# Towards Human-centric Story Understanding in Video


Supervised by Professor Andrew P. Zisserman

Bruno Korbar

St.Cross College

University of Oxford

A thesis submitted for the degree of

*Doctor of Philosophy*

Trinity 2024

# Abstract

With endless amounts of data being uploaded every day, the potential for swift development of artificial intelligence has never been higher. Videos in particular contain a plethora of information for learning about the world. We can discern actions, interactions, movement patterns, speech, etc. But all too often, research tends to group and classify: a dog is a dog is a dog.

One of the challenges lies in transcending conventional class-based visual understanding and exploring the realm of instances. This thesis concerns itself with both named instances – more specific than traditional classes – and open-world, open-set instances – more general than conventional class frameworks. In it, we discuss methods that address these challenges and could later serve as building blocks for holistic story understanding.

The thesis is structured in two broad themes: (1) identity-agnostic video understanding methods, and (2) personalisation of various video understanding tasks.

We first develop methods that are class-agnostic, and serve towards better tracking, re-identification, retrieval and semantic video processing. Our work demonstrates that localisation and re-identification of a person or an object in a video can be trained jointly, using semantically-initialised embeddings. Furthermore, we show that by designing a *task-agnostic* video sampler, we can increase the number of frames a large-language model can process, allowing us to learn from progressively longer videos.

We then focus on making video-understanding tasks identity dependent. We first design a method that tackles problems of compound retrieval, being able to jointly reason about '*who* is doing *what* and *where*'. We then generalise this approach to work on not only humans, but any arbitrary object. We show that large visual-language models can recognise a specific instance (e.g. 'my dog Chia') amongst a

large corpus of images. Finally, we recognise that not only visual representations, but also speech needs to be personalised. To this end, we present a method able to assign character names to speech segments even across multiple TV shows. Thus, we demonstrate crucial building blocks necessary for a more in-depth story understanding.

**Keywords** – video understanding, deep learning

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

Bruno Korbar, October 2024.

# Acknowledgement

No researcher is an island. This work wouldn't be possible without the support of many kind people. This thesis is dedicated to all of you.

First and foremost, to Andrew Zisserman – thank you for your trust and guidance for the past four years. Your service to the field of computer vision is exemplary, and I couldn't have imagined having a better supervisor. Thank you for your flexibility with respect to timing and geographical locations, understanding of medical issues, countless rewrites of poorly written manuscripts, and general kindness. I will forever be inspired by your creativity and dedication to your students and will strive towards the ideal you have set.

To Lorenzo Torresani, for sending me off on this path and taking a gamble on an undergrad student. Hopefully, this serves as a proof that I did occasionally listen to your advice.

To all the VGG members, those met in person and virtually, I owe my thanks for the jolly good time we have spent together. To Andrew (B), Luke, Daffy, Lili, Sagar, Tengda, and Sam for all the conversations that indulged and encouraged my inner cynic. To Andreea and Tom for many (and still way too few) lunches at St. John's. To Jaesung for being collaborator and friend par excellence, making sure I don't run out out of Peppero and sharing your mum's precious kimchi recipe. To Vladimir, Anna, and Olivia, for countless meals, walks, and coffees. Last but certainly not least, big thanks to Ash, Abhishek, David, Jenny, and Cassandra for their technical and logistical support. It was an honour to work amongst you.

And I would be remiss not to share my gratitude to the research community outside of VGG: Ivan Laptev and Joseph Sivic for offering their mentorship; to Christoph Feichtenhofer, Gedas Bertasius, Rohit Girdhar (and many others) for enticing conversations, advice, and opportunities along the years; to Ralf Gommers and the entire Quansight team for their mentoring and financial support during my PhD; to Alessio Tonioni, Yon-

qin Xian, Federico Tombari, and all the kind people at Google Zurich for their support and patience during my internship; to Andrea Vedaldi and Ishan Misra for supporting my application in the first place.

Many thanks are also owed to the broader Oxford community: to the abominable roommate, Dr. Paul, for tolerating my messiness and sharing many meals together; to Will, Dom, and Alfie, for keeping me happily caffeinated; to the staff at Churchill hospital, for getting me through this alive.

To my own community: to Robert for always engaging in friendly rivalry (I finally admit, Cambridge is nicer); to Tonka for being by my side when things were rough, never quitting on this entire project even when I was ready to, and for being an unfaltering support every step of the way, for finding joy when I couldn't, and being a voice of reason when I needed one; to Sonja for all adventures it took us on; to Kasia, Jon, Marija, Wei, and everyone else who made me feel at home in the UK; to Danko, for teaching me that sometimes, it's ok to quit. To all of my Croatian and American friends who never shied away from buying a beer for a poor PhD student.

To my family: to my dad Boris who never asked when this will be finished and to my mum Gordana who constantly asked when this will be finished; to ujo Zvonko and striko Kiko for making me laugh for the past 30 years; to deda Andrej for showing me his "exemplary" PhD thesis about 500 times and nona Milena for making sure I always have a chocolate in my bag; to Ivana and Lara for showing me that it's possible to stay sane with a crazy family; to baka Branka, who never got to see this, for teaching me how to be kind.

And finally, to Tamy. For every dream, every adventure, every long-distance call, and every moment spent together. Thank you for saving my life.

# Contents

# Chapter 1

# Introduction and Background

An image is said to be worth a thousand words. A video, then, contains a library's worth of information in a single file. For instance, a film narrates the human condition through a myriad of small actions performed by actors, which collectively build the larger story. Each small action conveys information about a character's reasoning and decision-making faculties, woven into a narrative that defines that character's identity, motivations, and intentions. In a sense, our actions manifest our human intelligence. Being able to construct a character from a sequence of their actions is therefore a significant milestone in the development of artificial intelligence.

The goal of this thesis is to explore techniques and methodologies that will allow us to do so. We want to be able to track and identify persons or objects from a video simply given a template, as this would allow us to connect various story arcs together. Then a longer story-arc (in terms of time) needs to be reasoned over, usually in language. We want to ensure that each action, object, and sound can be attributed to an instance – an object or a person.

Such ambitious goals are met with many challenges and naturally, this thesis only deals with a small subset of them. For example, many of the tasks that we explore are class-specific. Tracking, one of the fundamentals for understanding who is where in a video, relies on the prior knowledge of classes, and requires a classifier that groups objects of a same class (e.g. humans) together. Being able to track any object regardless of its class, while still keeping identities of every tracked object separate distinct is an important building block for story understanding.

Similarly, it is important to be able to find a relevant section of the video to the story or the individual. In recent literature, all current video samplers have been task- and class-specific. For example, a current sampler might be able to identify all frames relevant to "petting the dog", but not to find every frame containing a "dog running in the field".

Lastly, to truly understand stories, one needs to be able to distinguish between named instances and characters. Current research allows us to retrieve videos containing a 'person running' or a 'dog in a field', or to determine that someone is speaking. But current models struggle if we want to find 'Tom Cruise running' or 'my dog Chia in the field'.

In this thesis, we examine challenges in story understanding from two distinct angles. First, we examine task-agnostic technical aspects of the problem, developing methods for tracking and reasoning over increasingly longer video sequences. We design a tracker that, instead of on the object classifier, relies on the matching of multiple queries. We describe a task-agnostic method for long video sampling, allowing us to process longer videos by conditioning the sampler in text.

Second, we address the issue of personalization in various story-understanding tasks, including diarization and moment retrieval. We first demonstrate a method for embedding identities into search queries by learning correlation between persons' faces, names, and scenes they appear in. We further relax this by describing a method which can correlate any object's name and image and embed that knowledge into a large visual-language model. Finally, we show that we can learn how to link people's names to their voices in an automatic fashion, allowing us to recognise automatically whether the person speaking is the author of this thesis or Tom Cruise.

This thesis focuses on a few key challenges in human-centric video understanding, which we believe are crucial pieces of a wider story understanding puzzle.

## 1.1 Related Work

It is important to note that most video understanding tasks are inherently human-centric. Many films and videos we create and upload are centered on human

subjects from the outset. These videos provide researchers with the material to build datasets tailored for tasks such as action classification [Kay et al. 2017; Soomro et al. 2012], tracking and localization [Milan et al. 2016; Dendorfer et al. 2020], and embodied (ego-centric) action recognition and prediction [Grauman et al. 2022; Damen et al. 2018].

This focus on human-centric tasks might stem from a fundamental human interest in stories and interactions involving other people. Researchers have explored this interest in various studies aimed at understanding human interactions and relationships [Kukleva et al. 2020; Vicol et al. 2018], emotions [Albanie et al. 2020], characters [Tapaswi et al. 2016], and events [Lei et al. 2018].

**Understanding Actions** The research direction that has significantly advanced our understanding of video content is action recognition, which involves identifying actions within videos [Laptev et al. 2008]. Most modern approaches utilize 3D convolutional networks (ConvNets). 3D ConvNets [Taylor et al. 2010; Tran et al. 2015; Tran et al. 2018] extend 2D image models [Krizhevsky et al. 2012a; K. He et al. 2016; Simonyan and Zisserman 2014] to the spatiotemporal domain, processing both spatial and temporal dimensions in a unified manner.

However, action recognition alone does not provide information about when an action occurs. To address this temporal aspect of understanding human actions, the field of action localization [Jain et al. 2014; Shou et al. 2016; Shou et al. 2017; Huijuan Xu et al. 2017; Y. Zhao et al. 2017] has emerged. The objective of action localization is to precisely identify the start and end times of each action within an untrimmed video and to classify the action. This is often achieved through a two-step process [Buch et al. 2017; Escorcia et al. 2016; Buch et al. 2017; J. Gao et al. 2017; Ruohan Gao et al. 2018; Heilbron et al. 2016; Tianwei Lin et al. 2018; Alwassel et al. 2018]. First, an action proposal method generates candidate action segments. Then, a more sophisticated approach validates the class of each candidate segment and refines its temporal boundaries.

**Speech in Videos** Speaker diarization aims to address the question of "who spoke when" within a given audio file containing human speech. Most current approaches in this field either rely on clustering techniques [Q. Wang et al. 2018;

A. Zhang et al. 2019; Kwon et al. 2021] or utilize end-to-end models to detect and distinguish between speakers [Fujita et al. 2019; Horiguchi et al. 2020].

**Character Identification in Videos** Labelling people in videos is a well-studied topic in computer vision [Everingham et al. 2009; Haurilet et al. 2016; Nagrani and Zisserman 2017]. Most approaches depend on the availability of prior information, such as scripts [Everingham et al. 2009] or image-label pairs [Nagrani and Zisserman 2017]. However, recent studies have reduced this dependency by employing automatic evidence search through search engine APIs [Brown et al. 2021a].

**Understanding Stories and Narrative Through Language** Once we have a grasp of individual actions, the next step is to extend our understanding to the narrative context in which these actions occur. To this end, various methods and large datasets have been developed with the specific goal of enabling video comprehension through textual descriptions [Miech et al. 2019a; Damen et al. 2018; Grauman et al. 2022; Radford et al. 2021]. A key task in this area is to retrieve the correct video clip based on a text description of the narrative.

Efforts have also been made to understand people and stories depicted in movies. Early works by Everingham *et al.* [Everingham et al. 2006] Cour *et al.* [Cour et al. 2008] and Sankar [Sankar et al. 2009] explored video understanding through language by aligning videos with movie scripts. Identifying key relationships [V. Delaitre et al. 2011; Kukleva et al. 2020] and determining scene semantics [Vincent Delaitre et al. 2012]. More recently, research has focused on retrieving the appropriate video clip given the plot summary of a movie [Bain et al. 2020a], character grounding [Rohrbach et al. 2017] as well as generating audio descriptions from videos and metadata [T. Han et al. 2023b; T. Han et al. 2023a; T. Han et al. 2024].

**Large Language Models** The integration of large language models (LLMs) [Vaswani et al. 2017a] into computer vision has enabled more sophisticated and context-aware image and video understanding [Maaz et al. 2023]. For example, the BLIP family of models [J. Li et al. 2022; J. Li et al. 2023a; Dai et al. 2023] enabled a wide range of visual-to-language tasks such as image and video captioning [Jun Xu

et al. 2016] and visual question-answering [J. Xiao et al. 2021].

## 1.2 Thesis Outline and Contributions

In this section, we summarise the main contributions of the thesis and outline each chapter. The technical part of the thesis is divided into two main topics: *1)* Class-Agnostic Video Understanding, and *2)* Personalisation of Video Understanding Tasks. We discuss the contributions of each chapter briefly below and in full detail in concluding chapters of the thesis.

### 1.2.1 Class-Agnostic Video Understanding

In this section, we present our contributions regarding the methodologies used for various video-understanding tasks. These methodologies enable further work on the personalisation of video tasks.

In Chapter 2, we investigate a novel class-agnostic method for tracking and re-identification of people and objects in video scenes. The key insight of this work is the use of multiple semantically linked queries that describe each track, without the need for additional annotation data. This work was released as a technical report.

In Chapter 3, we examine bridging the gap between the visual and text domains for extremely long videos. Our method allows us to solve challenging tasks by reasoning over a large number (up to 180) of frames. We achieve this by resampling semantic queries using an efficient, task-dependent cross-attention transformer. Unlike in Chapter 2, where we use semantically linked queries, we condition the model's input directly from text—in other words, we 'ask' the model for the features we want. This architecture enables us to tackle a variety of different tasks, ranging from text-to-video retrieval to question answering. This work was accepted at ECCV 2024.

### 1.2.2 Personalisation of Video Understanding Tasks

The ideas from the previous topic enable us to identify the same person if they appear in a video multiple times or condition frame sampling to look for a specific

person. The goal of this chapter is to demonstrate how various video tasks can be personalised for a unique individual (or object). In other words, it addresses the question of whether we can retrieve 'Tom Cruise' running as opposed to 'the author of this thesis', or answer questions about 'my coffee mug' as opposed to 'my supervisor's coffee mug'.

**Personalising Action Retrieval With Multi-Modal Queries** Traditional text-to-image or text-to-video retrieval tasks aim to retrieve an image given a specific text prompt. In Chapter 4, we demonstrate how we can train a model to retrieve the images or videos of famous people in complex scenarios using their names or faces. We curate a benchmark dataset of videos and text pairs in the format 'Someone is doing something somewhere'. Using this dataset, we present a method that can query a large video corpus with a personalised query using either an actor's name or an image. This work was accepted at BMVC 2022.

**Meta-personalisation of Retrieval** The task of retrieval has been revolutionized by the introduction of large visual-language models. These models are trained on billions of images of every imaginable semantic category. They, however, do not have capability to reason about a specific instance such as 'my phone' or 'my dog Chia'. To this end, in Chapter 5, we design an adaptation method that can learn personalised concepts from text and template images using very few examples. This work is released as a technical report and under review for CVPR 2025.

**Audio-Visual Character-Aware Diarisation** Diarisation is a task that answers the question 'who is saying what and when'. Traditionally, however, diarised segments are not annotated with speakers' names but rather grouped as ('speaker one', 'speaker two', etc.). In Chapter 6, we demonstrate that diarisation can be personalised, meaning that names can be automatically assigned to spoken words with minimal labelled data. This innovation enables the community to build text and video paired datasets at a significantly reduced cost. It also has the potential to enhance the accessibility of subtitles for hearing-impaired individuals. This work was accepted at ICASSP 2024.

# Chapter 2

# End-to-end Tracking with a Multi-query Transformer

*The paper was released as a technical report.*

This chapter addresses the issue of class-agnostic tracking. Knowing who or what is where at any given point in a video is one of the key challenges for holistic story understanding.

# End-to-end Tracking with a Multi-query Transformer

Bruno Korbar Andrew Zisserman

Visual Geometry Group, University of Oxford

March 29, 2026

**Abstract**

Multiple-object tracking (MOT) is a challenging task that requires simultaneous reasoning about location, appearance, and identity of the objects in the scene over time. Our aim in this paper is to move beyond tracking-by-detection approaches, that perform well on datasets where the object classes are known, to class-agnostic tracking that performs well also for unknown object classes. To this end, we make the following three contributions: first, we introduce *semantic detector queries* that enable an object to be localized by specifying its approximate position, or its appearance, or both; second, we use these queries within an auto-regressive framework for tracking, and propose a multi-query tracking transformer (*MQT*) model for simultaneous tracking and appearance-based re-identification (reID) based on the transformer architecture with deformable attention. This formulation allows the tracker to operate in a class-agnostic manner, and the model can be trained end-to-end; finally, we demonstrate that *MQT* performs competitively on standard MOT benchmarks, outperforms all baselines on generalised-MOT, and generalises well to a much harder tracking problems such as tracking any object on the TAO dataset.

## 2.1 Introduction

The objective of this paper is *multi-object tracking* (MOT) – the task of determining the spatial location of multiple objects over time in a video. This is a very well researched area and, broadly, two approaches are dominant: the first is *tracking-*

*by-detection*, where a strong *object category detector* is trained for the object class of interest, for example a person or a car. This approach proceeds in two steps: the detector is first applied independently on each frame, and in the second step, the tracking task reduces to the data association of grouping these detections over time (over the frames in this case). Examples of this approach include [Bergmann et al. 2019; X. Zhou et al. 2020; Zhongdao Wang et al. 2020; L. Chen et al. 2018]. The second approach is *class agnostic tracking* where *any* object can be tracked. The object of interest is specified by a bounding box or segmentation in one frame, and the task is then to track that object through the other frames. Examples of this approach include [Bertinetto et al. 2016; Danelljan et al. 2017; Held et al. 2016].


Figure 2.1: An overview of the functionality of the multi-query tracking transformer (*MQT*). Each frame generates location and appearance embeddings of the target object. These embeddings are used as queries for the subsequent frame. By propagating information between frames in this simple manner the object is tracked over time through the video.

The tracking-by-detection approach generally outperforms class-agnostic models at the moment, but the approaches often suffer from overly complex processing pipelines (using multiple separately trained models for each step) and they rely on prior knowledge of the object class of interest. More importantly the detection model and the data-association model are in tension: with one model trained to tolerate object class variations (to better detect all instances of the same class), whilst the other is trained to maximise discrimination of two instances of the same class (to prevent identity switching). Such models are generally not trained end-to-end. Lastly, such models are highly specific – the results of these trackers often don't generalise well to the more general tracking scenario [Dave et al. 2020].

In this paper, we present a *class-agnostic* tracker that can be trained end-to-end,

but also build on the lessons of a strong object category detector. To this end we base the tracker on the DETR object category detector [Carion et al. 2020], using a transformer-detector modified in such a way that it can attend to multiple objects locations and identities simultaneously. We introduce dual 'object-specific location' and 'identity' encodings (dubbed *semantic queries*) which allow the model to selectively focus on the *location* or *appearance* of objects we want to track, irrespective of their classes. These object-specific embeddings enable the model to be optimized *jointly* for track prediction and re-identification by training in a class agnostic manner. In this way we achieve a single model class-agnostic tracker that performs competitively on several MOT benchmarks [Milan et al. 2016; Dendorfer et al. 2020], and can outperform all previous work on the class agnostic-MOT task [Bai et al. 2021] where the class-prior is not known. Lastly, we show that the tracker trained in this way can also generalise well to tracking task such as TAO [Dave et al. 2020], where the categories and number of tracking targets is far more general than on MOT benchmarks.

To summarise, we make the following three contributions: First, we introduce the concept of semantic detector-queries and show their effectiveness for multi-object tracking. Second, we design a transformer-based class-agnostic tracking model around semantic detector-queries that is capable of simultaneous detection and re-identification of multiple objects in the scene. Finally, we achieve competitive results on various MOT benchmarks [Milan et al. 2016; Dendorfer et al. 2020] where object identity is used, demonstrate state-of-the-art class-agnostic performance on generalized MOT [Bai et al. 2021], and show the potential of the model to generalise to even harder tracking tasks on TAO [Dave et al. 2020].

## 2.2 Related work

To put this work into context, we compare it to the modern tracking approaches that use a similar tracking paradigm to ours. There are, of course, many other tracking approaches (e.g. tracking-by-segmentation [Osep et al. 2018; Voigtlaender et al. 2019; Z. Xu et al. 2020; Bertasius and Torresani 2020]) that are not as closely related to our method.

**Tracking by detection** approaches form trajectories by associating detections

over time [X. Zhou et al. 2020; Zhongdao Wang et al. 2020; L. Chen et al. 2018]. A common way of representing the data-association problem is to view it as a graph, where each detection is a node linked by possible edges and formulating it as a maximum-flow problem [Berclaz et al. 2011] with distance based [Pirsiavash et al. 2011; L. Zhang et al. 2008] or learned costs [Leal-Taixé et al. 2014]. Alternative formulations use association graphs [Ma et al. 2018], learned models based on motion models [C. Kim et al. 2015], or a completely learned graph-neural-network [Brasó and Leal-Taixé 2020]. A common issue with graph-based approaches is the high optimization cost that doesn't necessarily translate to better performance.

Detections can also be associated by modelling *motion* directly [Alahi et al. 2016; Leal-Taixé et al. 2011]. Pre-deep learning approaches often rely on assumptions of constant motion [Choi and Savarese 2010; Andriyenko and Schindler 2011] or existing models of human behaviour [Scovanner and Tappen 2009; Pellegrini et al. 2009; Yamaguchi et al. 2011], whilst more modern approaches attempt to learn the motion models directly from the data [Leal-Taixé et al. 2014]. Our model doesn't model motion explicitly, although, we do rely on the assumption of small motion within frames to account for appearance similarity.

**Tracking by appearance** methods use increasingly powerful image-representations to track objects based on the similarities produced by either Siamese-networks [Leal-Taixé et al. 2014; Shuai et al. 2021], learned reID features [Ristani and Tomasi 2018], or other alternative methods [L. Chen et al. 2018; Chu and Ling 2019; Pang et al. 2021].

**Tracking by regression** *refines* (instead of detecting) the bounding box of the current frame by regressing the current bounding box given the bounding box at the previous frame [Bergmann et al. 2019; Brasó and Leal-Taixé 2020; Feichtenhofer et al. 2017; X. Zhou et al. 2020]. As these models usually lack information about the object identity or relative track location, additional reID and motion models [Bergmann et al. 2019; Feichtenhofer et al. 2017; X. Zhou et al. 2020] or graph methods [Brasó and Leal-Taixé 2020] are necessary to achieve competitive performance. Our model falls roughly in this category, although we show that it can learn reID information directly from data.

**Tracking with transformers** uses aspects of the transformer architecture [Vaswani

et al. 2017b], such as self-attention and set-prediction [Carion et al. 2020; X. Zhu et al. 2021]. The *Trackformer*, a transformer tracker proposed by [Meinhardt et al. 2021], is the closest approach to ours, employing largely the same architecture model, but use class information for tracking and do not employ semantic queries. The TransTrack model [P. Sun et al. 2020] operates in the same way as [Meinhardt et al. 2021] but with a different underlying backbone. MOTR [F. Zeng et al. 2022] extends this framework by adding a "query-interaction-module" to reason about track-queries over time. [E. Yu et al. 2021] leverage the importance of semantically-decoupled embeddings. They employ the "global context disentangling unit" to separate the final layer output of a backbone CNN directly to semantic embeddings; we on another hand, do it in the transformer decoder. TrackCenter model [Y. Xu et al. 2021] introduces two key improvements: pixel-level dense-queries, and semantically-decoupled representation learning via model separation. TransMOT [Chu et al. 2021] utilises transformers in a different way, by introducing a spatio-temporal graph transformers for post detection data-association. MeMOT [Cai et al. 2022] introduces a memory module on top of the transformer encoder to further boost performance. Note that *none* of these works can be generalised to GMOT or TAO tasks, as they are tracking-by-detection approaches and cannot be used for class-agnostic tracking. For more in-depth comparison to most-similar works please refer to the supplementary material.

**Class-agnostic tracking** leverages powerful appearance embeddings to track objects based the similarity of the embeddings. The method does not leverage class information explicitly. These models often use a form of a Siamese architecture to learn a patch-based matching function [Leal-Taixé et al. 2014; Held et al. 2016; Danelljan et al. 2017; Tao et al. 2016; Bertinetto et al. 2016; Leal-Taixé et al. 2016; Shuai et al. 2021]. However, even if the model is in principle capable of class-agnostic inference, models such as [Shuai et al. 2021] are not fully class-agnostic, as they require class information for successful training of their tracker in the form of an object detection loss (that requires ground-truth class information for every object in the training triplet). Our work differs in that it does not require this explicit object class labelling.

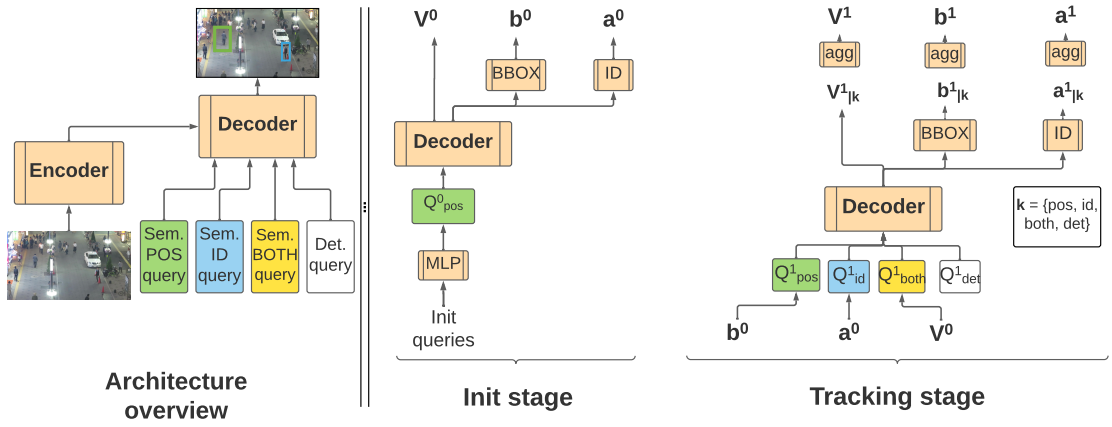



Figure 2.2: We show the high level overview of MQT on the left. Two distinct training stages with a single query during initialization, and multiple queries during tracking stage are on the right. Tracks are initialised either by using `det` (detection) queries, or with existing detections projected into semantic queries (e.g. $Q^0_{\texttt{pos}}$ as shown in the figure). Each query is then processed to obtain the decoder output ($V^0$), bounding-box prediction ($b^0$) and appearance vector ($a^0$). These are passed to the following frame in form of semantic queries, and their corresponding outputs ($V^1_{|k}$, $b^1_{|k}$, $a^1_{|k}$ respectively) are aggregated for each object to obtain final predictions ($V^1$, $b^1$, $a^1$).

## 2.3 Multi-query transformer for tracking

The goal of multi-object tracking is to obtain the trajectories of $n$ objects over a sequence of frames from a video. For example, given the initial set of object locations (bounding boxes) in the first frame, the task is to predict a new set of bounding boxes and associate them to the correct objects for every subsequent frame thus forming trajectories.

