward Z Yang, et al. (2022). "Torchaudio: Building blocks for audio and speech processing". In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo (2016). "Image captioning with semantic attention". In: *CVPR*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier (2014). "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions". In: *Transactions of the Association for Computational Linguistics*. URL: https://www.aclweb.org/anthology/Q14-1006.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu (2022). "CoCa: Contrastive Captioners are Image-Text Foundation Models". In: *Transactions on Machine Learning Research*.

Jiwen Yu, Haibo Su, Junnan Liu, Zhizheng Yang, Zhouyangzi Zhang, Yixin Zhu, Lu Yang, and Bingliang Jiao (2019). "A strong baseline for tiger re-id and its bag of tricks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*.

Youngjae Yu, Jiwan Chung, Heeseung Yun, Jongseok Kim, and Gunhee Kim (2021). "Transitional adaptation of pretrained models for visual storytelling". In: *Proc. CVPR*.

Youngjae Yu, Jongseok Kim, and Gunhee Kim (2018). "A joint sequence fusion model for video question answering and retrieval". In: *ECCV*.

Youngjae Yu, Jongseok Kim, Heeseung Yun, Jiwan Chung, and Gunhee Kim (2020). "Character grounding and re-identification in story of videos and text descriptions". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer.

Jiahong Yuan, Neville Ryant, Mark Liberman, Andreas Stolcke, Vikramjit Mitra, and Wen Wang (2013). "Automatic phonetic segmentation using boundary models." In: *Interspeech*.

Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici (2015). "Beyond short snippets: Deep networks for video classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Beste F Yuksel, Pooyan Fazli, Umang Mathur, Vaishali Bisht, Soo Jung Kim, Joshua Junhee Lee, Seung Jung Jin, Yue-Ting Siu, Joshua A Miele, and Ilmi Yoon (2020). "Human-in-the-loop machine learning to increase video accessibility for visually impaired and blind users". In: *Proceedings of the 2020 ACM Designing Interactive Systems Conference*.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny (2021). "Barlow twins: Self-supervised learning via redundancy reduction". In: *International Conference on Machine Learning*. PMLR.

Andrew Zhai and Hao-Yu Wu (2019). "Classification is a Strong Baseline for Deep Metric Learning". In: *BMVC*.

X Zhai, A Kolesnikov, N Houlsby, and L Beyer (2021). "Scaling vision transformers. arXiv". In: *arXiv preprint arXiv:2106.04560*.

Bowen Zhang, Hexiang Hu, and Fei Sha (2018). “Cross-modal and hierarchical modeling of video and text”. In: *ECCV*.

Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li (2021). “Tip-adapter: Training-free clip-adapter for better vision-language modeling”. In: *arXiv:2111.03930*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi (2020). “Bertscore: Evaluating text generation with bert”. In: *Proc. ICLR*.

Zhou Zhao, Zhu Zhang, Shuwen Xiao, Zhenxin Xiao, Xiaohui Yan, Jun Yu, Deng Cai, and Fei Wu (2019). “Long-form video question answering via dynamic hierarchical reinforced networks”. In: *IEEE Transactions on Image Processing*.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba (2017). “Places: A 10 million image database for scene recognition”. In: *IEEE transactions on pattern analysis and machine intelligence*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu (2021). “Learning to prompt for vision-language models”. In: *arXiv preprint arXiv:2109.01134*.

Luowei Zhou, Chenliang Xu, and Jason Corso (2018a). “Towards automatic learning of procedures from web instructional videos”. In: *AAAI*.

Luowei Zhou, Chenliang Xu, and Jason J Corso (2018b). “Towards Automatic Learning of Procedures From Web Instructional Videos”. In: *AAAI*.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong (2018c). “End-to-end dense video captioning with masked transformer”. In: *CVPR*.

Linchao Zhu and Yi Yang (2020). “Actbert: Learning global-local video-text representations”. In: *CVPR*.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015). “Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books”. In: *Proc. ICCV*.

Appendix A

Statements of Authorship

In this thesis, we include an authorship declaration for every multi-authored paper. Each declaration outlines the distinct research contributions made by the candidate and co-authors for each publication included in the thesis. For every publication featured in the thesis, there is a complete statement that has been been prepared, signed, and endorsed by both the candidate and the supervisor.