We formulate an auto-regressive tracking process as illustrated in Figure 2.1. At the current frame, the model produces three outputs for each object: (1) a bounding box of the object's location in the current frame, (2) an appearance embedding of the visual appearance of the object given its location, and (3) a raw transformer-decoder embedding. This information is then passed to the following frame in the form of semantic queries to the decoder. For the following frame, the model either looks for an object given its location, its appearance, or any additional information carried over by the raw decoder output in queries. The output embeddings are aggregated, and if the appearance output of an object at the frame $k$ matches the known appearance of the track (usually the appearance output at the frame $k-1$, but it can be earlier when using reID from memory to overcome occlusions), the location output is then added to the trajectory of an object. This makes our model applicable in generalised tracking scenarios where class information is not

available.

On a high level, this work is performed by a transformer [Vaswani et al. 2017b]. The current image is processed by a convolutional neural network and fed into a transformer-encoder, whereas all semantic queries from the previous image are fed into the transformer-decoder module – see Figure 2.2. This differs from traditional tracking-by-detection approaches (e.g. [Bergmann et al. 2019]) where detection is separate from data-association and where each step commonly uses separate embeddings. Our method merges these two steps into one, and the initial object embedding is disentangled within the transformer-decoder directly into embeddings used for detection and data-association (reID).

The rest of this section outlines the main parts of the model and their application for tracking. For more detailed information on architecture, hyperparameters and implementation details please refer to the supplementary material.

### 2.3.1 Transformer-decoder queries

The key insight of our work is the fact that the queries passed to the decoder part of a transformer can be customized for the tracking task. For example, if we know the approximate bounding box of an object from a previous frame, then a query can be formed from this bounding box and used to search for the new position of the object in its vicinity (in a similar manner to the RoI pooling module of a traditional two-stage detector that extracts the image embedding corresponding to the input bounding box, and is then used for bounding box regression and classification).

What if we wanted to update the appearance (or maybe find the location of an object defined by its appearance)? We simply extend this approach by having a query encode the object appearance.

These *semantic queries* are used in an auto-regessive manner for tracking, in that the output of the decoder of one frame is used as the query input for the subsequent frame. We also include another type of query that is not used auto-regressively, but instead is applied independently on each frame. This second type of query, termed (`det`), acts as spatial anchors in traditional detection transformers [Carion et al. 2020], and allows the tracker to find new objects. For each image, a number of

these queries are fed into the decoder module, and their outputs from the decoder feed into bounding-box and appearance heads. If any of these heads matches any recently “lost" or “current" track better than semantic queries, we assign the appearance and location outputs of that detection query to be a part of a trajectory.

### Location queries (`pos`)

Location queries are non-linear projections of an object bounding box, telling the model *where* to look for an object. A bounding box is passed through a single-layer MLP that projects it to the model dimension $d$. It is then passed through the decoder producing a $d$-dimensional embedding $V_{|pos}$, which is then passed through: (a) a bounding-box regressor head trained to produce an output bounding box $b_{|pos}$, and (b) an appearance head trained to produce an output vector $a_{|pos}$ containing the information about object's appearance. The output of the bounding-box regressor head is used as a bounding box to form trajectories for an object.

### Appearance queries (`id`)

Sometimes objects get occluded, or their location can change due to camera shift or variable frame rate. In these cases, we need a mechanism that would tell the model *what* to look for. To this end, we feed in the appearance query, which in our case is simply an output of the appearance head from a previous step $a^{i-1}$. This query is then also passed through the decoder to obtain vector $V_{|id}$ which is then passed through the bounding-box regressor and an appearance head producing outputs $b_{|id}$ and $a_{|id}$ respectively.

At tracking time, outputs of the appearance queries are used for track confirmation and re-identification (“matching" the track). Simply put, the track is only considered to be active if the cosine distance between outputs of the appearance query at the previous frame and current frame is smaller than a hyper-parameter $\tau_{\text{conf}}$, and the track is associated with $k$-th object if it is the object with smallest cosine distance to the output of the appearance query at the previous step. If a track became inactive (e.g. due to occlusions), we keep the track information for several frames in order to attempt to pick that track up again. For subsequent frames, we compare the appearance embeddings of detected objects to the “known"

inactive tracks in order to establish correspondence if the track is re-discovered. We empirically established that keeping several (3) frames worth of appearance information aids tracking performance overall.

### Joint queries (`both`)

Finally, we reason that a raw output embedding of the decoder $V$ contains valuable information about the track location and identity. Therefore, we form a joint query from $V^{i-1}$ that is simply a raw decoder output at the previous step. It is used to reinforce and strengthen the signal not captured from `pos` and `id` queries alone. They are passed through the same machinery as the formerly mentioned queries, producing outputs $V_{|both}$, $b_{|both}$ and $a_{|both}$.

### Detection queries (`det`)

Much like anchor-queries in transformer-based detectors [Carion et al. 2020], `det` queries are used detecting additional objects in tracking settings where additional detections are allowed. They are static, randomly initialized, trainable parameters used to detect and associate new objects to the existing semantic queries. They produce outputs $V_{|det}$ $b_{|det}$ and $a_{|det}$, which are then matched to the existing tracks independently from semantic queries.

For each frame, a number of `det` queries are fed into the decoder module, and their outputs then pass through the bounding-box and appearance heads. If any of these heads matches any “lost" or “current" track better than semantic queries (often caused when the objects are occluded), we assign the appearance and location outputs of that detection query to be a part of a trajectory, and the outputs corresponding to that `det` query become semantic queries for the following frame.

### Query aggregation

The eagle-eyed reader will have noticed that for every known object at a current frame $i$ there will be three sets of outputs $V_{|q}$, $b_{|q}$ and $a_{|q}$, for $q \in \{\texttt{pos}, \texttt{id}, \texttt{both}\}$ (i.e. three outputs for each of the semantic query types). To make a final prediction for that object, we need to learn which one of the decoder outputs to trust. To this end we learn an aggregation function over each set of semantic queries, $V^i_{|q}$, $b^i_{|q}$ and $a^i_{|q}$, to obtain final results $V^i$, $b^i$ and $a^i$. Final appearance output for the

frame will then be computed as

$$a^i = \phi_q(a_{|q}) \qquad\qquad q \in \{\texttt{pos}, \texttt{id}, \texttt{both}\}$$

where $\phi$ is an aggregation function. Each `det` query is passed through the aggregation function as well, but the effect is nullified as it is passed through it on its own. In our work, we use collaborative-gating as an aggregaton function [Yang Liu et al. 2019] as it performs better than other schemes (see ablation studies) such as taking an average of the embeddings. In the case where different query types are missing (e.g. at initialisation or when dealing with `det` queries), we zero-pad the missing query types. To compensate for the implicit scaling introduced by missing query types. We follow [Yang Liu et al. 2019] and remove the weights for missing queries, and then re-normalise the remaining weights such that they sum to one.

### 2.3.2 Training the tracker

In order to be able to track objects frame-to-frame, we train the our model on two adjacent frames as illustrated in Figure 2.2. To pre-train our model, we follow [Meinhardt et al. 2021] and simulate tracking data from COCO [Tsung-Yi Lin et al. 2014] (in order to run ablations on object detection and train the model for GMOT40 [Bai et al. 2021]) and Crowd-Human [S. Shao et al. 2018]. As in [Meinhardt et al. 2021], the adjacent frames are generated by applying random spatial augmentations of up to 5% with respect to the original image size. For COCO [Tsung-Yi Lin et al. 2014], we apply an additional constraint that no objects should be lost between the transformations.

At the initialization step, we compute bounding-box regression loss from either semantic or detection queries. In the tracking step, we optimize the model jointly for location prediction and appearance matching (i.e. tracking) for all objects initialized in the initialization step. We map the ground truth objects to the set of predictions from our model in one of two ways. For all known tracks we attempt to match them to the ground-truth based on the cosine similarity between the appearance vectors produced by the model in the initialisation and tracking step. If the track is “lost" or undiscovered, i.e. the appearance vector at the current step doesn’t match the appearance vector of the corresponding query at the

initialization step (e.g. due to occlusions or objects moving out of the scene), we follow [Carion et al. 2020] and try to find injective minimum cost mapping between the ground truth and the set of predictions generated by detection (det) queries. In this way, we are able to train the model end-to-end, and the model is able to simultaneously reason about both known tracks and new, undiscovered objects. Note that during training, we always use both semantic and detection queries (100 per frame, following [Carion et al. 2020]). Similarly, while the ground-truth detections are provided during training, we do not use the class information explicitly.

**Class-agnostic vs class-specific tracking:** Modern detectors [Girshick 2015; Carion et al. 2020] pass the region-of-interest embedding through regression and classification heads to obtain final bounding box and class predictions. When the object domain is known (i.e. we know the objects are people), the latter can very effectively be used in a tracking mode as a track confirmation mechanism – if an object is a pedestrian, mark it as a part of a track. While this approach is intuitive, it has several drawbacks. The first and the most obvious one is the fact that the object class has to be known in advance, in order to finetune the classification head for that particular object class. Additionally, it does not carry information about the particular instance, thus necessitating the use of some other data-association mechanism. When our model is trained for tracking, we ignore the classification head and instead determine the track validity and identity by comparing the cosine distance between two subsequent appearance outputs and picking a minimum one. While this has a slight detrimental impact on our overall tracking performance (see ablation studies), it allows us to apply our tracker in a class-agnostic fashion on generalised-MOT task, and we empirically found that our model retains track identities better.

**Loss functions:** The final transformer set-prediction loss [Carion et al. 2020] at each tracking step is computed over all semantic and detection queries, in the same fashion as [Meinhardt et al. 2021]. However, we use the proposed appearance loss instead of the class-prediction loss of [Meinhardt et al. 2021]. The appearance loss is formally defined in Sec 1.4 of the supplementary material.

### 2.3.3 Enhancing our tracking results

**Memory:** One important way of enhancing the performance of a tracker is to increase the memory of a track. Specifically, for each *known* track, we are keeping the information of the last frame in which the track was active. This makes sense as the appearance is unlikely going to change in such a short time span. However, there can be instances of occlusions, rotations, and tracks becoming inactive for (relatively) long periods of time where having more data points about the appearance of the object could be useful. Hence we explore keeping multiple frames worth of appearance information. We found that keeping the appearance information of the first and the last five frames gives us optimal results. We simply proceed to use the minimum distance between the current appearance vector and the "memory" appearance vectors.

**Multi-hypothesis tracking:** Transformer-based methods such as ours tend to suffer from a large number of false-positives as multiple query-anchors can detect the same object, an effect we largely mitigate by considering only objects that have maximum appearance similarity to the track at the previous frame. This can cause more identity shifts as in the ambiguous cases, model might shift between two tracks (which can lead to it losing the track completely). We relax this requirement and keep track of the top-$k$ possible track-candidates (that are still within the threshold $\tau_{\text{conf}}$), and then choose the longest tracked sequence as a part of our trajectory.

**Leveraging multiple queries:** Keeping multiple hypothesis for every object however can significantly increase the memory requirement, especially when keeping multiple hypothesis is not always necessary. To this end, we leverage our multi-query setup and introduce a simple heuristic in order to reduce the memory requirement in the cases where the model is completely certain in an existing track. Specifically, we define confidence as an agreement between two query types: if the distance between the appearance vectors corresponding to the pos and id queries before aggregation is smaller than a hyper-parameter $\tau_{\text{agree}}$, we do not consider additional hypothesis.

## 2.4 Experiments

In this section, we first present the results of our best model on various tracking tasks. Then, we present the ablation study in which we experimentally verify the design choices from the previous section. Final hyper-parameters for each dataset are given in the supplementary material.

### 2.4.1 Datasets and tasks

**Person-MOT:** The **MOT17** dataset consists of train and test sets, each with 7 sequences containing full-body bounding boxes of pedestrians for up to 51 people in a sequence. Three sets of public detections are provided: DPM, Faster R-CNN and SDP [Felzenszwalb et al. 2010; S. Ren et al. 2015; F. Yang et al. 2016]. For our tracking ablation studies, we select a single sequence from the training split as a validation set, following [Bergmann et al. 2019].

Compared to MOT17, **MOT20** [Dendorfer et al. 2020] is more challenging as it contains much more crowded scenes, some with up to 220 pedestrians. Finally, we evaluate our model on the “person" class of the **TAO** [Dave et al. 2020] dataset. See below for more details about the dataset.

**Generalized-MOT:** MOT challenges focus on a specific object category of interest (pedestrians) and rely on models trained specifically to recognise them. In contrast, generalised MOT (GMOT) requires no prior knowledge of the objects to be tracked, and is an evaluation of class-agnostic tracking. GMOT-40 contains 40 sequences which cover ten categories loosely related to categories of popular object detection datasets, with four sequences per category. Each sequence contains multiple objects of the same category.

**Tracking any object:** TAO [Dave et al. 2020] is a diverse dataset for a task of tracking any object. It consists of 2,907 high resolution videos, captured in diverse environments, with a vocaulary of over 800 objects. The dataset is split into “train", “validation", and “test" splits, containing 500, 988 and 1,419 videos respectively.

**Metrics:** Aspects of MOT are evaluated with different standardised metrics [Bernardin and Stiefelhagen 2008]. The community focuses on two complementary metrics: Multiple Object Tracking Accuracy (MOTA) which focuses on track coverage of the detections, and Identity F1 Score (IDF1) which focuses on identity preserva-

tion across the tracks [Ristani et al. 2016]. For TAO, we compute mAP metric using 3D IoU (with a default threshold of 0.5) as specified in [Dave et al. 2020].

**Private vs. public tracking:** For the evaluation on MOT datasets such as MOT17 [Milan et al. 2016] tracking works often refer to the *private* and *public* detections. *Public* detections are provided with the dataset and allow for a comparison of tracking methods independent of the object detection performance of the model. *Private* detection setting during evaluation allows for the use of detections obtained in any other way. For our evaluation, if the setting is marked as *public*, we strictly use detections provided with the dataset; if the setting is marked *private*, we initialise the tracks with detection `det` queries only and at each step we allow a number of them to detect potentially missed objects. In addition to these traditional settings, we denote two additional ones. If setting is marked as *private & public*, we use the detections provided *and* `det` queries, and rely on the set-matching algorithm to find an optimal mapping to the ground-truth tracks. If any of the settings is marked with "private+", it means we augment our detections with those obtained from a state-of-the-art object detection model [Z. Ge et al. 2021] trained separately and feed them into the decoder as additional `pos` queries. For a more detailed description of each stage in the context of our model, please refer to the supplementary material.

### 2.4.2 Results

**MOT17:** We evaluate our model on the MOT17 [Milan et al. 2016] test set and report the results in Table 2.1. For private detections, our results are comparable to the most modern works even though our model is not inherently trained for detection. When detection queries are added to the initialization stage (technically making our detection not truly public, but not fully private either as we do leverage the detections provided with the dataset and augment them with better detections), it gains additional 3.2 MOTA points advantage over fully public setting, and it even surpasses most trackers using private detections only.

**MOT20:** We also evaluate our model on a much more challenging MOT20 dataset. Results can be seen in Table 2.1. Our performance is on par with modern trackers in public detection setting, however, our model suffers in a private

Table 2.1: Comparison of modern MOT methods evaluated on MOT17 and MOT20 test sets. "+" in our private setting denotes use of externally computed private detections from [Z. Ge et al. 2021]. Models denoted with "*" are public methods not associated with a peer-reviewed publication. For fairer comparison, we specifically mark models with in-model association solvers (IAS).

| Method | Setting | IAS | MOT17 | | MOT20 | |
|---|---|---|---|---|---|---|
| | | | MOTA ↑ | IDF1 ↑ | MOTA ↑ | IDF1 ↑ |
| Tracktor++ [Bergmann et al. 2019] | Public | | 56.3 | 55.1 | 52.6 | 52.7 |
| CenterTrack [**zhou2020centertrack**] | Public | | 60.5 | 55.7 | | |
| *Trackformer [Meinhardt et al. 2021] | Public | ✓ | 62.5 | 60.7 | | |
| *TransCenter [Y. Xu et al. 2021] | Public | | **71.9** | 62.3 | 62.3 | 50.3 |
| SiamMOT [Shuai et al. 2021] | Public | | 65.9 | 63.1 | | |
| MQT (ours) | Public | ✓ | 65.4 | **63.3** | **63.3** | **55.8** |
| CenterTrack [**zhou2020centertrack**] | Private | | 67.8 | 64.7 | | |
| *Trackformer [Meinhardt et al. 2021] | Private | ✓ | 65.0 | 63.9 | | |
| *TransCenter [Y. Xu et al. 2021] | Private | | 73.2 | 62.2 | 61.9 | 50.4 |
| *RelationTrack [E. Yu et al. 2021] | Private | | **73.8** | **74.7** | **67.7** | **70.5** |
| MeMOT [Cai et al. 2022] | Private | ✓ | 72.5 | 69.0 | 63.7 | 66.1 |
| MQT (ours) | Private | ✓ | 66.5 | 65.2 | 62.1 | 62.5 |
| MQT (ours) | Private+ | ✓ | 68.2 | 65.8 | 66.1 | 64.5 |
| MQT (ours) | Prv&Pub | ✓ | 67.9 | 64.2 | 64.8 | 63.9 |
| MQT (ours) | Prv+&Pub | ✓ | 69.4 | 65.9 | 66.9 | 65.7 |

setting. This is largely due to the superior performance the two-stage detectors exhibit on frames with many small objects in them. When we make use of the superior detections, our results are much more competitive.

It is worth noting that RelationTrack [E. Yu et al. 2021] and TransCenter [Y. Xu et al. 2021], methods that significantly outperform ours, separate the detection part of the model completely from the data-association part whilst still training end-to-end. TransCenter, however, is limited by the detection capabilities of their backbone, which is what allows us to outperform their model on the more challenging MOT20 dataset [Y. Xu et al. 2021]. The RelationTrack algorithm fully separates the detection and association, and refines generated tracklets in a two stage procedure at inference time [E. Yu et al. 2021]. In the supplementary material we show that a similar method can aid our model as well with a trade-off in inference speed and flexibility.

**TAO-person:** Finally, we evaluate our model on the person category of the TAO dataset [Dave et al. 2020]. We tune the threshold parameters on a small hold-out section of the training set, but we do not re-train our model on TAO.

Table 2.2: Evaluation of state-of-the-art person trackers on the person category of the TAO dataset. Methods marked with '*' denote our re-implementation of the methods and the evaluation protocol. Tractor++ and *MQT* are using the same private detections whilst Trackformer uses its own detections.

| Method | MOTA ↑ | IDF1 ↑ |
|---|---|---|
| Tractor++ [Bergmann et al. 2019] | 66.6 | 64.8 |
| *Tractor++ [Bergmann et al. 2019] | 68.1 | 66.1 |
| *Trackformer [Meinhardt et al. 2021] | 71.3 | 67.2 |
| MQT (ours) | **71.9** | **69.6** |

We include our re-implementation of [Bergmann et al. 2019] and [Meinhardt et al. 2021] evaluated following the same protocol and using the same set of detections whenever possible for a fair comparison. Full results can be seen in Table 2.2. MQT outperforms the reported performance of [Bergmann et al. 2019] by 5.3 MOTA points, and improves upon our re-implementation of [Bergmann et al. 2019] and [Meinhardt et al. 2021].

### 2.4.3 Expanding tracking capabilities:

Our model formulation allows us to successfully apply our model "as-is" to more general tracking scenarios such as *class-agnostic* MOT or *tracking any object*. To demonstrate this capability, we evaluate our model on **GMOT40** [Bai et al. 2021] and **TAO** [Dave et al. 2020] datasets.

**GMOT40:** MOT17 and MOT20 datasets contain multiple instances of pedestrians on the street. Intuitively, this discards a lot of detection capacity of models trained on detection data and assumes prior knowledge of the domain. To this end, we evaluate our model on GMOT40 – a generalised MOT benchmark where categories are related (but not completely overlapping) with common detection datasets. Our model outperforms all baseline benchmarks on the test set, largely due to its class-agnostic nature. For this task, we forgo CrowdHuman [S. Shao et al. 2018] pre-training and train the models on tracking data simulated from COCO [Tsung-Yi Lin et al. 2014].

**TAO:** To demonstrate the promising generalization of our MQT model even for the task outside of the "traditional" MOT scope, we present results on the *val* split TAO dataset [Dave et al. 2020]. First three rows of Table 2.4 present "user-initialised" tracking setting with oracle ground truth class assigned using the proto-

Table 2.3: Performance of our tracker on one-shot GMOT protocol, as described by [Bai et al. 2021]

| Method | MOTA ↑ | IDF1 ↑ | MT ↑ | ML ↑ |
|---|---|---|---|---|
| MDP | 19.80 | 31.30 | 142 | 1161 |
| FAMNet | 18.00 | 28.30 | 166 | 1197 |
| Ours | **23.95** | **31.06** | **182** | **1077** |

Table 2.4: User-initialized tracking results on “val" split of the TAO dataset.

| Tracking | Detection | Track mAP |
|---|---|---|
| SORT [Dave et al. 2020] | MaskRCNN [K. He et al. 2017] | 30.23 |
| Ours | ours | 32.11 |
| Ours | ours + MaskRCNN [K. He et al. 2017] | **39.62** |

col described in [Dave et al. 2020]. Detections are either provided (MaskRCNN [K. He et al. 2017], akin to “public" setting on MOT tasks), or directly computed by us (“private" setting). Note that a lack of fully trained detector hurts our performance on this task, as can be seen by a boost seen when additional detections are provided to our model.

### 2.4.4 Ablation studies

In this section, we validate the model design choices of our tracker outlined in Section . For additional ablations, please refer to our supplementary material.

#### Tracking

Table 2.5: Ablation study of *MQT* on MOT17 held out training sequence.

| det | id | pos | both | MOTA | IDF1 |
|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | ✓ | 68.3 | 66.1 |
| ✓ | | | | 55.9 | 54.7 |
| | ✓ | | | 52.7 | 54.1 |
| | | ✓ | | 56.1 | 55.8 |
| | | | ✓ | 56.9 | 56.2 |

(a) Single vs multi-query tracking performance. For a full table with all permutations, please refer to the supplementary material.

| Method | MOTA | IDF1 |
|---|---|---|
| Heuristic | 65.7 | 64.6 |
| Avg. pool | 65.8 | 64.7 |
| Max. pool | 61.4 | 59.3 |
| Colaborative gating | 68.3 | 66.1 |

(b) Comparison of various query aggregation methods for tracking, evaluated on held-out validation sequence from MOT17.

| Class head | Track confirmation | MOTA | IDF |
|---|---|---|---|
| yes | class | 68.5 | 66.0 |
| yes | appearance | 68.2 | 66.1 |
| no | appearance | 68.3 | 66.1 |

(c) MQT trained with and without classification head. Results indicate that training the classification head is not necessary from the tracking standpoint when appearance query is used for track-confirmation.

We examine the impact of various tracking design choices on a single held-out sequence of MOT17, following the ablation procedure described in [Bergmann et al. 2019].

**Single- vs. multi-query decoder:** We investigate the gain in having multiple types of semantic queries (rather than just having `pos` only for example). The results of single-vs-multi query tracking performance are given in Table 2.5 (a), whilst full results can be found in the supplementary material. When only `det` queries are used, we get the performance akin to what we would get with a detection transformer only. Having an auto-regressive component (e.g. `both` query) improves performance by a small margin (1 MOTA point). The performance benefit of multiple queries is clear, outperforming any single-query tracker by 11 MOTA points.

**Feature aggregation – tracking:** We examine different ways to aggregate outputs with respect to multiple decoder query-types. The *heuristic* method refers to simply using the location output w.r.t. the previous location to regress the bounding box, and appearance output w.r.t. the previous appearance for track-confirmation and reID (full evaluation of heuristic method, including all permutations of three query-types can be found in the supplementary material). We found that collaborative gating [Yang Liu et al. 2019] performs better than any other aggregation method by 2.5 MOTA points. Full results can be seen in Table 2.5 (b).

Table 2.6: Various methods of improving our tracker's performance.

| # frames | dist metric | MOTA | IDF1 |
|---|---|---|---|
| 1 (F) | n/a | 67.1 | 65.8 |
| 1 (L) | n/a | 68.3 | 66.1 |
| 2 (F + L) | avg | 69.2 | 66.3 |
| 2 (L) | avg | 69.0 | 66.1 |
| 3 (F + L2) | avg | 69.6 | 66.5 |
| 6 (F + L5) | avg | 69.6 | 66.7 |
| 6 (F + L5) | min | 70.0 | 66.9 |

(a) Impact of various memory sizes (and metric for computing the appearance distance) on tracking performance, evaluated on held-out validation sequence from MOT17. (F) indicates the first frame, (L$k$) indicates the last $k$ frames.

| Number of proposals | MOTA | IDF1 |
|---|---|---|
| 1 | 68.3 | 66.1 |
| 3 | 69.7 | 67.8 |
| 5 | 70.4 | 68.4 |
| 5 (MQC) | 70.5 | 68.6 |
| 10 | 69.5 | 68.1 |

(b) Multi-hypothesis tracking: analysing the impact of keeping multiple candidates for each track. Note that multi-query confirmation (MQC) does not impact the results, but it reduces the computational requirement by only keeping multiple tracks when model confidence drops.

**Class-agnostic vs. class-specific tracking:** The majority of modern trackers that fall into tracking-by-detection category rely on class-specific information in order to achieve good performance on MOT challenges. We ask if that is really necessary, given that our model is capable of producing appearance-specific embeddings for every track. To this end, we compare two different track confirmation methods: the traditional approach which uses a class-confidence threshold to confirm the track, and our suggested approach that relies on setting a threshold for

the distance between the appearance information of two subsequent frames. In order to rule-out the effect of training the classification head on downstream performance we also show results when our scheme is used, but classification head is still being trained. Using classification score for track-confirmation is marginally more effective (by 0.3 MOTA), but we find that the performance benefit is not worth the limitations it poses (mainly inability to track “unknown" classes). Furthermore, we find that training the classification head during tracking pre-training has little to no impact on downstream tracking performance if the classification score is not being used. Full results are given in Table 2.5 (c).

**Appearance memory size:** In Table 2.6 (a), we explore the impact of various memory sizes (and metric for computing the appearance distance) on tracking performance. We either compare the distance of a current track to the minimum distance of all the embeddings (min), or to the average-pooled embedding (avg). As a broad trend, the more memory we store, the better the performance get. Due to computational constrants, we were not able to extend this beyond six frames, however, even between 3 and 6 frames the performance difference becomes marginal, indicating we might hit diminishing returns.

**Multi-hypothesis tracking:** In order to evaluate the performance benefits of multi-hypothesis tracking, we evaluate our model with a varying number of proposals at each step. The tracking performance saturates at 5 proposals, indicating that there is no benefit in keeping more than this. Furthermore, using the distance between `loc` and `id` queries as a measure of confidence in predictions of known objects doesn't impact performance of the model whilst reducing memory requirements. Full results are given in Table 2.6 (b).

## 2.5 Conclusion

We have introduced the multi-query transformer tracking model that achieves admirable performance on several known-class multi-object tracking challenges, while simultaneously outperforming all baselines on class-agnostic generalised multi-object tracking benchmark. We show the benefit of using decoupled *semantic*

decoder queries for both object detection and tracking, and we conjecture that similar strategy can be employed in different areas of computer vision.

## 2.6 Supplementary material

### 2.6.1 Implementation details

In this section, we describe the details of the model, and training recipes of our tracker. Code, models and training configurations will be made publicly available upon publication.

#### Detection transformer

Our detector transformer is based on deformable-DETR [X. Zhu et al. 2021] transformer with ResNet50 [K. He et al. 2016] backbone. We use a four feature-level deformable detection module [X. Zhu et al. 2021], 256 dimensional embeddings and 2048 feed-forward dimension size, 6 encoder layers and 6 decoder layers with 8 attention heads. No additional modifications to the transformer architecture were made.

#### Training details

All our models are initialised from an object detector [X. Zhu et al. 2021] that is pre-trained on the COCO dataset with additional `pos` queries (that encode perturbed ground-truth bounding box). Encoding perturbed ground-truth bounding boxes and passing them as `pos` queries during detection training not only boosts detection performance (as seen in Table 2.8), but reduces the number of pre-training epochs before convergence. Since this is detection-only training, we do not train with queries which are used over multiple frames (`both` and `id`).

**MOT17/20:** For MOT challenges, the model is first trained on simulated motion pairs of images from the CrowdHuman dataset for 50 epochs with backbone learning rate of $1e-5$, and encoder-decoder learning rate of $1e-4$, and we reduce the learning rate by a factor of 10 after the 40th epoch. Then, the model is finetuned for the particular downstream MOT dataset (e.g. MOT17) for an additional 20 epochs, reducing the learning rate again half-way through the finetuning.

**GMOT40:** For generalised MOT, we train the model on pairs of images from COCO with motion simulated by an affine transformation (as described in the main body of the paper) for 50 epochs with the same initial learning rate as above, decreasing the learning rate after 20th and 40th epoch.

Table 2.7: Additional model hyper-parameter values.

| Dataset | $\tau_{\text{conf}}$ | $\tau_{\text{agree}}$ |
|---|---|---|
| MOT17 | 0.75 | 0.1 |
| MOT20 | 0.65 | 0.05 |
| TAO-person | 0.80 | 0.2 |
| GMOT40 | 0.85 | 0.2 |
| TAO | 0.65 | 0.1 |

**TAO:** For TAO, we follow the same training procedure as for GMOT40, but additionally fine-tune the model on TAO training set for an additional 10 epochs with learning rate of $1e-5$ across all modules. For the TAO-person dataset, we use the model trained on MOT17, and tune the $\tau_{\text{conf}}$ parameter on the TAO-person training set.

### Tracking hyper-parameters

In this subsection we give the details on two hyper-parameters of *MQT* model: $\tau_{\text{conf}}$ and $\tau_{\text{agree}}$. $\tau_{\text{conf}}$ is a track-confirmation hyper-parameter. A track is only considered active if the similarity of the appearance query corresponding to the object $k$ at frame $i-1$ and $i$ is greater than $\tau_{\text{conf}}$. $\tau_{\text{agree}}$ is used in a version of our model where we leverage multiple queries in order to determine track quality. If the cosine distance between the appearance vectors corresponding to the pos and id queries refering to the same object $k$ (before aggregation) is smaller than a hyper-parameter $\tau_{\text{agree}}$, we do not consider additional hypothesis. On datasets where the model is fine-tuned, hyper-parameters are tuned via linear search on the held-out validation set. For datasets where it would be too expensive to fine-tune the model (e.g. TAO), we determine the optimal parameter by conducting a linear search with a pre-trained model. Specifically, we run a 5-fold cross-validation on a held-out validation set. The values are presented in Table 2.7.

### Appearance loss

For tracking purposes, we train the model with an appearance head on top of the transformer decoder. The purpose of the appearance head is to make two corresponding instances similar in the latent space, and at the same time push them away from all other instances of the same class (or indeed from any non-tracked objects). For this to be satisfied, the appearance head must remove the

location information from the raw embedding that is outputted from the decoder.

Consider object $q^k \in Q$ being the appearance encoding of the object $q$ at frame $k$, where $Q$ is a set of all outputs from the decoder for that frame. For simplicity of notation, let us denote a set of all *other* outputs from the decoder in the same frame as $Q_- = Q \setminus q$. We then compute the appearance loss for $q$ as

$$\mathcal{L}_q = -log\frac{\exp(q^{k+1} \cdot q^k)}{\sum_{j \in Q_-} \exp(q^{k+1} \cdot j^k)}$$

In this way, we can train the appearance head to associate related objects without any additional supervision.

Note, however, that the number of negative samples outweighs the positives. During training with `det` queries, the size of $Q_-$ gets large ($> 100$, depending on the number of `det` queries). To offset this, we down-weight the negative samples by a factor of 0.1.

Note that the appearance head replaces what would be a class-prediction head in the traditional object-detection model, and appearance loss replaces the classification loss in detection-transformer framework. Therefore, we need to assign it the matching costs for Hungarian matching algorithm and loss coefficients in order for the transformer to train and utilise the new information [Carion et al. 2020]. We cross validate these hyper-parameters on the MOT17 dataset, and train the final models with matching cost of 1 and loss coefficient equal to 2.

### 2.6.2 Tracking protocols

In this section we expand on the use of varying tracking protocols, and how they reflect on our model. Despite the intuitive simplicity of our model, the various intricacies of tracking benchmarks require different types of inputs to the decoder. Therefore, we describe various scenarios for each dataset. Visual illustration of the protocols can be seen in Figure 2.3.

#### MOT

Fundamentally, MOT tasks fall in one of two categories: private and public. In the *private* setting, the task of the model is to detect all possible objects of interest in each frame, and track them thus forming trajectories over multiple frames. In

the *public* setting, we are given detections for each frame. The sole task of the model is then to possibly refine and associate the detections with one another thus forming trajectories.

**Public:** In the public setting, the detections are given. In that case, we initialise the sequence with the given detections, passed to the transformer decoder as semantic `pos` queries. At the tracking step, we propagate semantic queries for all tracked objects, and add on the detections for that particular frame as additional `pos` queries.

**Private:** In the private setting, we initialise the tracking sequence with a number of static `det` queries. They are a set of learnt vectors that are fed into the decoder at each step, and don't change between frames (hence static). If the appearance output of any two `det` queries have a similarity greater than $\tau_{\text{conf}}$, we establish the two respective queries as known object tracks. The three semantic queries for each object track are then passed to the following frame in an auto-regressive manner, together with additional `det` tracks. This initialisation approach is sub-optimal compared to the class-specific models that output confidence score per track as class probabilities (though this is only applicable in class-based tracking by detection). However, empirically, we notice that the difference between the two methods is minor (see Table 5 (c) of the main paper).

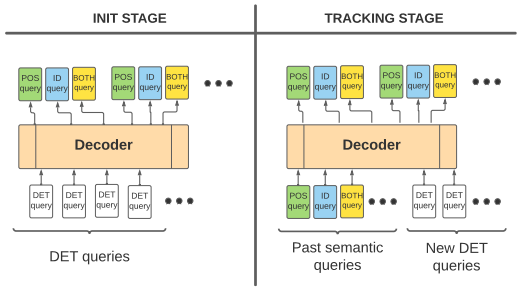
**Private & Public:** In a setting we denote as "Private & Public", we combine "private" and "public" setting. Namely at both initialisation and tracking stages feed in both `det` queries as well as the detections passed in as `pos` semantic queries.

**Private+:** In a setting we denote as "Private+", we follow the same procedure as in "Public", but replace the given detections with independently obtained detections from a state-of-the-art detector [Z. Ge et al. 2021].

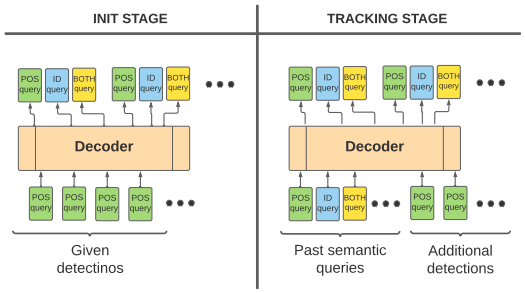
### TAO

We report the results on TAO in a "user initialised" setting with a standard "init first" approach [Dave et al. 2020]. For each object in TAO, the tracker is initialised using the fist frame an object appears in and it runs for the rest of the video. Similar to their experiments, we consider the object absent when the confirmation threshold falls under a certain value (0.65, cross validated on TAO training set).

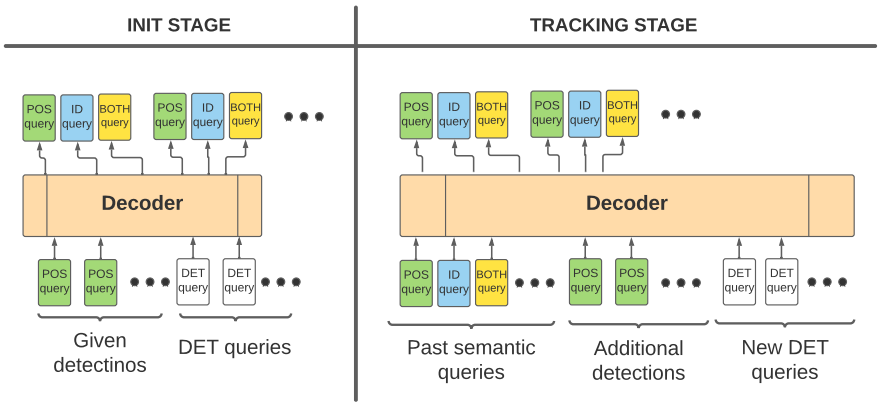
Figure 2.3: Various tracking settings are illustrated bellow. We show different semantic queries (`pos`, `id`, `both`) in colour (green, blue and yellow respetively), and static `det` queries are shown in white. Best seen in colour.




(a) Illustration of initialisation and tracking for *private* detection setting. `det` queries are a set of learnt vectors that act as spatial anchors and are used for detection at each step.




(b) Illustration of initialisation and tracking for *public* detection setting. Detections supplied with the sequences are passed to the model as `pos` queries.




(c) Illustration of initialisation and tracking for *private* & *public* detection setting.

Finally, the tracks are supplied with a class oracle in order to be able to report the "mAP" score, as in [Dave et al. 2020].

### 2.6.3 Additional ablation studies

In this section, we expand upon the ablation studies in the main body of the paper (Sec. 4.3). First, we verify the effectiveness of our semantic queries in a simpler object-detection scenario. Then we investigate the impact each query type has on tracking to further expand our numbers from the main body of the paper.

#### Object detection

For ablation purposes, we evaluate our model on COCO [Tsung-Yi Lin et al. 2014] to verify our performance on object detection. In Table 2.8 (a), we evaluate on

full-size COCO images, whilst in Tables 2.8 (b,c) we evaluate on COCO images *transformed* in order to train the tracker.

**Location queries:** Although the concept behind semantic queries is intuitive, we demonstrate their effectiveness on the validation split of COCO dataset. We first train the model for object detection. At training time we feed in perturbed ground truth bounding boxes as queries to the decoder, and during evaluation we use either encoder proposals generated by [X. Zhu et al. 2021] in their two-stage model, slightly perturbed bounding boxes, encoder-proposals as queries. For upper-bound comparison, we evaluate the model by feeding in the direct ground truth embeddings as queries as well. Table 2.8 (a) shows that the model can indeed use the position embedded in the queries to further refine their location. By training the model to refine object proposals, we can even improve upon the two-stage proposal model proposed by [X. Zhu et al. 2021] when using their encoder proposals. We see that detection becomes an easy task, therefore location-based queries are forced to learn how to retrieve important identifying information about the objects given their location.

**Object detection whilst tracking:** During tracking, we are still performing object detection: we might want to initialize new tracks at the initial stage (*init*), or "pick up" new ones whilst tracking (*tracking*). Therefore, we show the object detection performance of our semantic queries in a simulated tracking scenario. We simulate tracking data for training and evaluation from COCO as described in Sec. 3.2 of the main paper. During the evaluation, we evaluate our initial `pos` queries for object detection during the init stage (original image), and we show how well various semantic queries perform during the tracking stage (spatially augmented image, where ground truth is augmented as well). During the evaluation, each type of query is fed into the decoder separately (avoiding information leakage) and subsequently matched to the ground truth using the matching process described in [X. Zhu et al. 2021]. When using `det` queries, we follow the protocol in [X. Zhu et al. 2021] by using $N_{\text{object}} = 100$ queries. Results in Table 2.8 (b) show that training for joint tracking and detection impacts the model's performance on the detection-only task (which is expected, as the regression-head now learns to *predict* the location at the next frame). They also show that the model is able to successfully disentangle the semantic information from various queries – appear-

ance query knows far less about the object location compared to the ground-truth location one.

**Feature aggregation – object detection:** In the ablation study above, we look at each query independently, but in our final model, we are aggregating embedding information together. Therefore, we investigate four different aggregation mechanisms, still whilst in the setting of object detection. Now in the tracking stage, instead of computing the embedding for each query separately, we pass them through the encoder together, thus allowing the model to attend to both the appearance and the location of the object. Since for each known object, we now produce three common embeddings $V_t$ for $t \in \{\texttt{pos}, \texttt{id}, \texttt{both}\}$. We aggregate these embeddings in four different ways: 1) max-pooling, 2) average-pooling, 3) concatenate and pass through an MLP, or 4) pass through collaborative-gating unit [Yang Liu et al. 2019]. Collaborative gating marginally outperforms other aggregation methods. Full results can be found in Table 2.8 (c).

Table 2.8: Ablation study of semantic queries used for object detection, evaluated on COCO validation set.

| Models | Query | AP |
|---|---|---|
| Baseline dDETR | detection | 43.8 |
| Baseline dDETR | two-stage proposals | 46.2 |
| Ours | two-stage proposals | 49.8 |
| Ours | perturbed GT bbox | 71.6 |
| Ours | GT bbox | 75.1 |

(a) Comparison of various semantic queries (including ground truth (GT) for the upper bound) when models are trained purely for object detection on COCO. Model learns to effectively use the provided ground truth data, and refine perturbed boxes demonstrating that refinement can be done accurately given precise enough bounding boxes.

| Query | Stage | AP |
|---|---|---|
| GT bbox | init | 71.9 |
| `pos` | tracking | 69.1 |
| `id` | tracking | 37.4 |
| `both` | tracking | 62.0 |
| `det` | tracking | 42.8 |

(b) Comparison of various semantic queries trained on tracking data simulated from COCO, and evaluated on object detection. We show that model is capable of object detection even when trained for tracking, and that our method is successful in disentangling the semantic information into separate queries for the tracking step.

| Aggregation method | AP |
|---|---|
| max-pool | 54.7 |
| avg-pool | 61.1 |
| cat + MLP | 71.1 |
| collaborative gating | 71.5 |

(c) Comparison of different embedding-aggregation strategies for known objects at tracking stage. We show that learnable aggregation strategies have significant advantage over simple pooling strategies.

### Single- vs multi-query tracking

In Table 2.9, we further expand upon the the ablation study from the main body of the paper (Table 5 (a)), namely by looking at various permutations of queries. The output of the queries is then aggregated using collaborative-gating aggregation function [Yang Liu et al. 2019] as described in Section 3.1.5 of the paper.

Table 2.9: Further analysis of different query-combinations.

| det | id | pos | both | MOTA ↑ | IDF1 ↑ |
|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | ✓ | 68.3 | 66.1 |
| ✓ | | | | 55.9 | 54.7 |
| ✓ | ✓ | | | 57.4 | 58.2 |
| ✓ | | ✓ | | 61.1 | 57.4 |
| ✓ | | | ✓ | 62.9 | 63.5 |
| | ✓ | | | 52.7 | 54.1 |
| | ✓ | ✓ | | 59.4 | 58.6 |
| | ✓ | | ✓ | 58.2 | 57.0 |
| | ✓ | ✓ | ✓ | 65.1 | 64.8 |
| | | ✓ | | 56.1 | 55.8 |
| | | ✓ | ✓ | 61.6 | 64.1 |
| | | | ✓ | 56.9 | 56.2 |

We can see two broad trends. First, that including `det` queries during tracking significantly increases the performance of our model. Second, that `pos` and `both` queries tend to carry more relevant information for tracking purposes compared to `id` queries.

### Heuristic method for query aggregation

In the main paper (Table 6 (b)), we show the efficacy of various output aggregation schemes and compare them to, what we refer to as, the heuristic method. Since most tracking-by-detection systems have two distinct tasks (detection and data-association), we define the heuristic method as using the output corresponding to a particular query type for each task. In Table 2.10, we present all permutations of queries that in the end lead to the result presented in Table 6 (b) of the main paper, here shown in bold. Note that, unlike in Table 2.9, `det` queries are present in all permutations and we void the query aggregation scheme completely.

Firstly, we can see that auto-regressive information propagation is key for good performance as the performance of `det` queries is lower than, for example, auto-regressively propagated `det` queries by 8.3 MOTA points. Furthermore, our findings reinforce those by [E. Yu et al. 2021]. While they use a learned module to disentangle detection and re-identification information, our embeddings are disentangled by separate feed-forward networks (appearance head, bounding-box detection head), we find that using specialised embeddings for each step outperforms using joint embeddings – in our case by 0.9 MOTA points with no aggregation,

Table 2.10: Further analyses of the heuristic method in Table 6b of the main paper. Instead of aggregating the outputs of semantic queries, we use a specific one for each stage of tracking process. Note that `det` queries are present in all cases. The last row shows performance of detection-queries only (i.e. model without explicit auto-regressive information propagation).

| Detection | Data-association | MOTA ↑ | IDF1 ↑ |
|---|---|---|---|
| `pos` | `pos` | 59.4 | 54.9 |
| `pos` | `id` | **65.7** | **64.9** |
| `pos` | `both` | 63.8 | 61.3 |
| `id` | `pos` | 56.9 | 52.2 |
| `id` | `id` | 55.7 | 55.1 |
| `id` | `both` | 55.4 | 53.8 |
| `both` | `pos` | 58.8 | 53.4 |
| `both` | `id` | 64.6 | 64.3 |
| `both` | `both` | 64.8 | 64.6 |
| `det` | `det` | 53.1 | 50.4 |

and 3.8 MOTA points when a learned aggregation module is applied.

### 2.6.4 Further performance enhancements via offline tracking

In Table 1 of the main paper, we RelationTrack [E. Yu et al. 2021] stands out compared to all other methods. Indeed, they report their baseline model, which shares the same backbone as MQT, achieves 68 MOTA points [1] (compared to 56 for MQT). A potential reason for such improvement is their tracking protocol, which consists of offline refinement of the initially proposed tracklets and further post-processing including trajectory-filling strategies discussed in [S. Han et al. 2020].

As our model doesn't have a separate data-association stage we instead attempt to imitate this two-step procedure by doing two separate tracking passes on the video: one from beginning to the end, and another from the end to beginning. If the track bounding boxes overlap by over 50% of the total area, we consider the tracks to be true positives. We refer to this as back-to-front track confirmation (B2F track). Additionally, we attempt to use the Hungarian algorithm with a matching threshold of 0.4 to match tracks from a beginning-to-end pass with the appearance

[1] There is no specification on the evaluation protocol used; we assume standard 7-way split cross-validation.

Table 2.11: Comparing the effects of various offline post-processing enhancements on our model on a standard 7-way split cross-validation for MOT17. Our model uses detections from [Z. Ge et al. 2021] in "Private+" setting.

| Method | B2F track | B2F id | TF | MOTA | IDF1 |
|---|---|---|---|---|---|
| RelationTrack–baseline [E. Yu et al. 2021] | | | | 68.5 | 73.3 |
| Ours | | | | 56.9 | 56.2 |
| Ours | ✓ | | | 60.1 | 59.8 |
| Ours | ✓ | ✓ | | 62.7 | 62.0 |
| Ours | ✓ | ✓ | ✓ | 63.4 | 64.2 |
| Ours | | | ✓ | 58.0 | 58.6 |

(a) Effects of the performance-enhancing post-processing methods on our baseline model using only `both` track queries.

| Method | B2F track | B2F id | TC | MOTA | IDF1 |
|---|---|---|---|---|---|
| RelationTrack [E. Yu et al. 2021] | | | | 70.2 | **75.3** |
| Ours | | | | 68.3 | 66.1 |
| Ours | ✓ | | | 69.7 | 69.8 |
| Ours | ✓ | ✓ | | 71.6 | 71.1 |
| Ours | ✓ | ✓ | ✓ | **72.1** | 73.3 |
| Ours | | | ✓ | 68.9 | 72.1 |

(b) Effects of the performance-enhancing post-processing methods on our best model. Note that semantic queries and our multi-query tracking protocol significantly reduce the need for an additional offline post-processing.

vectors of an end-to-beginning pass and vice-versa and repeat the overlap process from above. We refer to this as back-to-front appearance confirmation (B2F id). Lastly, we adopt the trajectory-filling (tf) strategy as in [S. Han et al. 2020; E. Yu et al. 2021] on final predicted tracks.

### 2.6.5 Comparison with similar methods

Several related works concurrently attempt to adapt the transformer architecture to the MOT problem. Most of these works differ from *MQT* either in the tracking paradigm, or in architectural details. Bellow, we outline differences to the most similar work. The reader should note however, that *none* of these concurrent works can be generalised to the class agnostic GMOT or TAO tasks as they use tracking-by-detection approaches.

**TransCenter** [Y. Xu et al. 2021] shares architectural similarities with *MQT* (in that they use a deformable transformer and semantically separable queries), but the finer details and their tracking paradigm are fundamentally different. On an architectural level, they utilise query leaning networks to separate queries from the encoder representation. Furthermore, they utilise *two separate* decoder modules, one for each query type. We feed all the queries concurrently in the single decoder. More importantly, their tracking paradigm is fundamentally different as they compute *dense* queries for location and tracking displacement (as opposed to appearance like us). They show that given dense detection and tracking memory, tracking can emerge from these inputs alone. To this end, they design and combine custom decoder modules in order to aid in matching the model outputs

over time. Their paradigm renders the reID module unnecessary, but it comes at the expense of higher model complexity. While clearly innovative and effective, their approach performs better than *MQT* on one metric (MOTA 71.9 vs 65.4) but worse on others (IDF1 62.3 vs 63.4, or ID switches 4626 vs 1104).

**MOTR** [F. Zeng et al. 2022] shares the transformer backbone with *MQT*, and they also leverage the power of semantically decoupled embeddings. Unlike us, however, their embeddings are decoupled via "global context disentangling unit" from the final layer output of a backbone CNN. We find that we can do it implicitly in the transformer decoder.

**Trackformer** [Meinhardt et al. 2021] is most similar in spirit to our *MQT* method. Their detection queries are analogous to our static `det` queries, and their tracking queries are analogous to our `both` queries. In the ablation studies (Tbl 5 (a); Tbl 3 in supp. material) we show the effectiveness of using multiple (and semantically decoupled) queries over the single tracking query paradigm of [Meinhardt et al. 2021].

### Running speed comparison

At the time of writing, only [Meinhardt et al. 2021] code was available for a direct comparison. In our experiments, we find that pre-training *MQT* on the CrowdHuman dataset takes 8 days using 6 V100 GPUs, which is comparable to the original implementation of [Meinhardt et al. 2021] on similar hardware.

# Chapter 3

# Text-Conditioned Resampler For Long Form Video Understanding

*The paper has been accepted at ECCV 2024.*

This chapter concerns itself with a technical aspect of story understanding, namely how to process exceedingly longer videos whilst maintaining the ability of the model to reason about them on the higher level. A key aspect of this work is the fact that such sampling is done in natural language and as such is relatively unrestricted in semantic scope.

# Text-Conditioned Resampler For Long Form Video Understanding

Bruno Korbar[1,2] Yongqin Xian[2] Alessio Tonioni[2]

Andrew Zisserman[1,3] Federico Tombari[2,4]

[1]Visual Geometry Group, University of Oxford

[2]Google Zurich

[3]Google Deepmind [4]TU Munich

March 29, 2026

**Abstract**

In this paper we present a text-conditioned video resampler (TCR) module that uses a pre-trained and frozen visual encoder and large language model (LLM) to process long video sequences for a task. TCR localises relevant visual features from the video given a text condition and provides them to a LLM to generate a text response. Due to its lightweight design and use of cross-attention, TCR can process more than 100 frames at a time with plain attention and without optimised implementations. We make the following contributions: (i) we design a transformer-based sampling architecture that can process long videos conditioned on a task, together with a training method that enables it to bridge pre-trained visual and language models; (ii) we identify tasks that could benefit from longer video perception; and (iii) we empirically validate its efficacy on a wide variety of evaluation tasks including NextQA, EgoSchema, and the EGO4D-LTA challenge.

## 3.1 Introduction

The development of visual-language models (VLMs) advanced exponentially in the past few years: new models pre-trained with increasingly larger scale, in terms of the number of parameters and size of the training set, continue pushing forward the

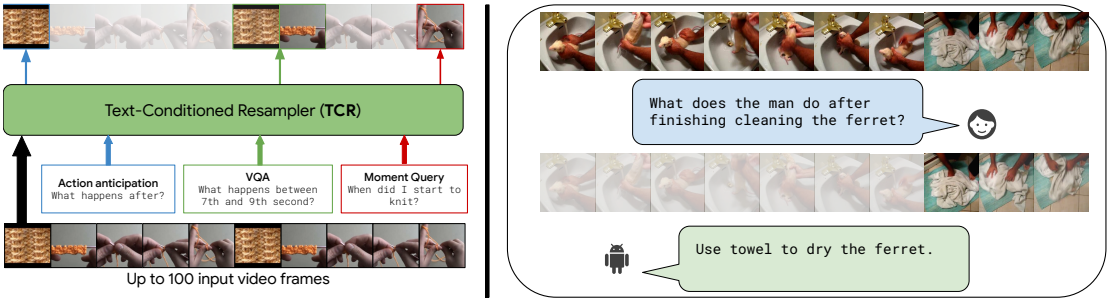



Figure 3.1: TCR resamples visual features that are relevant for the downstream tasks before passing them to the LLM. A qualitative example can be seen on the right.

state of the art on multiple tasks every couple of months. These models often have the ability to reason about the relationships of objects in their environment through natural language, often in an interactive fashion. This capability is appealing for multiple video applications. For example, it would be helpful for a model to be able to answer questions about a video: "Does this recipe use eggs?", "what does he do after he removes the tire?", etc. It is also appealing for users of augmented-reality devices: for example to be able to answer "where did I leave my phone?". Unfortunately, the computational requirements of such models made them impractical for use in video applications as the memory requirement rises quadratically with the input size. Furthermore, to our knowledge, a large-enough source of even loosely labelled video data for training such a model from scratch does not readily exist.