Statement of Authorship for the paper "Condensed Movies: Story Based Retrieval with Contextual Embeddings".

Paper title	Condensed Movies: Story Based Retrieval with Contextual Embeddings
Authors	**Max Bain**, Arsha Nagrani, Andrew Brown, Andrew Zisserman
Publication status	Published
Publication details	Asian Conference on Computer Vision (ACCV), 2020.

Student Confirmation

Student name	Max Bain	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • collection of datasets • writing and presentation of the paper	
Signature and Date		June 13th, 2023

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		June 13th, 2023

Statement of Authorship for the paper "Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval".

Paper title	Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval
Authors	**Max Bain**, Arsha Nagrani, Gül Varol, Andrew Zisserman
Publication status	Published
Publication details	International Conference on Computer Vision (ICCV), 2021.

Student Confirmation

Student name	Max Bain	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • collection of datasets • writing and presentation of the paper	
Signature and Date		June 13th, 2023

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		June 13th, 2023

Statement of Authorship for the paper "A CLIP-Hitchhiker's Guide to Long Video Retrieval".

Paper title	A CLIP-Hitchhiker's Guide to Long Video Retrieval
Authors	**Max Bain**, Arsha Nagrani, Gül Varol, Andrew Zisserman
Publication status	Unpublished and unsubmitted work written
Publication details	–

Student Confirmation

Student name	Max Bain	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • collection of datasets • writing and presentation of the paper	
Signature and Date		June 13th, 2023

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		June 13th, 2023

Statement of Authorship for the paper "WhisperX: Time-Accurate Transcription of Long-Form Audio".

Paper title	WhisperX: Time-Accurate Transcription of Long-Form Audio
Authors	**Max Bain**, Jaesung Huh, Tengda Han, Andrew Zisserman
Publication status	Published
Publication details	INTERSPEECH, 2023.

Student Confirmation

Student name	Max Bain	
Contribution to the paper	First-author contribution: • conception of research ideas • design, implementation and evaluation of models • writing and presentation of the paper	
Signature and Date		June 13th, 2023

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		June 13th, 2023

Statement of Authorship for the paper "AutoAD: Movie Description in Context".

Paper title	AutoAD: Movie Description in Context
Authors	Tengda Han*, **Max Bain***, Arsha Nagrani, Gül Varol, Weidi Xie, Andrew Zisserman (* Equal Contribution)
Publication status	Published
Publication details	IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

Student Confirmation

Student name	Max Bain	
Contribution to the paper	Joint first-author contribution: • joint conception of the idea • design and implementation of data collection pipeline • collection of Audiovault and MAD-v2 dataset • writing and presentation of the paper	
Signature and Date		June 13th, 2023

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		June 13th, 2023

Statement of Authorship for the paper "AutoAD II: The Sequel – Who, When, and What in Movie Audio Description".

Paper title	AutoAD II: The Sequel – Who, When, and What in Movie Audio Description
Authors	Tengda Han, **Max Bain**, Arsha Nagrani, Gül Varol, Weidi Xie, Andrew Zisserman
Publication status	Submitted for Publication in a manuscript style
Publication details	International Conference on Computer Vision (ICCV), 2023.

Student Confirmation

Student name	Max Bain	
Contribution to the paper	Second author contribution: • joint conception of the idea • design, implementation and evaluation of time point prediction • curation of character bank cast and features • writing and presentation of the paper	
Signature and Date		June 13th, 2023

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		June 13th, 2023

Statement of Authorship for the paper "Automated Audiovisual Behavior Recognition in Wild Primates".

Paper title	Automated Audiovisual Behavior Recognition in Wild Primates
Authors	**Max Bain**, Arsha Nagrani, Daniel Schofield, Sophie Berdugo, Joana Bessa, Jake Owen, Kimberley J. Hockings, Tetsuro Matsuzawa, Misato Hayashi, Dora Biro, Susana Carvalho, Andrew Zisserman
Publication status	Published
Publication details	*Science Advances* 7, no. 46, 2021.

Student Confirmation

Student name	Max Bain	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • collection of datasets • writing and presentation of the paper	
Signature and Date		June 13th, 2023

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		June. 20th 2023