That is why we are specifically interested in a subset of these models that are not trained from scratch, but rather 'bridge' pre-trained models via different types of 'visual-to-language adapter modules' [J. Li et al. 2023b; Alayrac et al. 2022; Sevilla-Lara et al. 2021]. The advantages of this approach, as opposed to training the model from scratch, are numerous: Only a small number of parameters are trained, which makes the memory footprint smaller; it allows us to utilise the capabilities of large visual backbones without overfitting to the downstream task; as well as to leverage the vast amount of knowledge stored in the LLM without suffering common limitations of smaller-scale fine-tuning such as catastrophic forgetting. Only a few of these models are trained on videos [Alayrac et al. 2022; Kuo et al. 2023; S. Yu et al. 2023], and these can usually ingest only a small number of frames – typically anywhere between 4 to 32. Allowing a large number of video frames to interact with text is demonstrably beneficial [Sevilla-Lara et al. 2021; Mangalam

et al. 2023] in visual models, thus, a relatively simple way of increasing the model performance is to increase the number of frames the model sees.

In this paper we present a *Text-Conditioned Resampler* (TCR), an architecture and pre-training method that tackles all of the challenges mentioned above: it is a reasonably lightweight, low-dimensional adapter which acts as an information bottleneck between visual and language models. As shown in Figure 3.1 (left), it is able to process over a 100 (and up to 180) frames at a time, selecting the most relevant frame features to pass to the LLM based on the "conditioning" text. TCR allows us to focus on analysing videos with longer temporal span, and identify gains that could be made on longer videos. Right side of Figure 3.1 illustrates an application of our model. This new method allows us to analyse aspects of video datasets we've never been able to before. Specifically, we were able to look at how many frames it takes for a VLM to solve a task, and to determine if increasing the temporal span of perceived video actually brings benefits in terms of performance. We found that increasing temporal span does improve results on the moment-queries EGO4D challenge, and allows us to set the state-of-the-art (SOTA) on long-video question answering on the validation sets of EgoSchema dataset [Mangalam et al. 2023] and EGO4D long-term forecasting challenges, as well as on the temporally-sensitive NextQA dataset [J. Xiao et al. 2021].

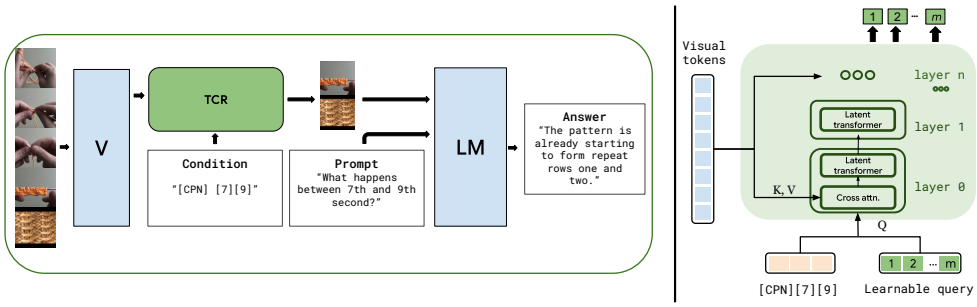



Figure 3.2: Left: overview of how TCR integrates in a VLM in order to process long videos. A long (30-120 frames) sequence from a visual encoder (V) is resampled to a fixed-length sequence fed to a language model. `[CPN]` indicates special token for captioning; `[7][9]` is a representation of tokenised time steps. Right: details of the TCR module. Elements in blue are kept frozen. Best viewed in colour.

## 3.2 Text-Conditioned Resampler (TCR)

In the following section we describe the model and the training procedures used for training a video-specific VLM able to handle very long video sequences.

### 3.2.1 Model

At a high level, the input to the TCR consists of video frames processed by a visual encoder and embedded text tokens. It outputs a fixed-length sequence of embeddings that, together with a text prompt, is consumed by a language model. The text specifies (conditions) the task, and the TCR selects different visual features according to the task and transforms them to be suitable for input to the language model. Finally, the language model generates the text response to the specified task. Architecture overview is given in Figure 3.2 on the left.

**Overview:** The visual inputs consist of RGB frames of the video that are ingested by a pre-trained frozen ViT-g [Q. Sun et al. 2023] model to obtain visual embeddings. Temporal encodings are added to them. The conditioning text tokens are prefixed with a learnable special token specifying the task the model is trying to solve and concatenated with a set of learnable query vectors. The queries and text interact with each other through self-attention layers, and interact with the frozen visual features through cross-attention layers (inserted every other transformer block). Output query vectors are then concatenated with an optional text prompt, and passed through a frozen Flan-T5 language model [H. W. Chung et al. 2022]. The TCR module is illustrated in Figure 3.2 on the right. We treat it as a plug-in replacement for the Q-former in the BLIP2 [J. Li et al. 2023b] architecture to enable handling of very long frame sequences as input.

The key design choices are: (i) the interaction of the query sequence with the visual features is only through cross-attention. This enables the TCR to ingest very long sequences (as it is not limited by the quadratic complexity of vanilla self-attention); and (ii) the output is a fixed length set (the transformed query vectors), so that the input to the language model is only a small number of tokens, irrespective of the length of the video sequence. Following these design principles we are able to significantly reduce the number of input tokens that the LLM needs to process with obvious gains in terms of inference time and memory requirements compared to full self-attention over all frame tokens.

**How does the TCR differ from the Flamingo Resampler and Q-former?** These design decisions build on the architectural innovations of the Perceiver resampler in Flamingo [Alayrac et al. 2022] and the Q-former in BLIP-2 [J. Li et al.

2023b]. However, there are a number of differences: (i) While Q-former is trained on images, TCR is optimised for video from the ground up – all training stages are done on videos. This is important as the TCR must learn to sample visual features from video frames conditioned on the task. (ii) TCR uses lower dimensional features than either Q-former or Perceiver Resampler (512 vs 768 vs 1536) and an overall smaller number of parameters (69M vs 188M). This is important as it allows us to process far longer video sequences. (iii) While TCR cross-attends visual features to text embeddings and learnable queries, Perceiver Resampler concatenates visual-embeddings and queries in a key-value pair, which makes the computation more expensive as it computes cross-attention and self-attention in a single pass. We keep the operations separate (i.e. first cross-attending text-query sequence with the video, and then self-attending the text-query sequence). This reduces per-layer computational requirements allowing us to increase video sequence length. These differences lead to a novel capability of processing many more frames at once, which subsequently leads to superior performance on downstream tasks.

**Conditioning sequence construction:** Most tasks can be represented as a basic Question and Answer (QA) pair. Inspired by multi-task language models [Raffel et al. 2020], we adopt a generic `[ST][task prompt][learnable query]` input structure (where `[ST]` is a task-specific special token, `[task prompt]` is, for example, a question in a QA, and `[learnable queries]` are passed on to the LLM). We prefix a special task token (`[CPN]`, `[TRG]`, `[QA]`, `[STG]`) for captioning, temporal grounding, question-answering, and spatio-temporal grounding respectively) to the task prompt, depending on what task the model is solving. Figure 3.2 shows an example wih the `[CPN]` task-specific special token. Since, in principle, *all* tasks can be formulated as QA (and would be specified in the model as `[QA][question text]`), why are special tokens used? We found that using tokens improves overall performance while making the model easier to train and reducing the sequence length required for conditioning the sampler (as opposed to spelling out the task in text).

### 3.2.2 Training

Recent works have shown that contrastive learning yields visual representations for video frames that perform better in discriminative tasks than training in a

Table 3.1: Effect of initialisation and pre-training stages on NextQA question answering and NLQ task. For NextQA, we use shortened fine-tuning procedure (see Section 3.3.2) and vary the checkpoints used. For NLQ, we evaluate on TCR w/LLM.

| Init | Pre-training (i) | (ii) | (iii) | NextQA Acc ↑ | NLQ MR@1 ↑ |
|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | ✓ | **66.1** | **11.42** |
| ✓ | ✗ | ✗ | ✗ | 52.1 | 7.88 |
| ✗ | ✓ | ✓ | ✓ | 63.3 | 9.41 |
| ✓ | ✓ | ✗ | ✗ | 64.1 | 8.94 |
| ✓ | ✓ | ✓ | ✗ | 65.6 | 9.37 |
| ✓ | ✗ | ✓ | ✗ | 63.4 | 8.91 |
| ✓ | ✓ | ✗ | ✓ | 64.2 | 8.13 |

purely generative fashion [K. P. Yu n.d.; Kuo et al. 2023]. Training models with a generative loss, however, seems to be crucial for developing reasoning regarding temporal grounding of unconstrained videos as well as the semantic relationship between text structure and video [A. Yang et al. 2023; Alayrac et al. 2022]. Hence, we separate our training in three distinct stages: (i) initialisation, where we train TCR without the LLM; (ii) pre-training, where we train TCR in conjunction with the LLM; and later, (iii) a task-specific fine-tuning. Note that the only thing we're training is the TCR module – the visual encoder and LLM remain frozen throughout. Initialisation and pre-training stages are done on the YTT-1B dataset [A. Yang et al. 2023]. Videos in this dataset are annotated by the transcribed speech sentences and their corresponding timestamps that are either user-generated or automatically generated via automatic-speech recognition. Speech in such videos is rarely visually grounded [Ko et al. 2022; K. Han et al. 2020], however, because our model can see the video sequence surrounding the annotated segment, it is well suited to implicitly learn the temporal grounding. We describe training stages below.

**Initialisation (without LLM):** To initialise our model, we follow BLIP2 [J. Li et al. 2023b]'s *image-text contrastive* and *image-text matching* objectives. Contrastive objective maximises mutual information between TCR text output, and learnable queries which are cross-attended with a video. Text and learnable queries are passed together to TCR. Their mutual attentions masked in such a way that text only attends to itself, while the learnable queries are cross-attended to the video frames and then self-attended to themselves. We compute the average of text

queries to get a text representation $t$, and compare it pairwise with all learnable queries. Query with maximum similarity to $t$ is denoted as $q$. We then align the representations $t$ and $q$ by contrasting each positive pair with in-batch negative pairs. At this stage TCR is *not* text conditioned. Image-text matching objective (video-text matching in our case) primes the model for text-conditioning. Both learnable queries and text are passed through TCR together, without attention masking. A binary classifier predicting whether the video and text are matching or not is applied to each of the learnable queries and predictions are averaged to obtain a final matching score. The negatives are sampled in-batch, following [J. Li et al. 2023b].

We skip the *generative* training step of [J. Li et al. 2023b], as our model is neither designed nor initialised from a language model, and we found no measurable benefit from this training stage. The reader is referred to the original paper [J. Li et al. 2023b] for in-depth description of attention-masks and losses used during each of the objectives.

**Pre-training (with LLM):** The goal of pre-training is twofold: first, to semantically and temporally align TCR's output with the expected input of the LLM, and second to train TCR's self-attention layer to attend to specific task-specifying special tokens and text conditioning tokens. We do this by training it on three tasks. (i) given an untrimmed video and annotated sentence, we ask it to retrieve *when* the sentence occurred; (ii) given the untrimmed video and a timestep, we ask the model to fully caption that particular segment; (iii) given the untrimmed video and a text sequence corrupted in multiple ways, we ask it to correct the sequence. All tasks are supervised by applying the generative loss on the outputs of the LLM. The examples of these tasks on an example from YTT dataset can be seen in the supplementary material. The effects of these training stages can be seen in Table 3.1.

**Fine-tuning:** After these two stages, TCR achieves competitive results on downstream tasks while still being a generalist model. However, as our pre-training dataset is comprised mostly of low- and mid-quality videos with noisy automatic annotations, we observe significant improvements through fine-tuning for a specific task. The goal of fine-tuning is to align the TCR with the domain of the downstream task in question. Only the TCR module and its vocabulary are fine-tuned,

while the visual encoder and the LLM are kept frozen. Fine-tuning is performed on each of the downstream datasets and is described in the results section for each dataset, while hyperparameters and ablation of the performance with or without fine-tuning are given in the supplementary.

### 3.2.3 Model details

**Video sequence construction:** We extract visual representations ($14 \times 14$ patches from frames with $224^2$ resolution) using ViT-g [Q. Sun et al. 2023], and add temporal embeddings. In order to reduce memory consumption, for every other frame we drop random 50% of its patches. Recent work [T. Han et al. 2022b; Tong et al. 2022] has shown no significant loss in performance when random patches have been dropped.

**LLM sequence construction:** We follow BLIP2 in the way we construct the input sequence [J. Li et al. 2023b]. We concatenate the output of the TCR module together with a `<BOS>` (beginning of sentence) token and the instruction context tokens (for example question in VQA, or previous action sequence together with instruction for EGO4D action prediction).

**TCR architecture details:** TCR is based on a transformer-decoder module [Vaswani et al. 2017a], consisting of 4 transformer blocks with 8 attention heads and hidden dimension equal to 512. Blocks 0 and 2 contain cross-attention layers. For each task we use 128 512-dimensional queries. These choices were tuned based on the downstream performance on NextQA validation set and then kept fixed.

## 3.3 Experiments

In the following section, we conduct a set of experiments with a baseline VLM and TCR in order to determine which tasks benefit from having access to longer or denser video sequences, and compare the results to the SOTA. Specifically, we analyse the datasets in section 3.3.1. We compare the results to the state of the art in sections 3.3.2, 3.3.3 and 3.3.4, and we present ablation of model decisions in section 3.3.5. Qualitative results can be seen in Figure 3.4.

**Datasets:** We evaluate the following datasets: Kinetics400 [Kay et al. 2017] containing around 260k 10s videos with human-action labels and Countix, a subset of Kinetics where actions are annotated with the number of repeats (e.g. how many time a push up is repeated) [Dwibedi et al. 2020]. MSR-VTT [Jun Xu et al. 2016], a large scale video captioning dataset. NextQA, a manually annotated video-question-answering dataset where the model is asked to answer questions regarding temporal actions in a multiple-choice fashion [J. Xiao et al. 2021] from (on average) 44s long videos. Finally, since egocentric videos are a new frontier in effective long-term video understanding, we evaluate on two diverse challenges from EGO4D [Grauman et al. 2022]. EGO4D videos are often minutes long, containing both fine-grained actions as well as long-term interactions [Grauman et al. 2022].

**Baseline:** We use a fixed BLIP2 [J. Li et al. 2023b] VLM as a baseline throughout our experiments. BLIP2 is not trained on videos, but it has been shown that it can be adapted to videos [K. P. Yu n.d.; Hang Zhang et al. 2023a] and we follow [K. P. Yu n.d.] to do so. BLIP2 can only process up to 8 frames at a time, so for the task where it can be done, we average predictions over multiple 8-frame video clips extracted at 1fps (noted as 'BLIP2(Avg.)'). These aggregation methods, however, can introduce unwanted noise [Sevilla-Lara et al. 2021; Korbar et al. 2019].

**Modelling longer sequences:** The design of TCR allows a VLM to "see" more frames than ever before. Therefore, we also present results using TCR which uses the same visual encoder and LLM as BLIP2 but is able to process *all* the frames at once. With TCR, each video can be processed in a single forward pass thus eliminating the effects of subsampling or averaging.

### 3.3.1 Video task analysis

Since videos are a highly redundant data-source, one has to ask how many frames, and at what sampling rate, does the model actually *need* to see to achieve good performance. For example, it has been observed that humans solve QA tasks with 8% higher accuracy when videos are sampled at 25fps as opposed to sampling them at 1fps [Mangalam et al. 2023]. In this section, we analyse the results on six common video-understanding tasks with respect to the number of frames consumed by the model. We look at the results from a high-level perspective, in order to determine which tasks will require higher number of frames for a model to solve.

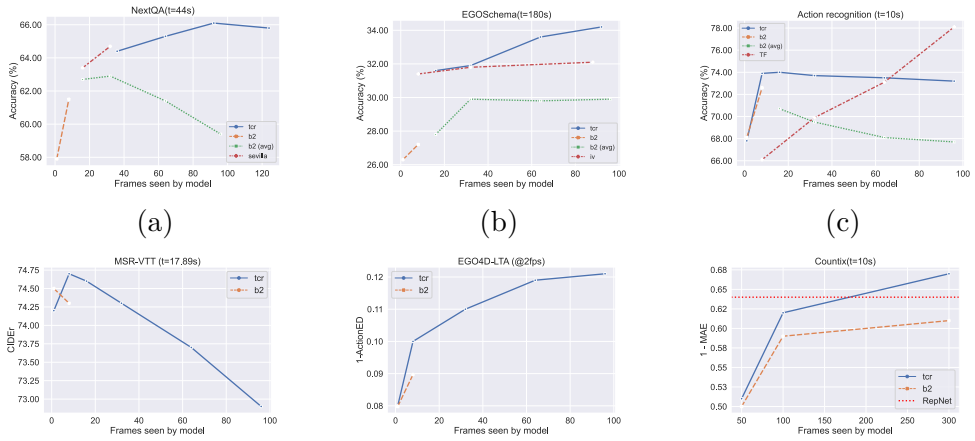



Figure 3.3: Performance vs number of frames utilised by the models on various different tasks. $t$ denotes average length of the video in the dataset. 'tcr'=Ours, 'iv'=IntenVideo [Yi Wang et al. 2022], 'TF'=TimesFormer [Bertasius et al. 2021], 'b2'=BLIP2 [J. Li et al. 2023b], 'sevilla'= [S. Yu et al. 2023], RepNet= [Dwibedi et al. 2020]

Overview of the results can be seen in Figure 3.3. BLIP2 and TCR models are initialised from the same checkpoint, and then finetuned on the downstream task using the target number of frames as an input. We describe evaluation procedure for each task in more detail in their respective sections.

Intuitively, question-answering is one of the tasks where longer spans and better understanding of temporal dependencies would be of utmost importance. On the NextQA dataset, where questions contain temporal aspects (3.3a), seeing the span of an entire video seems to be crucial for performance. There is a clear peak at about 2fps, and a sharp decline past 1fps. This means that it is important to observe most of the frames, but frame density is not strictly required.

Curiously, although long video sequences at higher frame-density are required for humans to solve problems in EgoSchema dataset (3.3b), most models' performance actually peak or plateaus at significatly smaller number of frames [Mangalam et al. 2023]. We argue that this is because they have not been trained with long input length, and subsequently fail to capture semantic interdependencies within the video. TCR's performance increases with the number of frames, but plateaus when more sparsity in frame patches is needed to keep memory consumption down (see Table 3.3 for more details). We believe that being able to *see* the span of entire video with more density (i.e. at sampling rates greater than 1fps) could further increase performance on this benchmark.

Human action recognition (3.3c) and short-video captioning (3.3d) are commonly used video-understanding benchmarks, however, we found that they do not require many frames to achieve strong performance with a VLM. On action recognition, specialised models such as [Bertasius et al. 2021] scale better with higher frame density, however, which can be attributed to their to inherently learned sampling [Korbar et al. 2019]. This intuition can be corroborated by a slight performance increase when using TCR module with BLIP2. Future prediction tasks (3.3e) on egocentric videos often span a long temporal range. We found that increasing the number of frames, hence covering larger video spans, helps in reducing the overall error. This is not unexpected, as action sequences in EGO4D tend to be repetitive, so the longer sequence of actions allows the model to recognise the action pattern from more samples. Finally, counting (3.3f) is an example of the task where frame density matters, and performance drops when fewer frames are utilised from a fixed length video. It also requires specialised architecture to solve it [Dwibedi et al. 2020], and LLMs are traditionally disadvantaged in tasks that require numerical reasoning. For action classification and counting problems, we only use visual and aggregator parts of the VLM as described in the supplementary.

To conclude, although the tasks that require the models to be able to reason over many frames either in span or density are fairly limited, they do exist. Below, we show how a module such as TCR that allows us to 'see' more frames in context could be beneficial to overall performance on these tasks.

### 3.3.2 Evaluation on video question-answering

We first evaluate our model on question-answering benchmarks, our prior experiment shows a clear benefit when more frames are observed. We use the NextQA validation set to compare our model to the current state-of-the-art model which is also based on BLIP2 architecture [S. Yu et al. 2023]. We follow fine-tuning and evaluation protocol from [J. Li et al. 2023b], with hyperparameters outlined in the supplementary.

**Input design:** Video is subsampled to a target numbers of frames (92 frames at approximately 2fps for the final model), and temporal embeddings are computed accordingly. Conditioning text is formed as "`[QA] Question`" where `[QA]` is learned special token reserved for VQA tasks. During fine-tuning, the prompt to

the LLM is formed following [S. Yu et al. 2023] as: "`[vis. features][question][options] Considering information in frames, select the correct answer`".

**Evaluation procedure:** During inference, we restrict generation to the answer vocabulary (i.e. "Option A", "Option B", ...), and select the most probable answer.

**Comparison to SOTA:** Results can be found in Table 3.2. Our model outperforms BLIP2 which demonstrates that the TCR module is successful in selecting the relevant frames, and also indicates the need for temporal modelling of this particular task. While like us, the SeViLA model is based on BLIP2, they train one model to sample relevant keyframes, and a separate model to solve the task from the sampled keyframes, effectively doubling the number of trainable parameters. In contrast, TCR requires only a single forward pass during training to both sample the features and solve the task. Our model outperforms SeViLA in overall accuracy (setting the new SOTA), hence showing that number of observed frames makes up for lack of trainable parameters.

### 3.3.3 Evaluation on long-form VQA

EgoSchema is a long-form VQA dataset sampled from EGO4D containing 5000 human curated multiple choice question answer pairs, spanning over 250 hours of real video data. Each question requires the model to select one out of 5 possible answers and is accompanied by a three-minute-long video clip [Mangalam et al. 2023]. Input and evaluation designs are the same as they are for NextQA.

**Comparison to SOTA:** Results can be seen in Table 3.3. Our model outperforms both the SOTA models (where inference was done over multiple forward passes and prediction was averaged) and our re-implementation of BLIP2 (with both

Table 3.2: Comparison to SOTA on NextQA dataset. Results are split into non-balanced 'causal' (C), 'temporal' (T) and 'descriptive' (D) questions. The overall accuracy in the last column to the right is balanced across the entire dataset, rather than across the categories. '*' denotes re-implementation by [Sevilla-Lara et al. 2021]

| Model | train params | accC ↑ | accT ↑ | accD ↑ | acc ↑ |
|---|---|---|---|---|---|
| SeViLA [S. Yu et al. 2023] | 346M | 73.4 | 68.8 | **83.5** | 73.4 |
| HiTeA [Ye et al. 2023] | / | 62.4 | 58.3 | 75.6 | 63.1 |
| BLIP2 [J. Li et al. 2023b] | 188M | 64.9 | 59.7 | 77.8 | 63.5 |
| BLIP2* [J. Li et al. 2023b; Sevilla-Lara et al. 2021] | 188M | 72.9 | 65.2 | 80.1 | 70.1 |
| Ours | 76M | **73.5** | **69.8** | 82.2 | **73.5** |

Table 3.3: Comparison to SOTA and human performance on EgoSchema split of EGO4D. × denotes multiple forward passes were used. * denotes higher proportion of patches was dropped.

| Method | Observed frames | QA acc (%) ↑ |
|---|---|---|
| InternVideo [G. Chen et al. 2022] | 8×11 | 32.1 |
| BLIP2 [J. Li et al. 2023b] | 8 | 27.2 |
| BLIP2 [J. Li et al. 2023b] | 8×12 | 29.9 |
| TCR (ours) | 92 | 34.2 |
| TCR (ours) | 92×2 | 34.5 |
| TCR (ours) | 184* | **35.1** |
| Human | 180 | 67.2 |

subsampled frames and iterative inference approach). Similar to [Mangalam et al. 2023], we observe relative saturation of performance with increasing the number of frames.

### 3.3.4 Evaluation on EGO4D challenges

**Long-term action anticipation (LTA):**

The goal of the LTA challenge is to predict a sequence of twenty actions in order of appearance from an input video. The last observed action and action boundaries are given as well. The current state-of-the-art method relies solely on the power of large-language models in order to predict the sequence of future actions [D. Huang et al. 2023]. Our model adapts this idea but leverages the ability of TCR to process increasingly longer videos in order to achieve superior results. We compare our model to the SOTA, as well as to fine-tuned BLIP2 using 8 frames as video input. We note that our model outperforms BLIP2 by a significant margin, clearly showing the benefits of being able to observe denser video sequences for this task. Results can be seen in Table 3.4.

**Input design:** We construct input for fine-tuning and evaluation in the following fashion: video is subsampled uniformly to a target number of frames (8 for BLIP2 with Q-former, and 96 for BLIP2 with TCR) and temporal embeddings denoting the frame timestamp are added to them. If the "Before" video is sampled, we only sample from a video clip before the last observed action, while "Whole" means we sample from the entire input video clip. The text prompts are designed as:

```
1    Complete an action sequence,
2    an action is one (verb, noun) pair.
```

```
A complete sequence consists of 28 actions.
Actions: (noun_1, verb_1) (verb_2, ...
```

and for the conditioning prompt we use:

```
[LTA][start_1](noun_1, verb_1),[start_2](noun_2, verb_2) ...
```

where `(noun, verb)` is an action pair, `[LTA]` is a learned special token, and `[start k]` is a tokenised start time of $k$-th action.

**Evaluation procedure:** Our evaluation procedure follows closely those of [D. Huang et al. 2023]. The model is fine-tuned to output comma-separated action pairs following the prompt formatting. During the evaluation, we softmax predictions over the reduced vocabulary of the label space for the LTA task. If both nouns and verbs fall into their respective label space, we append them to our prediction. For predictions with less than 20 action pairs, we pad it with the last action. Models denoted with (*) are sampled in an iterative fashion.

**Comparison to SOTA:** Table 3.4 shows the comparison to the state-of-the-art on long-term action prediction. Note that SOTA [D. Huang et al. 2023] uses only language model to predict the future actions. We demonstrate that being able to perceive frames after the indicated timestep (which are given) helps for future action prediction, but we outperform sota even without seeing them. Finally, we find that iterative evaluation (i.e. asking the model to predict action by action, as opposed to the whole set of 20 actions, increases performance even further.

**Moment queries (MQ):**

The MQ task is similar to temporal action localisation or moment retrieval tasks. Given a textual description of an action, the goal is to localise all possible instances of it in the given video clip. Results can be seen in the Table 3.5.

**Input design:** Video is subsampled uniformly to the target number of frames, and temporal embeddings denoting the frame timestamps are added to it. The

conditioning prompt is formed as "`[TRG] action query string`" where `[TRG]` indicates the special token for temporal grounding and `action query string` denotes the name of the action label parsed as a string. The language model is prompted by the following string

> Return a sequence of frame timestamps
> where <action name> is happening. The
> timestamps range from 1 to 1000.

**Evaluation procedure:** We softmax the model predictions from a reduced vocabulary of integers from 1 to 1000 (temporal coordinates are quantised similarly to [T. Chen et al. 2021]) and aggregate them.

**Comparison to SOTA:** Results in Table 3.5 show that despite the disadvantage of solving a discriminative task in a generative way, our model still performs admirably (-2.4 MAP) when compared to the state-of-the-art. In the supplementary material, we present an additional evaluation procedure (using direct classification) which can yield even better performance (+1.25 MAP).

### 3.3.5 Model design decisions

In the following section we investigate our model choices and seek to explain their impact on the performance of our model. All experiments were done on the validation set of NextQA dataset and results can be seen in Table 3.6. Note that we fine-tune the model on a shorter training schedule which yields lower results, but

Table 3.4: Comparison of various models on the validation set of *EGO4D LTA challenge*. Edit distance is reported and the lower the score the better. The "Video" column indicates whether the whole video was observed (given) or just the video clip before the last action. Models denoted with '*' are sampled iteratively.

| Method | Video | VerbED ↓ | NounED ↓ | ActionED ↓ |
|---|---|---|---|---|
| PALM* [D. Huang et al. 2023] | No | 0.7165 | 0.6767 | 0.8934 |
| BLIP2 [J. Li et al. 2023b] | Before | 0.7512 | 0.6873 | 0.9103 |
| BLIP2 [J. Li et al. 2023b] | Whole | 0.7500 | 0.6799 | 0.9086 |
| Ours | Before | 0.7009 | 0.6472 | 0.8792 |
| Ours | Whole | 0.6763 | 0.6180 | 0.8522 |
| Ours* | Whole | **0.6585** | **0.6171** | **0.8482** |

Table 3.5: Comparison to the state of the art on the validation set of *Ego4D Moment Query Challenge*.

| Method | Avg. mAP ↑ | R@1, tIoU=0.5 ↑ |
|---|---|---|
| Intern Video [G. Chen et al. 2022] | 23.59 | 41.13 |
| ASL [J. Shao et al. 2023] | **27.85** | **46.98** |
| Ours (96f) | 24.51 | 42.99 |
| Ours (192f) | 25.45 | 43.72 |

allows for a quicker turnaround. We keep the same fine-tuning parameters for all ablation studies.

**Does text conditioning impact the results?** We investigate the performance of our model in three different scenarios: (1) when the conditioning prompt is unchanged in the evaluation setting, (2) we completely remove the conditioning prompt, and (3) we modify the temporal word ('before' to 'after' and vice-versa) in a hope to confuse the model. The results can be seen in Table 3.6a. Conditioning indeed allows the TCR module to extract more relevant features (+3.8). Furthermore, adversarial conditioning greatly impacts the performance of the model (-7.6).

**Do we need special tokens for conditioning?** If the model is fine-tuned for a specific task without the special tokens, it still performs reasonably well (73.5% vs 72.7% acc on NextQA with and without special tokens respectively).

**Does the number of frames matter?** The input video-sequence length is important to the model performance. In Table 3.6b we show the performance dependence on the input sequence length. Note that videos are on average 44s long, thus 124 frames equates to sampling at a rate of 2.5fps.

**How many queries should the LLM see?** While there is a benefit of perceiving

Table 3.6: Ablation studies on validation set of the NextQA dataset. Note that the ablations were done on a short training schedule.

| cond. | acc ↑ |
|---|---|
| yes | **64.9** |
| none | 61.1 |
| corrupt | 55.3 |

(a) Different conditioning prompts on *temporal-question set only*.

| #frms | acc ↑ |
|---|---|
| 32 | 64.4 |
| 92 | **66.2** |
| 124 | 65.9 |

(b) Impact of number of frames on model performance.

| #queries | acc ↑ |
|---|---|
| 32 | 62.7 |
| 64 | 65.8 |
| 128 | **66.2** |
| 256 | 64.3 |

(c) Impact of the total number of queries on model performance.

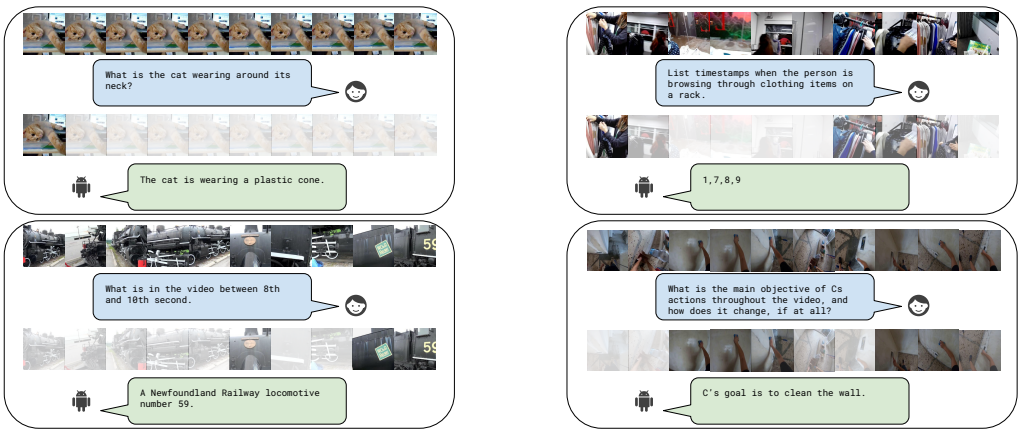



Figure 3.4: Examples of our model responding to various textual prompts taken from NextQA, EGO4D-MR, and YTT datasets. The opacity of the images in the second row is correlated to the mean patch attention score for that frame. Note that frames are subsampled and the TCR conditioning is not included for clarity.

a longer length of a video input-sequence, it has been observed that including more visual tokens as input to the LLM does not lead to a better performance [S. Yu et al. 2023]. Therefore in Table 3.6c we investigate how many queries the LLM should observe. Reducing the number of queries to a total of 128 (equivalent to four frames according to [J. Li et al. 2023b]) achieves optimal performance.

## 3.4 Related work

Our work spans many fields of video-understanding and we outline the most relevant related work below.

**Video-sampling techniques:** Sampling relevant frames from videos has long been a challenge in video understanding due to the highly redundant nature of video data. These methods either use a pre-processing module [D. Chen et al. 2011; Yeung et al. 2016; Korbar et al. 2019; Gowda et al. 2021; Yulin Wang et al. 2021; Zhi et al. 2021; Buch et al. 2022] to guide their model through multi-modal attention-like mechanisms [Ruohan Gao et al. 2020; Panda et al. 2021], or employ recursive reinforcement learning techniques [Wu et al. 2019] to select relevant parts of the videos. In temporal action detection, models are tasked with precisely locating action boundaries. Most commonly used datasets [H. Zhao et al. 2019; L. Wang et al. 2014; Caba Heilbron et al. 2015] are dominated by custom solutions or transformer architectures [J. Shao et al. 2023; C.-L. Zhang et al. 2022] built upon strong features [Tran et al. 2018; Yi Wang et al. 2022; Carreira and

Zisserman 2017; Tong et al. 2022; L. Wang et al. 2023].

**Egocentric videos understanding:** Extreme length and temporally sensitive nature of the tasks introduced in EGO4D required researchers to think about the problem of video-length on a different scale [Grauman et al. 2022; Mavroudi et al. 2023]. This has already yielded creative approaches to solve various challenges in the egocentric space [Tan et al. 2023; Jiang et al. 2023; G. Chen et al. 2022]. Also new and exciting benchmarks have been developed: for example the recent EgoSchema dataset [Mangalam et al. 2023], a manually annotated subset of EGO4D where each QA pair corresponds to a 3 minute-long video. A work particularly relevant to ours is SpotEM [Ramakrishnan et al. 2023], a lightweight sampling mechanism that makes use of low dimensional features in order to select video segments important for solving natural-language query challenge . Though impressive in performance, their method is limited to the type of embeddings used for sampling, and is thus less general than our approach.

**Video-language models and feature resampling:** VLMs have revolutionised the field of computer vision – the scale of the models and data they were trained on increased exponentially in a short period of time [Alayrac et al. 2022; Zellers et al. 2021; Jia et al. 2021a; J. Li et al. 2023b; Radford et al. 2021], some even being jointly optimised for images and video [Alayrac et al. 2022; Kuo et al. 2023; J. Li et al. 2023b]. The length of the videos these models can process often varies – [Alayrac et al. 2022] can process up to 8 frames, [Kuo et al. 2023] can process longer tubelets (at reduced receptive fields). None of these models can process videos over 16 frames at full resolution outright.

**Concurrent works on VLMs for video:** Extending capabilities of VLMs is a fast-paced area of research, and many works have appeared without being published. [Yanwei Li et al. 2023] conducted an orthogonal study, exploring how the embedding quality can reduce the amount of visual information necessary for large LLMs. We on the other hand introduce a bottleneck module to increase the amount of data processed without increasing complexity. [Maaz et al. 2023] focuses on interactive aspects by improving the LLM pipeline. We keep the pipeline fixed to seek improvements from the data. [Hang Zhang et al. 2023b] increases the amount of information by introducing additional modalities which would be an interesting next step for our work as well. Similar to us, [Song et al. 2023] aims

to increase video length in a BLIP2 model, but they do so via a memory bank. Combining a memory-augmented approach with reasoning over longer sequences would be a promising future work. Techniques like FlashAttention [Dao et al. 2022] or RingAttention [H. Liu et al. 2023] also allow the context window of a VLM to handle long sequences of frames but at the cost of significant growth in inference speed. These techniques are complementary to our proposal and could be integrated in the TCR to support even longer videos in the future.

**Future directions:** During the development of this work, plethora of large-language models with increasingly longer context-length have been released [G. Team 2024; R. Team et al. 2024]. These models allow for much more complex 'chain-of-thought' reasoning where models are asked first to identify key frames or sequences, and then answer questions about these sequences in particular. In fact, early pre-prints of such work have already been released as a proof-of-concept [Thawakar et al. 2025]. Our work is analogous, even if it performs such search "in software" rather than "in model".

## 3.5 Conclusion

We present a parameter-efficient, text-conditioned module and training method for bridging video-to-text gap that can be applied to a large number of frames in videos. Even though our model is entirely based on BLIP2 [J. Li et al. 2023b] architecture, introducing it in other VLMs would be straightforward. We believe that models capable of perceiving long video sequences such as TCR will open up a promising new direction in research.

## 3.6 Supplementary material

### 3.6.1 Further model details

**Implementation details:**

The model is implemented in FLAX [Heek et al. 2023], based on the scenic framework [Dehghani et al. 2022]. We use BLIP2 ViT-g Flan$T5_{xl}$ as a starting point and keep the vision and text models frozen. The number of *trainable* parameters is about 1% of the total memory footprint. JAX allow us to avoid storing gradients for non-trainable parameters, thus freeing up additional memory. Without training any part of the VLM, the model can process up to 124 frames during training time per TPU.

**Time tokenisation:**

In order to make the model time aware, we add time tokens to text and video-frames. We tokenise the time in half-second increments following an analogue procedure from [T. Chen et al. 2021] with $n_{\text{bins}} = 2048$. This means we can tokenise up to 17 minutes of video with a precision of half a second. We use this time format to express the temporal localisation of a segment in a longer video if that information is required (for example in moment retrieval or when describing a segment of the video). For each frame, we pass its timestep, tokenised as above, through a single-layer MLP in order to obtain learnable temporal embeddings we add to every frame.

### 3.6.2 Further training details

Section 2.2 of the main paper give the purpose of each training stage, and what is done, while Table 1 in the main paper investigates the impact of each stage on the downstream tasks. In this section, we first detail the pre-training stage and illustrate different tasks, then we give the fine-tuning details, and finally investigate 0-shot performance of multi-task trained models. All models are trained using the AdamW optimizer with global norm clipping of 1. Training hyperparameters are given in Table 3.7.


**Original caption:**
[6] [8] You can see my Nikon camera is in here.

**Legend**
Conditioning prompt (to TCR)
Text prompt (to LLM)

**Task 1: captioning**
[CPN] [6] [8]
What happens from 6 to 8 second?

**Task 2: temporal grounding**
[TRG] You can see my Nikon camera in here.
Reconstruct the following sentence: [MASK][MASK]
You can see my Nikon camera in here.

**Task 3: denoising**
[CPN] [6] [8]
a) [MASK][8] You can [MASK] is in here.
b) [6][8] You can see my [MASK].

Figure 3.5: Pre-training task examples with a condition sequence (TCR input) and the context sequence (LLM input). [CPN] (captioning), [TRG] (temporal grounding) are special tokens, and [6] is a sample of a tokenised timestamp. We use [MASK] (masking) token to form LLM prompts where applicable as they are integral part of T5's training [Raffel et al. 2020]. The model is tasked with predicting a caption "You can see my Nikon camera is in here" which is happening between 6th and 8th second of a video.

### Pre-training

After initialisation, we pre-train the model using the three outlined tasks (captioning, temporal grounding, denoising). Specifically, we define a dataset with each task, assign it a weight (1.0, 0.5, 0.5 respectively), and then, following procedure in [Alayrac et al. 2022], we accumulate gradients across each task with loss being weighted per-dataset. Loss weights were defined empirically and kept fixed throughout. An illustration of dataset tasks can be seen in Figure 3.5.

### Fine-tuning procedure

Due to a diversity in our downstream tasks, both in terms of the data type (e.g. instructional videos, vs egocentric videos) and tasks for which we train special tokens, devising a model that can follow specific dataset conventions or match a particular answer format from noisy pre-training data is a challenging task. We

| Stage | Dataset | Batch size | LR | Epochs | Warmup steps |
|---|---|---|---|---|---|
| Pre-training | YTT - captioning<br>YTT - temporal grounding<br>YTT - denoising | 256 | 1e-4 | 10 | 1k |
| Fine-tuning | MSR-VTT | 128 | 1e-5 | 20 | 2k |
| | NextQA | 128 | 1e-5 | 40 | 2k |
| | NextQA - short | 64 | 5e-5 | 10 | 1k |
| | EgoSchema | 64 | 1e-5 | 20 | 1k |
| | EgoLTA | 64 | 1e-5 | 20 | 1k |
| | Ego4D MQ | 128 | 1e-5 | 10 | 500 |

Table 3.7: Pre-training and fine-tuning hyper-paramerters

found that fine-tuning the model alleviates these challenges and helps our model's performance. If heplful, we introduce a new special token for the task, and proceed to fine tune the model on the downstream task specifically.

**EgoSchema fine-tuning:** EgoSchema [Mangalam et al. 2023] dataset does not come with a dedicated training set. We use EGO4D narrations to fine-tune the model as follows. We first identify video IDs from EGO4D that are also present in EgoSchema benchmark and remove them. We also remove all videos where narrations are tagged with an '#unsure' tag. We then fine-tune the model with a template "What does the templ do from start to end?" if narration is tagged with '#C' or '#O', or "What happens from start to end" if the narration is tagged with '#summary'. Fine-tuning hyper-parameters can be found in Table 3.7.

**Zero-shot performance and multi-task fine-tuning.**

Models such as Flamingo [Alayrac et al. 2022] can be 'prompted' to solve tasks in a low-shot setting, thus making fine-tuning unnecessary. Flamingo, however, is trained on much wider variety of datasets (mostly from an image domain, where large datasets with less noisy supervision are available). Our model lacks this pre-training and, additionally, faces a significant challenge in this setting, as the number of output queries from TCR (128) is significantly higher than that of Perciever Resampler (8) resulting in longer sequences. However, having a single general model capable of solving multiple tasks is an appealing proposition. Therefore, we present the results with and without fine-tuning in 0-shot and $k$-shot setting in Table 3.8. First two rows use a model initialised and pre-trained without any modifications. Second two rows show the performance of a *single*

| FT | k-shot | NextQA (Acc) | Ego4D-LTA (ActionED) | Ego4D-MQ (avg mAP) | EgoSchema (acc) |
|---|---|---|---|---|---|
| None | 0 | 34.2 | 0.9354 | 19.88 | 14.5 |
| | 2 | 37.3 | / | 21.75 | 19.0 |
| Multitask | 0 | 61.7 | 0.9199 | 21.74 | 28.7 |
| | 2 | 64.1 | / | 23.01 | 30.9 |
| Per task | 0 | 73.5 | 0.8782 | 24.51 | 34.2 |
| | 2 | 73.5 | / | 24.83 | 34.1 |

Table 3.8: Comparison of pre-trained vs fine-tuned model on the downstream datasets in 0- and few-shot setting. Numbers reported in the main comparison are fine-tuned per-task and evaluated in a 0-shot setting for consistency. Note that for Ego4D-LTA task, we couldn't fit the entire action sequence of a multi-shot example due to the maximum sequence length of our LLM.

model (one set of weights) that is trained on all downstream datasets jointly with equal weighting. In other words, after pre-training, we fine-tune the model jointly on *all* downstream datasets for 20 epochs, accumulating the gradients over all datasets with equal weight. Finally, in the last two rows, we present the results of four fully fine-tuned models. As observed in models utilising large instruction-tuned language models, our model performs better in a few-shot setting [Alayrac et al. 2022]. As expected the improvement in fully fine-tuned models is minor since models are already specialised for the specific task. We want to highlight how our model with a single set of weights can perform well across different tasks simply with the addition of a few prompts for in context learning. In case even stronger performance are needed, we can further improve by specialising the model for the specific task/dataset, and we report these numbers in the main paper.

### 3.6.3 Study of standalone TCR architecture

In order to further validate our architecture choice, below we conduct a series of experiments with various uses of TCR without the LLM. In section 3.6.3, we run experiments with TCR simply acting as a "head" for a visual model. In this case, the visual backbone is fixed and TCR is trained from scratch on a target task without special tokens. In section 3.6.3, we initialise and pre-train TCR as described in Section 2.2 of the main paper, and instead of connecting it to an LLM, we fine-tune TCR with a two-layer MLP on top of it.

### Study of TCR's sampling capabilities

Transformer decoder modules have been shown to perform well on action-detection tasks [T. Han et al. 2022a], and several works showed they can cross-attend image information with semantic information [Kamath et al. 2021; Korbar and Zisserman 2022a]. In this section, we seek to validate our architecture design on a set of simplified tasks. Specifically we show that *a)* TCR can be trained independently to sample relevant features for action classification, *b)* 'select' the frames corresponding to the query action within a couple of videos, and finally *c)* perform semantic tasks such as counting on actions. We conduct these experiments on various splits of the Kinetics dataset [Kay et al. 2017; Dwibedi et al. 2020].

**Sampling for action recognition:** First, we train the model for action recognition. Most models average predictions over multiple uniformly-sampled frames or short clips in order to obtain a video-level prediction [Tran et al. 2018]. Dedicated samplers such as [Korbar et al. 2019; Zhi et al. 2021] have been trained to specifically sample relevant frames or clips. The goal of this experiment is to see whether TCR can sample features in such a way to improve overall video-level accuracy. In other words, given a 300 frames input video, can the model sample 32 frames in such a way that classification performance is better than other naive benchmarks. To do so, we extract frame-level features from the frozen ViT-g model, add sinusoidal positional embeddings, and pass them to TCR as a key-value pair to the cross attention layers. We "prompt" the model with 32 learnable queries, each of which is processed through a 400-way classifier whose predictions are then averaged to obtain the final prediction. This model is trained for 10 epochs on Kinetics-400 dataset and compared to: linear classification of features from 32 random frames, self-attention pooling where the classification layer is done on a learned `CLS` token, and on two top video clips sampled by SCSampler [Korbar et al. 2019] (32 frames). TCR is able to outperform naive sampling methods and bests standard attention-pooling approach, which means it can learn about relative importance of input features to the text prompt. It, however, cannot best models which were trained with explicit saliency supervision such as SCSampler. This experiment indicates that incorporation of explicit saliency supervision such as in [Korbar et al. 2019; Zhi et al. 2021; S. Yu et al. 2023] is a promising direction of further research. Results can be seen in Table 3.9.

| Sampling method | Accuracy ↑ |
|---|---|
| random | 74.3 |
| Attention | 75.2 (+0.9) |
| TCR (ours) | 75.7 (+1.4) |
| SCSampler [Korbar et al. 2019] | 77.8 (+3.4) |

Table 3.9: Standalone TCR (without using a LLM): Linear classification of 32 frames sampled from the Kinetics400 [Kay et al. 2017] val set using various sampling methods.

**Text-conditioned sampling:** Next, we want to verify whether the model can "select" correct frames from a video given an action class prompt. To this end we design an artificial experiment by stitching 128 frames of up to 8 kinetics videos, and the task is to classify frames containing the prompt action as positive examples. Only a single video of a query class can be present, it's location within the video is randomised, and the number of frames sampled from each video is random (from 1 to 64). We add position encoding (from 0 to 127) to each frame, and feed them to TCR. For query input, we use the action prompt followed by 32 learnable queries. After TCR, we pass each query through 128 one-vs-all binary classifiers, one for each of the input frames. The classifiers are trained to predict the presence or absence of the action at the corresponding input frame. At inference time, if the confidence score for classifier $j$ is higher than 0.5 we add to the predictions of query $k$ the frame $j$ (note that each query can predict more than one frame). Finally we consider the set of all predictions across all the queries and compare it to the ground truth frame ids. If the frame ids of the frames corresponding to the action prompt are contained in the predicted set we count that as a positive paring and the rest as negative. We compare this TCR approach to simply training the per-frame 400-way action classifier on top of ViT, and count the label as a positive if the target class is predicted. We were able to achieve precision of 0.75 and recall of 0.68, which is higher than simply classifying action label for each frame of the video (0.70, 0.44).

**Text-conditioned semantic processing:** Finally, with minor addition, we show that TCR architecture can be used as a counting mechanism as well due to its potential to "look" and reason over every single frame. We consider the settings of [Dwibedi et al. 2020], feed the video at full temporal resolution, and form the query the same as in previous paragraph. We then concatenate the queries and

| Method | MAE ↓ | OBO ↓ |
|---|---|---|
| [Dwibedi et al. 2020] | 0.36 | 0.30 |
| TCR | 0.33 | 0.28 |

Table 3.10: Standalone TCR (without using the LLM): Counting repetitions in videos on the Countix [Dwibedi et al. 2020] dataset.

pass them through a TransRAC Period Predictor [H. Hu et al. 2022] to obtain the final count. The TCR module and TransRAC predictor are trained on syntetic data as descibed in [Dwibedi et al. 2020] for 24 epochs. Results outperform the SOTA benchmark, as can be seen in Table 3.10.

### TCR without LLM

In this section, we show that the pre-trained TCR module can be easily adapted to solve discriminative tasks without the need for an LLM. The difference to the previous section is the fact that TCR is initialised and pre-trained using the procedure outlined in Section 2.2 of the main paper. For this set of experiments, we keep the pre-trained visual encoder and TCR module frozen and train simple 2-layer MLP adapters on top of them. For the EGO4D moment query task, we adapt the alignment module from [T. Han et al. 2022a] which trains two linear layers on top of TCR output with set-prediction loss consisting of two elements: one predicting whether the sequences correspond to a given action and one regressing the temporal span of the query. For the natural language query task, we adopt the same training procedure as [A. Yang et al. 2022a] for spatiotemporal localisation on top of TCR output. Results can be seen in Table 3.11. It is notable that, without additional pre-training of the TCR module, the features it outputs can be generalised well to tasks that benefit from video sampling. Our general pre-trained architecture comes close to the specialised solution [J. Shao et al. 2023] to the Moments Query challenge ($-1.14$ Avg.MAP), giving us future direction for improving it for tasks of video retrieval. Similarly on the Natural Language Query challenge, with only a small adapter trained for spatiotemporal localisation our model challenges the state-of-the-art [Ramakrishnan et al. 2023] ($-0.14$ MR@1) which has a sampler and detector head trained for this task specifically. We believe that optimising VLMs for regressive tasks requiring high spatiotemporal resolution will be an important future research direction.

| Method | MQ | | NLQ | |
|---|---|---|---|---|
| | AVG MAP | R@1 | MR@1 | MR@5 |
| SpotEM [Ramakrishnan et al. 2023] | / | / | **11.56** | 19.90 |
| ASL [J. Shao et al. 2023] | **27.85** | **46.98** | / | / |
| TCR | 26.71 | 44.96 | 11.42 | **19.95** |
| TCR w/ LLM | 25.45 | 43.72 | 10.12 | 18.99 |

Table 3.11: Performance of TCR as a standalone module on discriminative downstream tasks. LLMs tend to struggle with regression tasks [T. Chen et al. 2021] and while we present these numbers in the main paper due to their flexibility to solve a wide variety of task, we also show that TCR can provide better performance for these tasks using ad-hoc regression adapters and significantly narrowing the gap with state of the art methods.

# Chapter 4

# Personalised CLIP or: how to find your vacation videos

*The paper has been accepted at BMVC 2022.*

While Chapters 2 and 3 deal with technical building blocks of story-understanding, the the following chapters deal with particular applications of personalisation that make video- and story-understanding tasks "human centric". In this chapter, we focus on an accurate compound retrieval using identities (names and likeness) as a part of the query. The resulting model is able to recognize actors in complex scenes from their names alone without losing expressivity of a complex language model. By learning a mapping from image to text, model is able to recognize actors it hasn't seen during training from a single query image.

# Personalised CLIP or: how to find your vacation videos

Bruno Korbar Andrew Zisserman

Visual Geometry Group, University of Oxford

March 29, 2026

**Abstract**

In this paper, our goal is a person-centric model capable of retrieving the image or video corresponding to a personalized compound query from a large set of images or videos. Specifically, given a query consisting of an image of a person's *face* and a text *scene description* or *action description*, we retrieve images or video-clips corresponding to this compound query. We make three contributions: (1) we propose CLIP-PAD, a model that is able to retrieve images/video given a personalized compound-query. We achieve this by building on a pre-trained CLIP vision-text model that has compound, but general, query capabilities, and provide a mechanism to personalize it to the target person specified by their face; (2) we share a new *Celebrities in Action* (CiA) dataset of movies with automatically generated annotations for identities, locations, and actions that can be used for evaluation of the compound-retrieval task; (3) we evaluate our model's performance on two datasets: Celebrities in Places for compound queries of a celebrity and a scene description; and our new CiA for compound queries of a celebrity and an action description. We demonstrate the flexibility of the model with free-form queries and compare to previous methods.

## 4.1 Introduction

Suppose that you want to find the video of your vacation where *you* ran in a red and white striped shirt in *front of the Parthenon* in Athens amongst all videos on your phone. Or imagine that you cannot remember the name of the movie

where *Ben Stiller* runs *on a boat*. Or maybe you don't even know Ben Stiller's name, but have seen him in another movie and can find his picture. On its own, finding out if you (or Ben Stiller) are in the video or if the video contains a boat or Parthenon (single-query) is a well-researched problem. Searching with multiple (and potentially multi-modal) *compounded* queries in large databases, however, is still a challenging and under-researched proposition. In this paper, we aim to tackle this problem and deliver a model that is capable of precise retrieval for compound queries looking for specific people in specific places or performing a specific action.

Vision-language models such as CLIP [Radford et al. 2021] and ALIGN [Jia et al. 2021b] have transformed the performance of many visual-language tasks. These models use a dual encoder and are trained on large-scale datasets [P. Sharma et al. 2018; Ng et al. 2021] using a contrastive loss. In particular, they can be used to retrieve an image or video given a free-form text description. However, this freedom of using a sentence to describe the sought visual content is also a limitation. How can a query also include specific *visual* content such as a particular person (specified by their face), or a particular object instance (specified by an image of it)? If the model doesn't have the notion of identity, how can it find a conceptual difference between different instances ("Brad Pitt" vs "George Clooney")? In this work, we propose a simple addition to the foundation models that would allow them to do just that.

The key idea is to provide a mechanism to adapt a face image (that specifies the identity) to 'act as' a text token that describes the identity within the query, as illustrated in Fig. 4.1. We show that making the model identity-aware works remarkably well compared to a zero-shot model. It significantly improves the retrieval performance on the Celebrities-in-Places (CiP) [Zhong et al. 2016a] compound-query retrieval dataset. Furthermore, we show that we can use our personalised model even in video scenarios. To this end, we annotate a human action movie dataset with person-specific labels, which we call *Celebrities in Action* (CiA), and evaluate performance compared to existing retrieval methods.

This person adaptive model, which we refer to as CLIP-PAD(Person ADaptive), has applications in real-world video-retrieval applications, such as searching a video archive for historical celebrities performing actions or in particular places. For


Figure 4.1: An outline of the CLIP-PAD architecture, when it is queried by a target face image (in this case of Michael Richards) and a sentence. Gray modules (dark background) are pre-trained and frozen, whilst blue ones (light background) are learned. The model correctly retrieves a video clip that matches the text description (specified by the sentence) personalized to Michael Richards (specified by his face).

example, it would enable a broadcaster such as the BBC or a stock company such as Shutterstock to carry out a personalized compound query, using free-form text, to search the archive on their visual content, without requiring any text annotation of the archive. The scheme is clearly applicable also for seaching personal images and videos.

In section 4.3 we outline the CLIP-PAD model, in particular the simple adaptor mechanism for the CLIP model that allow us to retrieve images (and video) given a personalized compound-query. In section 4.4 we describe the CiA dataset, a new benchmark for video compound retrieval, and how it is generated. Finally, in section 4.5 we demonstrate the high performance of the model against baselines on both CiP and CiA under various different retieval scenarios. In the supplementary material, we additionally show a real-world retrieval example from various episodes of the Seinfield TV Show.

## 4.2 Related work

**Foundation video-text models.** Ever since AlexNet [Krizhevsky et al. 2012b], pre-trained models have been used to boost performance or bootstrap models on downstream tasks [S. Ren et al. 2015]. Using video-to-text correspondence to develop strong pre-trained models is not a novel phenomenon [Miech et al. 2019b; Miech et al. 2020; Gabeur et al. 2020], however, recently the scale of training data became so large that a new generation of pre-trained (sometimes also called foundation) models was developed with the capacity of implicitly solving even the

tasks they weren't trained for explicitly (CLIP [Radford et al. 2021], ALIGN [Jia et al. 2021b], BLIP [J. Li et al. 2022], FILIP [Yao et al. 2022], to name just a few). Following these advances, we use CLIP as a backbone embedding for our model and use a trainable-prompting paradigm to achieve our tasks.

**Text-to-video retrieval.** Video-text retrieval has long been a fruitful research direction, with a plethora of datasets available [Hendricks et al. 2018; L. Zhou et al. 2018; Lei et al. 2021a; Krishna et al. 2017; Rohrbach et al. 2017]. Due to the processing costs, most models were traditionally developed on top of feature "experts" [Croitoru et al. 2021; Yang Liu et al. 2019; Brown et al. 2020; Miech et al. 2019b; Miech et al. 2018; Gabeur et al. 2020]. First, the features are extracted from models that were pre-trained for a specific task (and often combined), and then the retrieval model was trained. With the rise of large-scale models that use video-to-text correspondence [Dong et al. 2019; Miech et al. 2019b; Miech et al. 2020; Gabeur et al. 2020; Lei et al. 2021b; Dong et al. 2021], the focus switched to direct similarity metrics between retrieval text queries and videos. Foundation models dominate the video-text retrieval leader boards, even in the zero-shot setting [Radford et al. 2021; Patrick et al. 2021; Hu Xu et al. 2021; Luo et al. 2020; Bain et al. 2021; Bain et al. 2022]. This task is closely related to the tasks of text-to-video-localization and (corpus) moment retrieval [Hao Zhang et al. 2021], for which various architectures have been proposed [S. Xiao et al. 2021; X. Yang et al. 2021].

**Compound (person-specific) query retrieval.** Text-to-video retrieval is a notably different task to compound (person-specific) query retrieval as it requires much less specificity. Traditional text-to-video retrieval datasets are often depersonalized (e.g. in LSMDC, all names are substituted by "someone" [Rohrbach et al. 2017], so a query "Kramer enters the apartment", and "George enters the apartment" would be indistinguishable). In the case of compound retrieval, the focus is on specificity. Despite it being a common everyday task, very few datasets have been released and we believe it is an under-explored task. We address this by presenting a new dataset based on the existing Hollywood2 benchmark [Marszałek et al. 2009] and High-Five human interaction dataset [Patron-Perez et al. 2010], annotated with identities to enable compound-queries and responses.

**CLIP and its shortcomings.** CLIP is a dual-encoder foundation model intro-

duced by [Radford et al. 2021]. The premise is rather simple, given an image processed by a visual encoder and corresponding text processed by a text encoder train the model contrastively (using symmetric contrastive loss), and train it on an unprecedented scale (over 400M labeled images). Since its publication and release, CLIP models have been used in a myriad different ways for a plethora of tasks, often unrelated to their original training task [Radford et al. 2021; Hu Xu et al. 2021; Luo et al. 2022; Narasimhan et al. 2021; Shen et al. 2022]. To harness emerging properties of these models, CLIP-based models are used in a zero-shot manner [Portillo-Quintero et al. 2021], simply by modifying the prompt to match the desired output [Radford et al. 2021; Narasimhan et al. 2021]. If we wanted to know if the image contains a bird or a dog, we'd simply feed an image and two text prompts (`"Image of a dog"`, `"Image of a bird"`), and then find which text prompt has higher similarity to the image. Other common ways of "adapting" clip are via learning models on top of the visual embedding [Radford et al. 2021; Shen et al. 2022; Luo et al. 2022], or via learnable prompting [Ju et al. 2021]. Despite the size and the variety of the training dataset CLIP is trained on, there are some tasks that it is inherently less suited to – e.g. tasks containing actions. This shortcoming is not surprising as the model is trained on images alone, and as recent works have shown, adapting it to the video domain does take additional ingenuity [Portillo-Quintero et al. 2021; Luo et al. 2022; Bain et al. 2021]. The most prevalent approaches are training specific prompts to feed into the model [Ju et al. 2021], augmenting the model architecture to accept different embedding types [Bain et al. 2021], or training aggregation models on top of it [Luo et al. 2022]. Our approach roughly falls amongst "learnable prompting" approaches.

## 4.3 Personalising CLIP: CLIP-PAD

To personalise CLIP, we adopt a prompt-learning method in order to adapt the model for our task. We wish to use a free-form text query but adapted to the specific target person. To achieve this we fine-tune the CLIP text encoder to "recognise" the target person given a text query and a prompt starting from a loose crop of the person's face.

**Architecture overview.** The architecture is illustrated in figure 4.1. The target face image is processed by a pre-trained face encoder, and passed through a Multi-Layer Perception (MLP) to adapt the face encoding to the space of the word encodings. The entire compound query is then encoded using the CLIP text encoder. So, for example, to find an image of Tom Cruise running, the face image would be of Tom Cruise, and the text query would be "An image of *TOK* running", where *TOK* is the output of the MLP adaptor. Note, the only trainable components of the architecture are the MLP adaptor, and in case of text-queries, the CLIP Text encoder. All the other modules: ConvNet face encoder, and CLIP image encoder are pre-trained and frozen. The details of each module are given in the implementation details.

**Dataset retrieval.** In order to perform a text-to-image retrieval, embeddings are generated for every image in the dataset using the CLIP image encoder. The match for a given (compound) query is then obtained by finding the image representation with the highest cosine similarity to the CLIP embedding of the query text. Query text can be formed of text only ("An image of Tom Cruise running"), text with a visual query ("An image of *TOK* running"), or a combination ("An image of Tom Cruise *TOK* running").

**Training.** We start from the pre-trained weights for CLIP [Radford et al. 2021], and keep as many parameters fixed as possible, training only the MLP adapter and the CLIP text encoder. The purpose of training the model is to make it identity-aware. To achieve this, we train the MLP adaptor and the CLIP text encoder to be able to discriminate amongst different identities. To do so, we fine-tune our model on a *person-recognition dataset*. During training, each batch contains 60% of queries containing the image token, 20% being names of celebrities as strings, and the rest a combination of both. We found empirically that this ratio yields the optimal performance in the general case where we might or might not have a visual query. The model is trained using the symmetric contrastive loss as proposed in [Radford et al. 2021] until it can perform the person-classification task sufficiently well, with the criterion being different for each evaluation dataset as outlined in the implementation details.

### 4.3.1 Implementation details

**Model details.** Our model starts from an unmodified pre-trained CLIP dual-encoder architecture: the visual encoder is a ViT-B/32 transformer, and the text encoder is a 3M-parameter 12-layer 512-wide model with 8 attention heads as outlined in [Radford et al. 2021]. The face encoder is a SE-ResNet-50-128D model pre-trained on the person dataset (from [Cao et al. 2018a]). It is a ResNet50 [K. He et al. 2016] with Squeeze and Excitement (SE) layers [J. Hu et al. 2018] that outputs 128-dimesional vector for each person. The output of the face encoder is processed via a two-layer MLP, taking the dimension from 128 to 256 and 256 to 512 respectively with ReLu non-linearity. This 512 vector is then used as an input embedding to a CLIP text encoder of the same dimension. This embedding is then processed as if it was a standard input to a CLIP text encoder – i.e. position embeddings are added following the protocol in [Radford et al. 2021] before it is ingested by the encoder. Note that on the video data, we employ mean pooling similarity calculator to aggregate visual embeddings from different frames of the video on top of the CLIP vision encoder as proposed in [Luo et al. 2022].

**Training details.** The training procedure and dataset depend on the target dataset. In general, we would want as many known celebrities to be a part of our model and thus ideally we would want to train it on a large VGGFace2 [Cao et al. 2018a] dataset. This dataset however contains some of the test 'unseen' faces from the Celebrities in Places (CiP) dataset, and having an embedding trained on these images would yield an unfair comparison to the baseline [Zhong et al. 2016a]. Therefore, for evaluation on CiP, the model is trained on the loose crops of the VGGFace dataset [Parkhi et al. 2015] – it being the same pre-training dataset used by [Zhong et al. 2016a]. We choose to use loose crops (as opposed to traditionally used tight crops) to minimise the domain gap to the target images that will often show entire body as we are not concerned by a potential impact on the overall person-classification task.

The model is trained for 10 epochs, evaluating it on a held-out validation set from *VGGFace2* at the end of every epoch. We early stop the model if it achieves 85% or 95% accuracy on validation when trained on VGGFace and VGGFace2 datasets respectively.

For CiA, we train the model on loose crops of VGGFace2 [Cao et al. 2018a] following the same procedure as above. We additionally fine-tune it for 5 epochs on the training movies of CiA, to account for the fact that CLIP might not have been trained with action classes in mind and that actors might not be present in the VGGFace2 dataset. We keep the ratio of the person queries constant during the fine-tuning process.

The hyper-parameters and training ratios of queries were found via a linear search based on model performance on the CiP held-out validation set. The optimal parameters were selected based on maximum average performance when using text queries, text+image queries, or image only queries. We then use the same hyper-parameters for all other datasets.

## 4.4 Celebrities in Action Dataset

In this section we describe how we annotate the Hollywood 2.0 dataset for person-centric retrieval. The Hollywood 2.0 dataset consists of 12 classes of human actions and 10 classes of scenes distributed over 3669 video clips from 69 movies (33 training, 36 testing) [Marszałek et al. 2009]. Examples of actions include 'eating' and 'running', whilst the scenes include the 'office', 'shop', and 'car'. Test sets for both actions and scenes have been manually cleaned, whilst the training set has been annotated automatically and has inherent noise (upon manual inspection of randomly selected 50 clips, 2 were unclear). To better evaluate classes with human interactions, we also include 172 clips from the High-Five human interaction dataset [Patron-Perez et al. 2010] which has collaborative actions such as 'hugging', 'kissing' and 'giving a high five'. The limitation for person-centric queries is that although the clips are labelled with the actions performed, the person performing the action is not labelled.

In order to form the *Celebrities in Action* (CiA) dataset [1], we automatically annotate the clips with the person performing the action. To achieve this we use the automatic video annotator by Brown *et al.* [Brown et al. 2021b]. In brief, this method uses the IMDB cast list from the movie to obtain face images for each actor in order to classify their occurrences. On the video side, faces are detected

[1] https://www.robots.ox.ac.uk/ vgg/research/celebrities-in-action/

| split | clips | % in VGGFace2 | person (per clip) | actions (per actor) | scenes (per actor) |
|---|---|---|---|---|---|
| train | 1308 | 23.6 | 2.5 | 4.1 | 3.1 |
| val | 328 | 23.6 | 3.1 | 4.6 | 2.9 |
| test | 1052 | 38.2 | 2.0 | 3.9 | 3.1 |

Table 4.1: CiA stats. We report the total number of clips in every split, percentage of annotated celebrities in the split clips that are present in the fine-tuning VGGFace2 dataset [Cao et al. 2018a]. Furthermore, we show how many people are on average present in each video clip, and we try to find if a given actor on average performs different actions or appears in multiple scenes in the data.

and tracked in each clip, and then an identity is associated with the face track if it is classified as one of the known actors from that film. In the case of multiple actors getting annotated (in 47% of video clips), we select the most confident one as our final annotation. This can potentially cause incorrect or ambiguous examples as seen in the last two columns of figure 4.2. We do not address this issue as ambiguous or incorrect labels are rare. We manually verify the label correctness on a randomly selected 100 clips from the test set and find the actor annotations to be correct for 97 of them (i.e. annotated actors are visible in the video clip). We separate the training set into the training, and held-out validation set (for model development) – not according to movies but rather, according to the clips in the training data. We only use the training data to fine-tune CLIP for better action performance.

To form queries from the data, we form template sentences in three ways: 1) "{`celebrity`} is doing {`action`}", 2) "{`celebrity`} in {`place`}", and 3) "{`celebrity`} in {`place`} doing {`action`}". For the "{`celebrity`}" token, we consider both text, image, and a combination.

Statistics of the dataset can be found in Table 4.1, and some example from the annotated dataset are given in Figure 4.2.


Figure 4.2: Examples of correctly labeled, ambiguous, and incorrectly labeled (all regarding the person annotation) from CiA.

## 4.5 Experiments

In this section we evaluate the CLIP-PAD model for two person-centric retrieval tasks using compound queries. The first is 'a person in a place', and for this we evaluate on the existing *Celebrities in Places* benchmark dataset where images are annotated with both the celebrity and the place. The second task is 'a person doing something', and for this we evaluate on the 'Celebrities in Action' dataset described in section 4.4.

Since CLIP has been trained on millions of images, it is likely that it will have seen examples of some of the celebrites labelled with their identity. However, the long-tailed nature of the "celebrity" classes makes it unlikely that the model would have seen *all* or even most of them. Similarly, is likely that CLIP has seen all the classes of places in the Celebrities in Places dataset. For this reason we include zero-shot baselines where we simply evaluate the retrieval performance given the celebrity's name. For example, for the 'Celebrities in Places' dataset we evaluate both for the celebrity alone with queries such as "An image of Tom Hanks", as well as for celebrities in a place with queries such as "An image of Tom Hanks in the supermarket".

### 4.5.1 Celebrities in Places

Celebrities in places (CiP) is an established benchmark for a person-centric compound retrieval. Its test set contains 15.1k images with 2.3k celebrities in 16 places. This dataset contains a wide variety of celebrities from the VGGFace


Figure 4.3: Qualitative retrieval examples for various queries on CiP [Zhong et al. 2016a] dataset sorted by retrieval rank. A green boarder indicate the correctly retrieved class, and a red one indicates an incorrect one.

dataset ('seen' – 0.6k), and some that were not included in VGGFace ('unseen' – 1.7k). Places include visually different scenarios such as the 'beach', the 'stage', and the 'golf course' to name a few. Note that the dataset is not class-balanced, so the most common place ('stage') has over 2k examples, while the least common place ('desert') has only 102 examples. Similarly, the most common celebrity is present in over 60 images, whilst the least common ones are present in only 1. The model is evaluated by forming a query such as '*Celebrity* in *place*' (where the provided queries cover only a subset of the 2.3k celebrities: 792 unseen and 223 seen), and 'place' is 1 of 16 possible places, and retrieving the correct example amongst 15.1k annotated images, and an additional annotated distractor-images provided sampled from other datasets (36k images in total for the annotated test set and distractors).

In table 4.2a, we first compare the performance of our model to other baselines on *non-compound* queries. For example, we would query our model with '`Image of John Wayne`' for a face retrieval, or '`Image of the golf course`' for the place retrieval. We compare our model to the compound query retrieval baseline by [Zhong et al. 2016a] which employs two separate ConvNets to extract face and place embedding respectively and trains a joint embedding used for retrieval with a multiclass hinge loss. We also compare our model to a zero-shot CLIP [Radford et al. 2021] model evaluated in the same way as our model with the *txt* query.

We can observe that the original CLIP model can very accurately retrieve places without fine-tuning, however, it lacks the capability of retrieving people with high precision. The difference in performance between the seen and unseen classes is negligible. This is likely explained by the fact that CLIP's visual descriptor is highly discriminative 'as-is' and is likely to have seen some of the celebrities belonging to the 'unseen' set. Lastly, we can see that personalising our CLIP text encoder doesn't impact the performance of the retrieval for 'places' (−1.6mAP) while it dramatically improves the performance on the face-retrieval (+22.0mAP).

In table 4.2b we compare the *compound-query retrieval* performance. As we did not observe large difference between the 'seen' and 'unseen' faces, we average the results weighted by the number of queries. The personalised model performs significantly better than both baseline results [Zhong et al. 2016a] and zero-shot CLIP [Radford et al. 2021]. Qualitative examples can be found in figure 4.2.

| Method | mAP (face:unseen) | mAP (face:seen) | mAP (places) |
|---|---|---|---|
| [Zhong et al. 2016a] | 74.1 | 73.7 | 51.1 |
| CLIP [Radford et al. 2021] | 60.1 | 59.1 | 83.1 |
| text | 82.1 | 82.0 | 81.5 |
| img | 84.0 | 83.7 | – |
| text+img | 86.7 | 86.6 | – |

(a) Baselines results for retrieving non-compound query-classes of the CiP dataset as compared to our model (bottom section) with different person prompts.

| Method | mAP | R@5 |
|---|---|---|
| [Zhong et al. 2016a] | 63.4 | 82.2 |
| CLIP [Radford et al. 2021] | 60.6 | 79.5 |
| text | 79.5 | 94.5 |
| img | 77.7 | 93.0 |
| text+img | 79.5 | 94.5 |

(b) Compound-retrieval results on the CiP dataset. Note that we average results on seen/unseen faces.

### 4.5.2 Celebrities in Action

We first evaluate three zero-shot baselines in Table 4.3; we measure if the model can solve actor classification, action classification and scene classification on the training set in a zero-shot setting. If the task is to classify an actor, we follow the original CLIP zero-shot protocol [Radford et al. 2021] and for a given video-clip we

compute similarities to text queries using the actor names as classes (e.g. `"Image of Tom Cruise"`) and pick the most similar one as a predicted class. Classification for actions and scenes is done in an analogous way. Note again, that a zero-shot CLIP model can classify both action and scene with a very high degree of accuracy, however where it fails is the classification of the actor. Our model alleviates this issue to a large extent. Furthermore, we see a notable improvement in classification accuracy (6.2%) when querying our model with images. This is likely due to the fact that only a small proportion of faces can be found in VGGFace2 dataset (see Table 4.1), thus the additional information does help significantly.

| Method | Actor classification | Action classification | Scene classification |
|---|---|---|---|
| Random | 0.7 | 8.3 | 10 |
| CLIP [Radford et al. 2021] | 9.1 | 73.1 | 91.6 |
| CLIP-PAD-*text* | 26.5 | 75.7 | 91.6 |
| CLIP-PAD-*img* | 32.7 | – | – |
| CLIP-PAD-*text+image* | 33.8 | 75.7 | 91.6 |

Table 4.3: Baseline zero-shot classification results on CiA can be found in the top section. Ours are in the bottom section, with modalities used as a query appended. Results are given in % accuracy.

In order to compare our results with a more-traditional retrieval models, we compare them to two modern baselines: a mixture of experts model [Yang Liu et al. 2019], and CLIP in a zero-shot setting using the straight-CLIP protocol [Radford et al. 2021; Portillo-Quintero et al. 2021]. We report recall at ranks 1 and 5 (R@1, R@5). The results can be seen in the table 4.4. Two things stand out immediately: First is the discrepancy between the 'action' and 'place' retrieval of zero-shot CLIP: even on 'simple' action classes, it does demonstrably worse when compared to the compound retrieval on 'places', which it recognises well. Second is the performance increase coming from our personalised model when compared to zero-shot CLIP. Note that the baseline models cannot query the model by using multi-modal queries. To overcome this limitation, we present a two-stage baseline in the supplementary material.

### 4.5.3 Real-world retrieval example

In order to present a real-world example of personalised retrieval, we apply our model on the entire Season two of the Seinfeld TV show. We want to see how many

| Method | person query | | action | | place | | action + place | |
|---|---|---|---|---|---|---|---|---|
| | text | rgb | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 |
| CE [Yang Liu et al. 2019] | ✓ | * | 35.3 | 65.1 | 36.7 | 65.3 | 32.1 | 62.5 |
| CLIP [Radford et al. 2021] | ✓ | | 42.4 | 73.5 | 58.2 | 78.1 | 39.9 | 72.0 |
| CLIP-PAD | ✓ | | 62.1 | 81.3 | 67.8 | 87.0 | 64.2 | 81.1 |
| CLIP-PAD | | ✓ | 64.5 | 83.7 | 68.2 | 87.3 | 64.9 | 81.8 |
| CLIP-PAD | ✓ | ✓ | 65.0 | 85.4 | 71.5 | 88.6 | 66.3 | 82.7 |

Table 4.4: Retrieval results on CiA with celebrities doing 'action', in 'place', or combined respectively. Our model's results are presented in the lower part of the table. 'query' column refers to the modality of the person-query used. CLIP has not been fine-tuned or modified in any way, and we train the CE model on the training set with experts and parameters given in the supplementary. '*' denotes that while CE does not use face embedding as a query, to make the comparison as fair as possible we include the face query embedding as an additional 'expert' input to the model.

occurrences of "Michael Richards entering the room" we can correctly retrieve from a total of 483 video clips. By our count, Richards is portrayed entering the room 26 times in the season. 21 of these were correctly retrieved in the top-25. The top-25 retrieved clips are presented in Fig. 4.4.

For more information about the clip extraction process, as well as expanded results with different models and in a different clip extraction regime, we refer the reader to supplementary material.

Figure 4.4: Center frames of top-25 retrieved clips from Seinfield Season 2, sorted from left to right and from top to bottom (top left is rank 1, bottom right is rank 25). Correctly retrieved examples have a green border, whilst incorrectly retrieved examples have a red border. Figure best seen in colour.

## 4.6 Conclusion

We have shown how CLIP can be modifed for person-centric retrieval using a face image as a form of prompt engineering for a free-form text query. The retrieval performance has been demonstrated on public datasets for celebrities. However, the same method could be applied to search personal image and video collections in order to find images of your father, say, in a particular place or doing a particular action.

More generally, this idea of prompt engineering using an image can be extended beyond faces to a particular instances of objects, for example a particular building or a particular car. The prompt does not necessarily have to be an image either – we could equally prompt the model using other modalities such as audio in order to add instance recognition in text-to-audio retrieval [Oncescu et al. 2021].

### Acknowledgements

We would like to thank the reviewers and the area chairs of BMVC2022 for their insightful questions and suggestions. This research is supported by of the EPSRC programme Grant VisualAI EP/T028572/1 and Royal Society Research Professorship RP/R1/191132.

## 4.7 Supplementary material

### 4.7.1 Visual queries for unknown faces

The results show that our model performs well, even for the faces it hasn't necessarily seen during training time. In this section, we clarify the relationships between different training datasets and give examples of succesful (and unsuccessful) retrieved and classified examples for both the Celebrities in Places (CiP) [Zhong et al. 2016a] and Celebrities in Action (CiA) dataset.

#### CiP

Celebrities in Places contains images for celebrities from the VGGFace dataset [Parkhi et al. 2015] (noted as *seen* in [Zhong et al. 2016a], as the backbone CNN is trained on VGGFace), and celebrities from other popular face recognition datasets (unseen). The retrieval network in [Zhong et al. 2016a] is also trained on a synthetic dataset for scene retrieval, and *might have* seen the unseen face there, but the face has not been explicitly labeled. A particularly curious aspect of their model is the fact that the network does better in faces-only retrieval on unseen categories than on the seen ones. This has been attributed in [Zhong et al. 2016a] to a much more discriminative face descriptor that is obtained during their training.

We similarly train our model on VGGFace, but we omit training on the synthetic dataset, as CLIP [Radford et al. 2021] has good zero-shot performance on scene retrieval as-is (see table 2 in the main body of the paper). Unlike [Zhong et al. 2016a], who initialise their retrieval model randomly, the CLIP model has seen 400M images, and quite likely some celebrities from the unseen faces too. We show examples of *unseen* faces that our model can retrieve correctly (recall@5), and some it cannot in figure 4.5

#### CiA

Similarly, only around 37% of annotated cast members from the test set are present in VGGFace2 [Cao et al. 2018a], and this increases to 76% when the faces in the CiA training set are included. Thus the network should not have seen around 24% of the test set celebrities during training. We show classification results from table 4 in the main body, broken down by 'seen' (103) and 'unseen' (32) celebrity

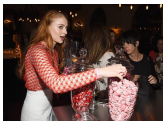
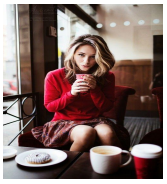
| Query | Tgt. Frame Rank | Tgt. Frame |
|---|---|---|
| Correct | | |
| Aamir Khan in the kitchen | 3 |  |
| Adam Driver in the desert | 5 |  |
| Sophie Turner in the banquet | 3 |  |
| Incorrect | | |
| Andrea Pirlo in the kitchen | 24 |  |
| Goran Visnjic in the supermarket | 13 |  |
| Zoe Levin in the coffee shop | 7 |  |

Figure 4.5: An example of correctly (within top-5) and incorrectly retrieved examples from the CiP dataset [Zhong et al. 2016a], where the person was *not* explicitly seen during training. For each example, we provide a rank at which they were retrieved.

categories in table 4.5.

The main takeaway from this table is the fact that unseen faces benefit disproportionately from the addition of the visual query.

| Model | Seen | Unseen |
|---|---|---|
| random | 0.9 | 3.1 |
| *text* | 31.3 | 11.1 |
| *img* | 35.8 | 22.7 |
| *text + img* | 36.7 | 24.4 |

Table 4.5: Performance of our model for person classification on the CiA dataset test set, evaluated on classes that have been seen or unseen during training (not accounting for CLIP [Radford et al. 2021] pre-training). Numbers are given in % accuracy.

### 4.7.2 Real world retrieval example

In order to present a real-world example of personalised retrieval, we apply our model on the entire Season two of the Seinfeld TV show. We want to see how many occurrences of "Michael Richards entering the room" we can correctly retrieve. Specifically, we form our query with the above text and a single randomly selected query image from the top-100 Google face images. See Fig. 1 of the main paper for illustration.

From each episode, we extract two 2 second long video clips each minute (16 frames per clip at 8fps, at : 00 and : 30 timestamp of each minute). For ground truth, the authors watched the Season, noting all occurrences satisfying the query. If an occurrence happens outside of the given time frames, we added additional video clip extracted with the same settings to the pool of clips. For 11 episodes, we extracted a total of 483 video clips. By our count, Richards is portrayed entering the room 26 times.

The performance of the model can be viewed in Table 4.6. Note that Michael Richards (the actor in the role of Kramer) has *not* been seen during training of the model which is not fine-tuned as he is not included in VGGFace or VGGFace2 datasets, however we are still able to localise him in 16 out of top-25 retrieved example.

**But what if we had more clips?** We try to push this setup by extracting 4 two-second video clips for each minute at the same frame rate as above (at :00, :15, :30 and :45 timestamps of each minute). This yields 994 total clips. Bear in mind that in this scenario, determining positive examples becomes a challenge; for example, authors note that Kramer enters the room at 7:17 in episode 9, however

one could argue that he's barley visible in the doors at 7:16. Would that count as a positive example, even if we were not exceptionally precise about it? With that in mind, our results in this scenario still show promise. In the top-50 examples, we retrieve 15 out of 26 positives, with additional 6 clips that could be considered a close-positive.

Figure 4.6: Center frames of top-25 retrieved clips from Seinfield Season 2 with the best model fine-tuned on Seinfeld cast. Frames are sorted from left to right and from top to bottom (top left is rank 1, bottom right is rank 25). Correctly retrieved examples have a green border, whilst incorrectly retrieved examples have a red border. Best viewed in colour.

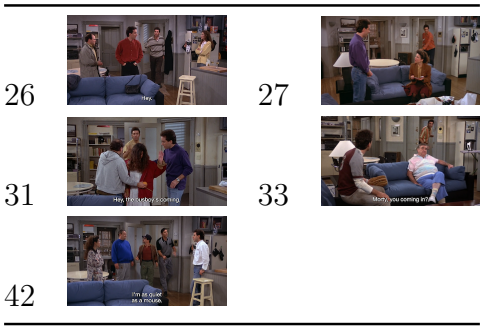



Figure 4.7: Center frames of examples retrieved outside the top-25 with a corresponding rank to the left of the frame. Note that all of these examples feature Richards' character in the background, totally obscured or potentially out of context.

**What if the model sees the characters?** Only one of the Seinfeld characters from Season two is present in either of the training sets we fine tune the CiA model on. To improve our chances, we additionally fine tune the model with images containing Seinfeld characters scraped from Google images (500 per character) with a fixed learning rate of $2e-5$ for 3 epochs. This method unsurprisingly

| Fine-tuned | 2 clips per min | | | 4 clips per min | | |
|---|---|---|---|---|---|---|
| (c)2-7 | R@25 | R@50 | last rank | R@25 | R@50 | last rank |
| no | 0.61 | 0.92 | 53 | 0.46 | 0.58 | 113 |
| yes | 0.84 | 1.0 | 42 | 0.72 | 0.88 | 72 |

Table 4.6: Quantified retrieval results from our Seinfeld experiment. Note that the model without fine-tuning has not seen Michael Richards as a character during training for person-awareness.

yields the best results as seen in table 4.6. The center frame of the top 25 retrieved results can be seen in figure 4.6, while the examples missed in the top-25 and their corresponding rank can be seen in figure 4.7. In the top-25 retrieved examples we count 21 occurrences of Richards entering the room, and all were retrieved in the top 50.

### 4.7.3 Celebrities in Action

In this section, we go more in-depth about our Celebrities in Action (CiA) dataset. We present further data-collection details, show the most common failure cases, and propose further improvements. We show several annotated examples in an external HTML gallery attached.

### 4.7.4 Data collection and annotation

To recap, we automatically annotate the video clips from the Hollywood2 [Marszałek et al. 2009] and High-Five [Patron-Perez et al. 2010] datasets with the person performing the action using the automatic video annotator by Brown *et al.* [Brown et al. 2021b].

We scrape 200 images for every cast member automatically found using the IMDB cast list to obtain a common face embedding for each cast member. Faces are then detected and tracked in each clip at 5 frames-per-second, and then an identity is associated with the face track if it is classified as one of the known actors from that film.

In the case where multiple face tracks with different identities are detected in the scene, we select the one the model is most confident in. Given that the number of training images for person classification is completely balanced, we argue that high confidence for a face track would signify the most dominant (or the clearest) face within the video clip.

If a face-track is not found, we discard the clip. For the test set only, if the face-track is not classified with high confidence (over 0.5), we discard that clip from our consideration. In total, we discard 153 video clips from the dataset.

We manually verify the label correctness on a randomly selected 100 clips from the test set and find the actor annotations to be correct for 97 of them (i.e. annotated actors are visible in the video clip and are performing an action class associated with the video clip).

For clips taken from the High-Five dataset, we additionally annotate them with a high-level place attribute from a ResNet-18 [K. He et al. 2016] pre-trained on Places365 [B. Zhou et al. 2017] dataset and manually verify correctness.

We separate the Hollywood2 training set into the training and held-out validation set (for model development) – not according to films but rather according to the clips in the training data. The Hollywood2 test set remains intact, other than clips discarded as noted above. All clips from the High-Five dataset are added to the test set.

Note that not all video clips contain annotations for 'action' and 'scene'. In total, there are 884 clips in our test set annotated with an actor and an action, 576 of them have a scene annotation attached. In total, the dataset contains 135 actor classes, 12 action classes and 10 scene classes. When results are reported, we only report results on the appropriate set of clips: e.g. when reporting results on 'scene' or 'action + scene' retrieval, we retrieve the examples from the appropriate clip-set.

**Annotation failure cases**

As our dataset is largely automatically annotated, we naturally observe some label noise. In this section, we discuss the Hollywood2 dataset noise and the three most common failure cases. In the last paragraph, we propose different ways to improve the dataset in the future.

**1. Innate scene annotation dataset noise.** Parts of Hollywood2 are automatically annotated, which inherently introduces noise into the dataset. The scene settings are potentially overlapping (would a driveway be considered house-exterior, road, or both?), how to distinguish a hotel room or a bedroom? Would a 12-year-old scene classifier be able to correctly distinguish between them? We find that for the test set, where data has been manually cleaned, these concerns are minimal. If an example is confusing, a scene category is not assigned to it. We do have to acknowledge the noise in the training data, however, in the main body of the paper we show that CLIP-PAD performs well despite the noise.

**2. The wrong actor was annotated.** On average, there is more than one (annotated) actor in each video clip, however, for a final annotation, we only select the most confident one. This may naturally lead to erroneous cases as the best-seen actor might not necessarily perform an action corresponding to the annotation for that video clip (e.g. **figure from the main paper**). Furthermore, more "famous"

actors might also have higher quality images available on Google images, hence leading to better training data for the automatic annotation algorithm. In the example above, Kevin Spacey is both the more famous, and clearer of the two actors in the scene, hence this issue is clearly prominent.

In our preliminary quality control, these issues are rare (2 in 100) due to a strong prior that the actor performing the action is the better seen and/or more famous of the actors in the scene.

**3. Ambiguous annotation.** Hollywood2 dataset contains multiple actions that require *interaction* [Marszałek et al. 2009]: hugging, kissing, handshaking and fighting. For all of these, there can be more than a single correct answer. For example in "Bruce Almighty", Jim Carrey is hugging Jennifer Aniston – whilst both actors are technically correct, only the first one would be accepted as correct.

In our preliminary quality control, we found this to be a common occurrence (16 out of 100),

**4. Low recognition confidence and false positives.** On average, the named cast of a film contains >200 people, but we only have a limited number of clips and people from each film represented in our data. We go through an effort of manually annotating each video clip with the name of the film it belongs to. In this way, the automatic face annotator only has to choose between the cast of that particular film, but our classification accuracy is still low. In the HTML example gallery, the name is associated with the generated video if the recognition confidence is higher than 0.9, and these clips are relatively rare. We observe that only 195 clips (out of over 1500) are annotated with such high confidence.

This means that whilst being mostly correct, our model is not fully confident in its predictions. In "It's a Wonderful Life" for examples, more than one actor in 3 clips is annotated as 'James Stewart', none with high confidence. In the sample gallery the reader can see that whilst our predictions are correct, confidence is not necessarily high. This is often due to the domain difference between the images sourced from Google Image Search and the character's appearance in the film.

Although we do not find this issue to be concerning (as we only select the most confident annotation which we find correct in 99 out of 100 clips we've looked at

manually), improvement in recognition confidence would aid our dataset overall.

**Future improvements.** We argue that even in the initial release, CiA presents a thorough benchmark for compound retrieval of actions on video. There are future improvements that could reduce the noise and further address the failure cases above. The optimal way of addressing these issues would be manual annotation which is an expensive and time-consuming solution.

One option would be the classification of scenes in automatically annotated and not annotated video clips using a modern scene classifier. This would potentially introduce additional categories to the data, but it would increase the variety of the dataset. It would also require additional rounds of manual annotations to keep the test set completely noise-free.

We could also optimise the video auto-annotator [Brown et al. 2021b] for our films. The results can be improved by manually cleaning the automatically-scraped Google images, or by running multiple iterative rounds of automated annotations. Specifically, we could discard all cast members that we know are not in the selected clips, gather additional data on those that are, and re-annotate the lot until the annotations (and their confidence) change no more. This would hopefully reduce the number of false positives and low-confidence classifications.

Even without the potential improvements, we believe our dataset is already a formidable benchmark for various compound retrieval scenarios.

### 4.7.5 Additional experiments

Bellow, we expand on the existing experiments.

### 4.7.6 Further benchmarks on CiA

Despite the existing benchmarks, compound image retrieval is a fairly unexplored task. Similarly, our method is the first one disambiguate between the more "traditional" text-to-video retrieval and our proposed task of compound query video retrieval.

Furthermore, most video-retrieval methods that we are aware of encode video and text in separate streams or do not allow querying with a multi-modal query.

| Method | Text | Image | R@1 | R@5 |
|---|---|---|---|---|
| CE [Yang Liu et al. 2019] | ✓ | | 32.1 | 62.5 |
| DE [Dong et al. 2019] | ✓ | | 35.1 | 66.2 |
| CLIP [Radford et al. 2021] (0-shot) | ✓ | | 39.9 | 72.0 |
| CE [Yang Liu et al. 2019]* | ✓ | ✓ | 39.1 | 71.5 |
| DE [Dong et al. 2019]* | ✓ | ✓ | 39.9 | 71.8 |
| CLIP-PAD* | ✓ | ✓ | 57.8 | 78.1 |
| CLIP-PAD | ✓ | ✓ | 66.3 | 82.7 |

Table 4.7: Additional results on our CiA dataset. [Dong et al. 2019] and [Yang Liu et al. 2019] have been finetuned using default parameters. [Dong et al. 2019] does not have a dedicated face embedding module which might impact its performance negatively. Models using only text effectively perform text-to-video retrieval, while models using text and image combine a visual query (an image of the actor) with the text query – see last paragraph in sec. 4 of the main paper for details on query formation). Note that models marked with "*" use the two-stage querying process as described in text.

However, we apply two modern retrieval methods [Yang Liu et al. 2019; Dong et al. 2019] for which code is available, and compare it further to our model. Specifically, we compare the performance for the 'action + place' metric on the CiA dataset using uni-modal and multi-modal queries as outline below. To retrieve the correct clip based on a text query only, we feed in the textual query in a form "*person* doing *action*" (where *person* and *action* are defined in the data) to each method's respective text encoder and use that to find the most similar video clip. As these models are not capable of multi-modal retrieval, when using the combination of text and image, methods marked with '*' in Table 4.7 are using a two-stage process. In the first stage, we rank the clips according to the similarity to the textual query and the similarity to the visual query (image of the target actor, processed by the visual encoder provided by the respective methods). In the second stage, we take the intersection of the two ranked lists by considering the top $n$ elements of each list, where $n >= k$, until $k$ common elements are found. In order to make the comparison fairer (as our model is capable of using multi-modal queries), we also apply our model in the same fashion and denote the result as 'CLIP-PAD*'. Note that *(a)* there is a clear benefit of a compound query compared to the intersection of ranked lists, and *(b)* our multi-modal method outperforms all others still.

| Method | High five | | | Hollywood 2 | | | CiA |
|---|---|---|---|---|---|---|---|
| | high-fiving (40) | hugging (48) | kissing (43) | kissing (103) | get out of car (57) | drive car (102) | all classes |
| CLIP | 49.1 | 47.5 | 54.9 | 55.1 | 81.8 | 85.0 | 73.1 |
| CLIP-PAD | 57.3 | 55.6 | 59.4 | 59.9 | 83.0 | 85.3 | 75.7 |

Table 4.8: Breakdown of action classification performance per-class for select classes. For evaluation, we follow the same protocol as in Table 4 of the main body of the paper, using only text as a query.

### 4.7.7 Looking closer at action classification

We notice that some action classes can be recognised with significantly less accuracy than others. As we can see in table 4.8, examples coming from the High-five dataset tend to score lower on that particular benchmark. We note that the number of examples is not as indicative of a performance nor is there a major distribution shift between the datasets (for example, 'kissing' has equal performance for examples coming from High-five and examples coming from Hollywood2).

# Chapter 5

# Personalizing Retrieval using Joint Embeddings; or "the Return of Fluffy"

*The paper has been accepted at CBMI2025.*

In Chapter 4, we personalise the text model to recognize people and identities. In this chapter, we extend this work in two meaningful ways. First, from a technical perspective, we keep the text model untrained. This further reduces the degradation of semantic expressivity in Chapter 4. Second, from a practical perspective, the model is now able to embed any arbitrary concept (such as 'author's phone', 'supervisor's car') as opposed to names only.

# Personalizing Retrieval using Joint Embeddings; or "the Return of Fluffy"

Bruno Korbar Andrew Zisserman

Visual Geometry Group, University of Oxford

March 29, 2026

**Abstract**

The goal of this paper is to be able to retrieve images using a compound query that combines object instance information from an image, with a natural text description of what that object is doing or where it is. For example, to retrieve an image of 'Fluffy the unicorn (specified by an image) on someone's head'. To achieve this we design a mapping network that can 'translate' from a local image embedding (of the object instance) to a text token, such that the combination of the token and a natural language query is suitable for CLIP style text encoding, and image retrieval. Generating a text token in this manner involves a simple training procedure, that only needs to be performed once for each object instance. We show that our approach of using a trainable mapping network, termed $\pi$-map, together with *frozen* CLIP text and image encoders improves on the state-of-the-art for three standard retrieval benchmarks designed to assess personalized retrieval.

## 5.1 Introduction

Large-scale pre-trained vision-language models (VLMs) alleviated the need for training task-specific models due to their emerging capability for both intra- and cross-modal retrieval. By enforcing the alignment of text and images, these models allow us to classify objects and scenes, retrieve relevant images given a textual description, and even spatially locate specific objects in an image. However, in practical uses, we are often interested in searching for a specific "thing" in an

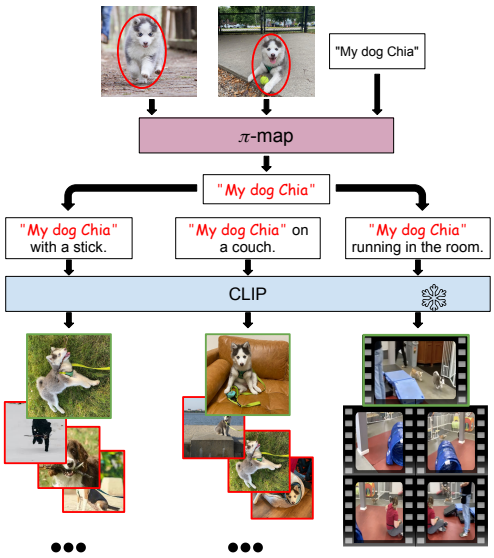



Figure 5.1: Given a few template images, corresponding concept named in text ('"My dog Chia"), our model learns a personalised embedding for the concept. This embedding can then be used in addition to text to search amongst a dataset of similar images or within video frames.

image. On our phones, we may have hundreds of images of dogs, but we may only be interested in one specific dog – our dog "Chia". Searching our library for "My dog Chia with a stick", since the VLMs have no knowledge of our dog Chia, might return either a generic dog or, for example, chia seeds. But what if we want to 'teach' a VLM what "my dog Chia" refers to? Given the name of the dog and a few template images, can we 'teach' a VLM to recognise our dog? In prior work, this problem has been referred to as the "personalization" of VLMs [Cohen et al. 2022; Yeh et al. 2023].

The great advantage of achieving this personalization is that we then can deploy the compositional power of the VLM, and search for "my dog Chia" carrying out various activities and in different environments simply by writing our query as a natural language sentence, as illustrated in Fig 5.1. Our approach is inspired by the language models almost infinite expressively; given a specific-enough query, the large language model used in most popular VLMs *should* be able to synthesise information necessary for better text-to-image retrieval. Therefore, one could argue that the task of personalization might be expressed as learning that 'my dog Chia' corresponds to 'an adorable[1] 4-month old blue-eyed husky mix with grey inverse mask and white socks and features, about 10 inches high'.

Our approach trains a 'translation' network that can map from a few example

[1] not strictly relevant

template images of the object of interest to a suitable text embedding. The text embedding is then used in query sentences for the personalized search for that object. We are not the first to attempt this, and our solution builds on those of others, but our method has (i) fewer requirements than prior work, and (ii) has superior retrieval performance to prior work.

In terms of requirements: we are able to use *frozen* CLIP image and text encoders (whereas previous work fine-tuned the text encoder [Korbar and Zisserman 2022b]); and by using a *local image embedding* we require fewer and less diverse training images than prior work for the personalization – avoiding the failing of learning the context of the image background rather than the foreground object of interest [Cohen et al. 2022]. Furthermore, we leverage a LLM's expressivity to automatically generate caption augmentations in the language domain. Also, unlike previous work [Yeh et al. 2023], training does not require retrieval from a large dataset, so it is efficient.

In terms of performance, we demonstrate superior retrieval performance compared to previous methods over **three** standard benchmark datasets: this-is-my' [Yeh et al. 2023], 'Celebrities in Action' [Korbar and Zisserman 2022b], and 'DeepFashion2' [Y. Ge et al. 2019; Cohen et al. 2022].

## 5.2 Related Work

**Methods for translating between image and text embeddings.** Translation between the modalities of VLMs is a well explored topic related to the task of personalization. Mokady *et al.* [Mokady et al. 2021] show that a single mapping network can translate encoding from images to the text model. They fully finetune the text encoder. Alayrac *et al.*propose training adapter models that map a visual input to the LLM domain using a model dubbed 'Perceiver Resampler'. With such mappings, they only train adapter layers within a LLM [Alayrac et al. 2022]. Li *et al.* [J. Li et al. 2023a] devise an even more efficient model ('Q-former') and a two-stage training method that translates any arbitrary large vision transformer into the domain of LLMs with no need for additional adapter layers. These methods have became a de-facto choice for tasks such as retrieval [Alayrac et al. 2022; Gorti et al. 2022; Dzabraev et al. 2021], and for visual question answering [S. Li et al.

2022; A. Yang et al. 2022b; Alayrac et al. 2022].

An inherent discrepancy between the text and image embeddings has also been a subject of extensive study. Nukrai *et al.*show that noise injection during the CLIP training process helps alleviate the 'modality gap' [Nukrai et al. 2022]. Schrodi *et al.*show that this modality gap can be attributed to as little as two dimensions within each embedding [Schrodi et al. 2024].

**Test-time adaptation.** The task of test-time adaptation (TTA) and various fine-tuning approaches are closely related to the personalization of VLMs. The goal of TTA is to leverages the unlabeled data that arrives at test time by adapting either the forward pass or parameters of the model according to some proxy task [Saenko et al. 2010; Alfarra et al. n.d.]. While in the task of personalization we aim to preserve model's capabilities and only specialise it for one or two instances, the task of TTA generally requires a distribution shift of an entire model. Zhao *et al.* [S. Zhao et al. 2024] show that VLMs can be adapted to out-of-distribution samples using reinforcement learning from CLIP's feedback. Gao *et al.* [P. Gao et al. 2024] show that a feature adapter can replace the need for fine-tuning VLMs. Wortsman *et al.* [Wortsman et al. 2022] present a robust method of fine-tuning VLMs to adapt to the test time data.

**Personalization methods for Joint Embedding Retrieval.** Korbar and Zisserman [Korbar and Zisserman 2022b] have explored how VLMs textual encoder can be augmented to associate a given face embedding with the corresponding name and use either interchangeably to retrieve relevant videos. This method relies on having strong face embeddings and is, therefore, limited to the domain of faces. Wang *et al.* [Zifeng Wang et al. 2022] demonstrated that expert embeddings from [Korbar and Zisserman 2022b] can be replaced by a method that finds the closest generic prompt embedding to the novel class. They learn an 'expert' prompt which is a function of the generic prompt. They focus on novel class discovery rather than on learning instance-specific attributes. Cohen *et al.* [Cohen et al. 2022] proposed extending VLM's language encoder's vocabulary with a newly learned token which represents a specific instance. Their method assumes a clean, manually annotated dataset of specific instances, which are seldom available. Yeh *et al.* [Yeh et al. 2023] learn a database of common traits of a given

category (a process they dub "meta-personalization") and then learn a specific personalised embedding as a weighted combination of global category features. While this approach does not need a large example database (as general-category objects can be discovered automatically), it is limited to the number of common category traits it can store.

**Compound retrieval.** In compound text-to-image retrieval – retrieval over multiple semantic axes, the focus is on specificity over each axis. Ventura *et al.*developed a large-scale compound retrieval benchmark collected automatically by mining web-video captions [Ventura et al. 2024]. Zhong *et al.* [Zhong et al. 2016b] present a compound retrieval image dataset containing the axis 'person' and 'scene', while Korbar and Zisserman [Korbar and Zisserman 2022b] present a video benchmark containing the axis of 'person', 'action', and 'scene'.

## 5.3 Method

In this section, we first outline our personalised retrieval method, and then describe what we dub a 'Personalised Image Embedding Mapping' model (PIE-map or $\pi$-map for short) and its training procedure. Finally, we compare our approach in more detail to the related work.

**Overview of retrieval and personalized training:** Given a few example (template) images of the object of interest, a personalized embedding (a text token) is obtained by carrying out a short training procedure with $\pi$-map. This only needs to be done once. Once we have a personalised embedding, e.g. corresponding to 'My dog Chia', we can form a query (e.g. 'My dog Chia playing in the park') simply by combining the token for 'My dog Chia' with the tokens for the words 'playing in the park'. The tokens of this sentence are then processed by the CLIP text encoder, and the resulting embedding is used to retrieve a relevant image(s) from a large dataset by ranking with their dot product.

To train the personalised embedding for 'My dog Chia' (and an image-to-text mapping $\pi$-map), we first obtain a localized image embedding (of Chia) from the example images. Training then proceeds by *(a)* learning a mapping from the image embedding to the text embedding, whilst also ensuring that *(b)* that embedding

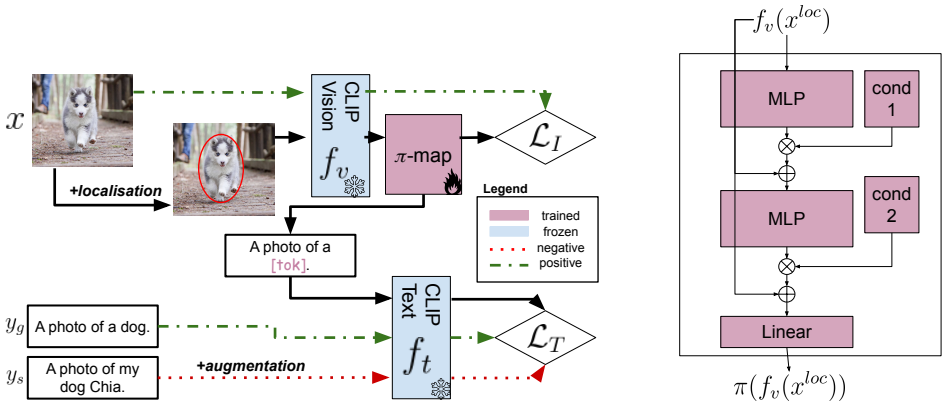



Figure 5.2: Training and architecture details. (a) An overview of how a personalized text token is produced for an object of interest (here the cute dog) from an image, and it is mapped to a textual query. This query is trained to be similar to the detailed description of an image, effectively learning to 'pull-out' this information from the image and translate it into text. (b) Architecture details of the $\pi$-map network.

is 'distinct' amongst all other embeddings containing the general category (dogs). The overview of the training method can be seen in Fig. 5.2.

### 5.3.1 Modelling

The goal of the model is to bridge the gap between the image local embedding output and text embedding input whilst maintaining the instance visual information. We do this through careful architecture design and a two-step training procedure.

**$\pi$-map architecture:** The core idea of our model is to transform the visual embedding into a text 'word'. To project an image to the text embedding, we use a three-layer MLP with residual connections, and two learnable embedding vectors used for conditioning the outputs of each MLP layer. We multiply each conditioning vector from the output of the first and second MLP layers respectively.

The conditioning vectors aim to guide the model to *adjust* the image to focus on the dimension of most disagreement to the text embedding. [Schrodi et al. 2024] notes that two dimensions completely separate the image and text embeddings. In other words, if we were to remove the one or two most significant components from either the text or the image vectors, the model would be unable to distinguish text from image vectors. If we amplify these two components, the model's focus shifts to them in a similar way an attention mechanism would [Vaswani et al. 2017a]. To

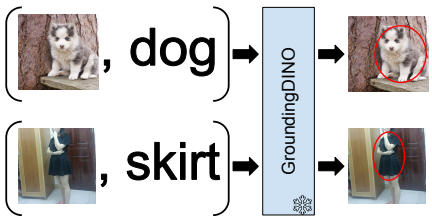
this end, we multiply the outputs of each MLP layer with conditioning vectors that contain higher values for the most relevant component of the embedding, before the final linear projection.

**Notation:** Let $x$ be a template image. Let $y_s$ be a text token denoting a specific instance ('my dog Chia'), $y_g$ text denoting a generic object category ('dog'), and $\mathbb{D}$ a dataset of image-text pairs. Let $f_t(\dot{)}$ and $f_v(\dot{)}$ be text and image encoders of CLIP VLM respectively [Radford et al. 2021]. Our model is denoted as $\pi$.

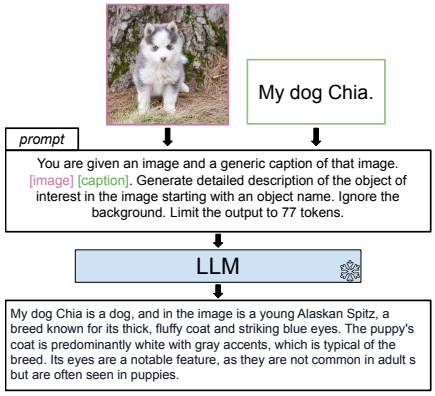
**Localised embeddings are more robust:** By definition, $f_v(x)$ is a global embedding. Therefore, it is sensitive to the image background and context. Say all photos of 'my dog Chia' come from a forest. The model would then be biased to all images of a dog in a forest and might completely miss 'Chia' in the street (an example of the former can be seen in the supplementary). We propose alleviating this issue by using $f_v(x^{loc})$ – a localised version of the embedding.

There are multiple ways to localise CLIP embeddings. Shtedritski *et al.* [Shtedritski et al. 2023] demonstrated that drawing a red ellipse around the area of interest focuses the semantic embedding to the region within it, and Sun *et al.* [S. Sun et al. 2023] showed that it visually augmented image can be used as a localised embedding for downstream tasks. Inspired by this work, we obtain an image with a red 'circle' (an ellipse) by using a pre-trained language-guided detector to detect objects in the image [S. Liu et al. 2023]. An illustration can be seen in Fig 5.3*a*. We empirically demonstrate that adding a red circle around the instance to the template images during training increases the overall performance and reduces the number of template images we need for forming a personalised embedding.

**Reducing information imbalance increases performance:** An image is worth 1000 words. Reducing the caption to 'a photo of a dog' drives the information imbalance between an image and the caption. Schrodi *et al.* [Schrodi et al. 2024] notes that a practical way of dealing with such imbalance on downstream tasks is to increase the information density of the captions. We augment the caption $y_s$ automatically by passing an image $x^{loc}$ and a prompt (see illustration in Fig. 5.3) to the large language model [R. Team et al. 2024] to form a dense text description $y*$. The effect of caption augmentation can be seen in Tbl. 5.1.




(a) Localisation of target object. [S. Liu et al. 2023] returns bounding box coordinates and an ellipse is fitted to it.




(b) Caption augmentation using an LLM [R. Team et al. 2024]

Figure 5.3: Example pipelines of localising the image embedding and caption augmentation.

### 5.3.2 Training

$\pi$-map undergoes two phases of training. The first one is general, i.e. we perform it once and keep it fixed throughout, and we refer to it as pre-training. The goal of the first stage is to initialise the model to bring the two modalities closer together. The second stage is the personalization stage. The goal of the personalization stage is to obtain a specific token for template images of the object of interest.

**Pre-training** In order to initialise model weights, we pre-train $\pi$-map on ImageNet by minimising symmetric cross-entropy loss (following [Radford et al. 2021]) between $f_v(x)$ and $f_t(\pi(f_v(x))$. Intuitively, we are trying to 'teach' *pi*-map to map an image of a 'dog', to that of the text encoding of 'dog'.

**Personalization training** The goal of personalization training is to learn a unique embedding for each specific object ('my dog Chia'). We first give a higher-level overview, and then specify the exact losses later in the section. An outline of our training pipeline is illustrated in Fig. 5.2 on the left.

We start with a template image of 'My dog Chia' $x$ on which we draw a red circle around the dog to obtain $x^{loc}$. The pair is then passed through CLIP visual encoder to obtain global embedding $f_v(x)$ and a local embedding $f_v(x^{loc})$. The local embedding is mapped to text input using $\pi$-map to obtain $\pi(f_v(x^{loc}))$ which becomes a basis of our personalised token. While most of the training is done in the text domain, we do not want $\pi(f_v(x^{loc}))$ to collapse to an encoding of a word

'dog'. Therefore we keep a regularization loss $\mathcal{L}_i$ which keeps $\pi(f_v(x^{loc}))$ and $f_v(x)$ close. Formally we use contrastive loss formulation:

$$\mathcal{L}_i(\pi(f_v(x^{loc}))) = -log(\frac{d(\pi(f_v(x^{loc})), f_v(x))}{\sum_{n \neq x \in \mathbb{B}} d(\pi(f_v(x^{loc})), f_v(n)}$$

where $\mathbb{B}$ is a randomly sampled training minibatch, a distance metric is given by $d(a, b) = exp(\frac{a^T b/\tau}{||a^T||_2||b||_2/\tau})$, and $\tau = 0.07$ is a temperature hyperparameter. Intuitively, the regularisation loss $\mathcal{L}_i$ ensures that the projected embedding $\pi(f_v(x^{loc}))$ does not drift from the original embedding and thus retains its semantic information.

Three text prompts are then formed. A generic text prompt ("A photo of a dog", $y_g$), a specific text prompt ("My dog Chia", $y_s$) and a text-prompt formed by taking "A photo of *" and replacing encoding of "*" with $\pi(f_v(x^{loc}))$. $y_s$ is then *augmented* with an LLM [R. Team et al. 2024] to form an augmented token $y*$ ("My dog chia is an alaskan spitz..."). These text embeddings $y_g$, $y*$, and $\pi(f_v(x^{loc}))$ are then passed through CLIP text encoder $f_t$. Since we are only training a single embedding, we want to specialise it by making it close to $y*$ (learning the semantic correspondence to the detailed information), while making it less sensitive to the more general class $y_g$ (therefore forcing our model to extract more specialised information). We achieve that by optimising a contrastive objective $\mathcal{L}_t$ which ensures that $f_t(\pi(f_v(x^{loc}))$ is similar in the embedding space to, $f_t(y*)$ while being away from $f_t(y_g)$ and all other examples in the batch.

$$\mathcal{L}_t(\pi(f_v(x^{loc})), y*) = -log\frac{d(f_t(\pi(f_v(x^{loc})), f_t(y*))}{\sum_{ni,nt \neq x, y* \in \mathbb{B}} \sum_{nt \neq y_s \in \mathbb{N}} d(f_t(\pi(f_v(ni)), f_t(nt))}$$

where $\mathbb{N}$ is a set of negative examples comprising all other specific and all generic captions in $\mathbb{B}$.

We then compute the total loss as

$$\mathcal{L} = (1 - \alpha)\mathcal{L}_t + \alpha\mathcal{L}_i$$

where $\alpha = 0.25$ is loss balancing hyperparameter determined through line search.

### 5.3.3 Querying the model

If the image is given as a part of the prompt, it is first processed using $\pi$-map and then appended to the rest of the text. If multiple template images were given as queries, their embedding is averaged after they are encoded with $f_v$, before entering $\pi$-map. This embedding is then joined with a text prompt and passed through the text encoder. We measure the similarity between the prompt embedding and the dataset of visual features following [Radford et al. 2021]. An overview can be seen in Fig 5.1.

### 5.3.4 Discussion: relation to previous methods

Compared to the CLIP-PAD approach of [Korbar and Zisserman 2022b], we do not train the language encoder but instead use frozen versions of CLIP's image and language encoders, training only a separate module for the image-to-text translation mapping $\pi$-map. This is a great advantage as $\pi$-map can simply be 'plugged in' to existing deployments of CLIP for retrieval.

Both PALVARA method of [Cohen et al. 2022] and personalization approach of [Yeh et al. 2023] learn direct text-replacement from images; [Cohen et al. 2022] from set encoding, and [Yeh et al. 2023] by learning a linear combination of known features. This means that *(a)* during the personalization stage, both of these are limited to learning from concepts they already 'know'. [Yeh et al. 2023] can only represent instances that can be expressed by the linear combination of their meta categories and [Cohen et al. 2022] only personalises tokens learned from object detection datasets. $\pi$-map can on another hand be trained on top of an arbitrary CLIP model directly as its pre-training stage is general. It also means that *(b)*, querying using an image token directly is impossible as [Yeh et al. 2023] requires a text prompt and [Cohen et al. 2022]'s prompts are fixed after the personalization stage. Because we learn direct mapping from image to text, any image can be used directly as a query by mapping it into the text embedding space.

### 5.3.5 Implementation details

**VLM details:** For fair comparison with prior work ([Yeh et al. 2023; Cohen et al. 2022]), we use OpenAI's CLIP (VIT-B/16) [Radford et al. 2021]. In the

supplementary, we show that the model can function with various models provided by [Ilharco et al. 2021].

**Initialise *pi*-map's conditioning vectors:** We found that initialisation of conditioning vectors matters. To initialise the vectors, we first compute the image embeddings of template images and corresponding text caption, compute the absolute pairwise difference between embedding dimensions, and finally find the two dimensions with the maximum abs. difference. We use this difference vector, zeroing out the largest and second largest dimension respectively, before taking the softmax to initialise the first and second vectors respectively. The illustration of our model can be seen in Fig 5.2 on the right. The effect of this initialisation scheme can be seen in Table 5.1.

**Localisation details:** To obtain localised image $x^{loc}$, we pass the original image and its general category to a pre-trained language-guided detection model GroundingDINO ('GroundingDINO-B') [S. Liu et al. 2023]. For a text prompt, GroundingDINO returns coordinate of the bounding boxes of the object. We superimpose an ellipse onto the image that passes through a centre of each side of the bounding box.

**Caption augmentation details:** In order to augment the caption, we forward the prompt defined in Fig 5.3 *b)* and feed it to the REKA-Core model [R. Team et al. 2024].

**Training details:** We pre-train the model using a batch size of 256 and a learning rate of $3e-4$ for 10 epochs. For personalization, we train the model for 50 epochs on 'this-is-my' and 'CiA' datasets, and for 80 epochs on 'DeepFashion2' with a cosine annealing learning rate starting at $1e-4$ with 200 steps of linear warmup. All training is done with AdamW optimiser. In practice, a training run for learning 15 personalised tokens on 'this-is-my' training set takes about 54 minutes, or 3.6 minutes per-token on a single A4000 chip.

**Extension to videos:** Keen-eyed readers would have noticed that $x$ has thus far been described as an image, but two of the datasets contain video training data. For such cases, we sub-sample a training query video to 10 frames uniformly, localise the specific instance if applicable, and encode the frames using visual encoder $f_v$. We then average embeddings to get $f_v(x)$ and $f_v(x^{loc})$.

**From $\pi$-map to the text embedding:** In order to use text as direct query, we append a personalised prompt to the CLIP tokenizer, initialising its embedding to be an average mapping of template images. The tokenizer then retrieves the learned embedding for the text concept.

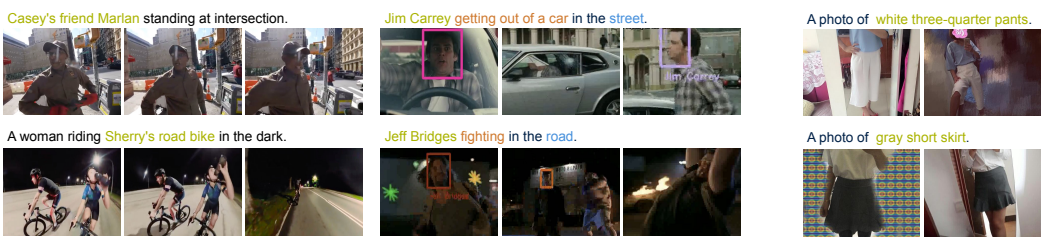
## 5.4 Datasets and Evaluation Measures

In this section, we describe the three datasets used for evaluating our personalization method, as well as the evaluation measures used for each of them. An example of each dataset can be seen in Figure 5.4.

### 5.4.1 This-is-my

Yeh *et al.*proposed 'this-is-my' [Yeh et al. 2023] for personalised text-to-*video* retrieval. The dataset consists of 104 training segments, 683 evaluation segments, and 30 test segments annotated with ten general categories (e.g. 'dog') and 15 specific categories (e.g. 'my dog Biscuit'). We use it for method development and downstream performance evaluation.

**Evaluation procedure:** In order to evaluate our model, we extract CLIP image features from 30 test segments, uniformly sampling frames at 1fps following the prescribed protocol in [Radford et al. 2021; Ilharco et al. 2021]. We embed the textual query using CLIP, with its tokenizer trained to recognise each of the 30 instances. We define similarity for a given video as the maximum dot product between the query and all video features. For the *generic* setting ('An image of *') and report mean average precision (mAP) and mean reciprocal rank (MRR)




(a) 'this-is-my' dataset [Yeh et al. 2023]. (b) CiA dataset [Korbar and Zisserman 2022b]. (c) DeepFashion2 dataset [Y. Ge et al. 2019].

Figure 5.4: Examples from our evaluation datasets.

following [Yeh et al. 2023]. For the *context*ualised setting ('A photo of * in the car park'), there is only one correct match, hence w report MRR and recall-at-5 (R@5). For a fair comparison with SOTA, once the training hyperparameters are set, we train the model on both the train and eval set as [Yeh et al. 2023] train their model on both. In the cases where image is used as a query, we sample a random 5 frames from held-out training videos for that concept to generate the query.

### 5.4.2 Celebrities in Action

Celebrities in Action [Korbar and Zisserman 2022b] is a compound retrieval dataset containing 2668 video clips from 69 movies separated into training, evaluation, and test clips. Each clip is annotated with an actor's name, action performed, and place ('Celebrity in place doing something'). We personalise a model on the training set and evaluate it directly on the test set. Note that some of the more famous celebrities (e.g. Tom Cruise) have almost certainly been seen in CLIP's training set, hence polluting the results. For more details, please see [Korbar and Zisserman 2022b].

**Evaluation procedure:** We follow the same evaluation procedure as above, but we report R@1 and R@5 in the 'action+place' setting (i.e. retrieving a clip using a template '*actor* in a *place* performing an *action*'). R@1 and R@5 are defined following [Alayrac et al. 2022], i.e. R@1=1 if a correct instance is the top-retrieved result. R@5=1 if the correct instance is contained in the top-5 retrieved examples.

### 5.4.3 DeepFashion2

Cohen *et al.* [Cohen et al. 2022] proposed a modified version of DeepFashion2 [Y. Ge et al. 2019] for personalization purposes. They curated a dataset of 653 training and 221 evaluation images that have assigned one of 50 [CONCEPT] tags: e.g. 'a white skirt', 'a short dress', etc. For the evaluation images, they collect in-context captions such as 'The [CONCEPT] is facing a glass store display' (short caption) or 'White cabinets, some with open drawers, are alongside and behind the [CONCEPT]' (long caption). Overall, 50 total concepts are contained in the dataset.

**Evaluation procedure:** As DeepFashion2 is an image dataset, we simply encode each image with a CLIP visual model, and follow the same evaluation protocol as for the 'this-is-my' dataset otherwise. We follow the benchmark setting from [Yeh et al. 2023] and use five images to train the embedding.

## 5.5 Results

In this section, we present the results of our method. Sec 5.5.1 presents various ablation studies taken into account while designing the model. We then compare our trained model with state-of-the-art (SOTA) on the datasets described in Sec 5.4: 'this-is-my' [Yeh et al. 2023] (in Sec. 5.5.2), 'Celebrities in Action' [Korbar and Zisserman 2022b] (in Sec 5.5.3), and 'DeepFashion2' [Y. Ge et al. 2019] (in Sec 5.5.4) datasets. Finally, we discuss our findings and limitations in Sec. 5.5.5. Qualitative results can be found in Figure 5.5.

### 5.5.1 Ablation study

In this section, we evaluate our design choices on the evaluation section of the 'this-is-my' dataset. As we want to obtain finer-grained insight into our model's performance, we compute what we call true R@5 (or tR@5, defined as a number of correct examples retrieved in top-5 over a number of all positives) and precision-at-5 (P@5; the proportion of positive examples retrieved in top-5). Note that maximum theoretical tR@5=40.3.

Table 5.1 (a) shows that our model significantly outperforms naive CLIP baselines. Table 5.1 (b) explores the effects of our modelling choices discussed in Sec. 5.3 on the downstream performance. It is notable that caption augmentation plays a significant role in achieving good results (+10.9 precision points). While localisation plays only a minor role in the overall result, it allows us to achieve similar results with a lower number of frames (Tbl. 5.2).

### 5.5.2 this-is-my

Results on this-is-my datasets are reported in Table 5.3. Our model using an image (randomly selected and held out from the training set) comfortably outperforms all other methods. It is notable that, although the text encoding is computed

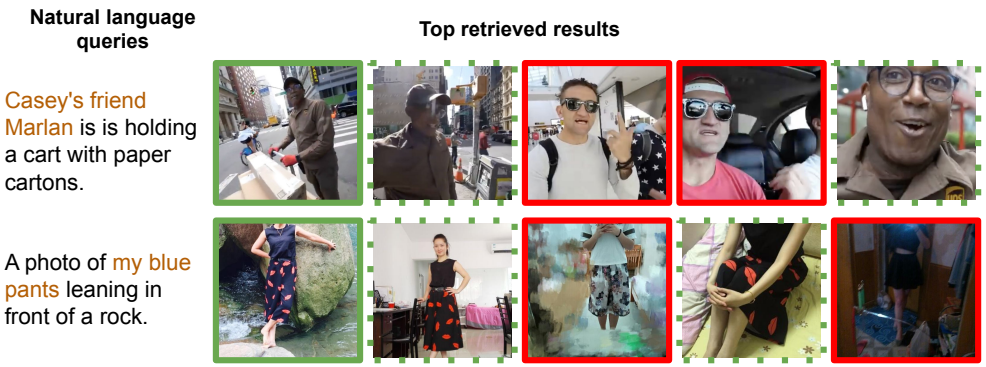



Figure 5.5: A qualitative sample of contextual retrieval sorted from left to right from this-is-my [Yeh et al. 2023] and DeepFashion2 [Y. Ge et al. 2019] datasets. Green and red rectangles correspond to the correctly and incorrectly retrieved segments/images. Dotted green line shows correctly retrieved instances but in wrong setting.

Table 5.1: Ablations on eval split of 'this-is-my' [Yeh et al. 2023] dataset.

(a) Baseline results. Results in grey denote CLIP [Radford et al. 2021] baseline.

| Method | tR@5 (max 40.3) | P@5 |
|---|---|---|
| text – generic | 12.3 | 56.1 |
| text – specific | 11.8 | 58.3 |
| image | 15.3 | 63.5 |
| text + image | 18.1 | 67.7 |
| ours | 33.7 | 87.2 |

(b) Ablating various model components.

| Ablation | tR@5 (max 40.3) | P@5 |
|---|---|---|
| Ours | 33.7 | 87.2 |
| w/o Reg Loss | 29.1 | 79.1 |
| w/o Caption Augmentation | 26.0 | 76.3 |
| w/o Localisation | 31.9 | 83.5 |
| w/o Pre-Training | 27.7 | 77.6 |
| w/o Init-Scheme | 32.9 | 86.2 |

using an average of the template training images, using an image as a query as opposed to the text yields better results.

### 5.5.3 Celebrities in Action

Results on the CiA dataset can be seen in Table 5.4. Our method outperforms the CLIPPad method despite its pre-training on VGGFace datasets. We attribute this to the much more efficient alignment training.

### 5.5.4 DeepFashion2

Our results on DeepFashion2 [Y. Ge et al. 2019] show marginal improvement over previous methods. DeepFashion2 is also the only dataset where caption augmentation did, in fact, cause adverse effects (54.7/78.2 with and 55.1/78.9 w/o). We hypothesise this is due to the relative simplicity of the object (e.g. 'white skirt')

Table 5.2: The performance of the method depends on a number of query images. Using local features reduces the amount of template images necessary. Results on the eval split of 'this-is-my' [Yeh et al. 2023] dataset.

| #template imgs | with localisation | | no localisation | |
|---|---|---|---|---|
| | tR@5 (max 40.3) | P@5 | tR@5 (max 40.3) | P@5 |
| 1 | 31.9 | 84.5 | 28.6 | 77.8 |
| 3 | 33.7 | 87.2 | 30.4 | 81.9 |
| 5 | 33.6 | 87.2 | 31.9 | 83.5 |
| 10 | 33.2 | 87.0 | 32.4 | 84.8 |

Table 5.3: Results on the test set of 'this-is-my' dataset [Yeh et al. 2023]. 'RGB' and 'TXT' columns denote whether an image or text was used as a query. '*' denotes our reproduction of the method.

| Method | IMG | TXT | Context | | Generic | |
|---|---|---|---|---|---|---|
| | | | MRR | R@5 | mAP | MRR |
| CLIP [Radford et al. 2021] | | ✓ | 30.8 | 36.7 | 16.6 | 44.2 |
| CLIP [Radford et al. 2021] | ✓ | ✓ | 20.9 | 23.3 | 51.7 | 82.9 |
| CLIP* [Radford et al. 2021] | ✓ | ✓ | 24.3 | 28.9 | 52.4 | 83,4 |
| Thisismy [Yeh et al. 2023] | | ✓ | 42 | 50.7 | 56.4 | 87.4 |
| Ours | ✓ | | **43.1** | **52.0** | **58.4** | **88.3** |
| Ours | | ✓ | 42.1 | 50.9 | 57.0 | 87.6 |

when compared to more complex descriptions of humans or particular objects in 'this-is-my' and 'CiA' datasets. Full results can be seen in Table 5.5.

### 5.5.5 Discussion and Limitations

We demonstrate that our model learns image-to-text mapping with less examples and achieving higher performance than all other personalization methods. With the ability to use images as queries, it allows us to outperform even methods fully fine-tuned on external datasets (Sec. 5.5.3). In the supplementary material, we

Table 5.4: Results on 'Celebrities in Action' dataset [Korbar and Zisserman 2022b]. 'IMG' and 'TXT' columns denote whether image or text was used as query.

| Method | IMG | TXT | R@1 | R@5 |
|---|---|---|---|---|
| CLIP [Radford et al. 2021] | | ✓ | 39.9 | 72.5 |
| CLIPPad [Korbar and Zisserman 2022b] | ✓ | ✓ | 66.3 | 82.7 |
| CLIPPad [Korbar and Zisserman 2022b] | ✓ | | 64.9 | 81.8 |
| CLIPPad [Korbar and Zisserman 2022b] | | ✓ | 64.2 | 81.1 |
| Ours | ✓ | | **69.7** | **85.2** |
| Ours | | ✓ | 64.9 | 81.2 |

Table 5.5: Results on 'DeepFashion2' dataset [Y. Ge et al. 2019], personalization split as defined by [Cohen et al. 2022]. Note that [Cohen et al. 2022] use ViT-B/32 instead of ViT-B/16. Results in grey denote CLIP [Radford et al. 2021] baseline.

| Method | Context | | Generic | |
|---|---|---|---|---|
| | MRR | R@5 | mAP | MRR |
| txt | 21.2 | 23.4 | 9.0 | 17.5 |
| img | 14.5 | 17.6 | 20.9 | 43.9 |
| img + txt | 21.0 | 26.9 | 21.7 | 43.6 |
| PALVARA [Cohen et al. 2022] | 28.4 | 39.2 | - | - |
| this-is-my [Yeh et al. 2023] | 38.4 | 51.4 | 53.4 | 77.7 |
| Ours (img) | 38.5 | 51.8 | 54.7 | 78.2 |

discuss model's capability to use template images to generalise to unseen classes as well. Our model significantly outperforms all other methods on the common personalization benchmarks, and the possibility of using images directly as queries alleviates this problem completely.

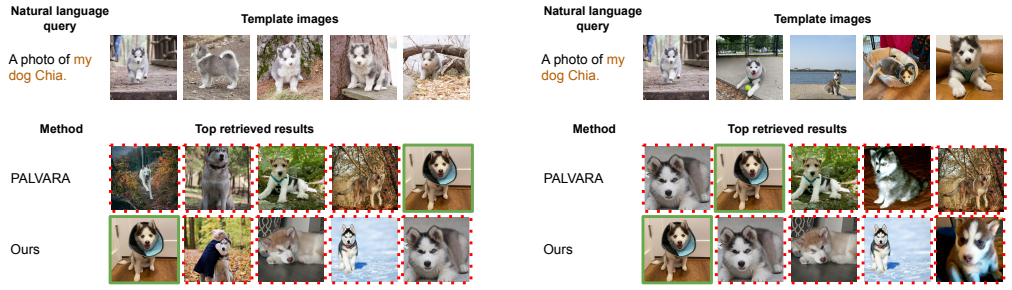
## 5.6 Conclusion

We present a conceptually simple and effective method for learning personalised tokens in VLMs using image-to-text mapping called $\pi$-map. It is highly capable in personalised text-to-image, text-to-video and image-to-image retrieval, outperforming all prior work on three personalization benchmarks while requiring only a few examples to fully personalise the embedding. In future work, we hope to expand on our method and develop a new, larger benchmark for personalised retrieval.

## 5.7 Supplementary material

### 5.7.1 Importance of Local Features

We investigate the quantitative importance of using local features for learning personalised embeddings in Tbl. 1 and Tbl. 2 of the main paper. To demonstrate the importance qualitatively, we learn personalised embedding with 5 different images of the dog 'Chia' using our method and those of PALVARA [Cohen et al. 2022]. We obtain 40 images with Google image search to use as hard negatives (prompts used: 'a dog in a forest' and an 'a small husky in a forest'), and display top-5




(a) (b)

Figure 5.6: Importance of using localised features: learning personalised features for 'My dog Chia' from two different sets of template images. In *a*) all template images come from the same time and place, while in *b)*, the images are varied. Our method ranks the correct image first on both occasions, while PALVARA [Cohen et al. 2022] remains sensitive to the diversity of the template images.

in Figure 5.6. Our method shows resilience to the type of background features, while PALVARA [Cohen et al. 2022] seems to exploit additional biases (such as background) in the images.

### 5.7.2 Invariance to CLIP Models

In the main paper, we present results only using OpenAI's CLIP (VIT-B/16) [Radford et al. 2021] in order to compare fairly with state-of-the-art methods. To demonstrate that our method can work on various CLIP variants pre-trained on different datasets in a 'plug-and-play' fashion, we use fixed setup described in the main body using personalised embeddings without image queries, and experiment with various CLIP variants provided by [Ilharco et al. 2021]. Our results (Table 5.6) demonstrate that our method can be applied in a plug-and-play fashion. Most variants deviate only slightly from the baseline model (within 1.1 recall point on 'this-is-my' and 'CiA' datasets).

### 5.7.3 Out-of-dataset Retrieval

All examples thus far were computed on curated datasets where template images are visually similar to the test data. We wonder if trained embeddings could be used for a more general setting – for example if we would be able to retrieve a video clip containing an actor from a large corpus of 'unseen' movie segments when the template images of the actor come from another source.

To evaluate this property, we personalise embeddings for 10 actors from the CiA

Table 5.6: Exploration of performance using various CLIP variants. All results are computed using text-only queries on the test sets of this-is-my [Yeh et al. 2023], CiA [Korbar and Zisserman 2022b], and DeepFashion2 [Y. Ge et al. 2019] datasets.

| CLIP Variant | **ThisIsMy Context** | | **CiA** | | **DeepFashion2 Context** | |
|---|---|---|---|---|---|---|
| | MRR | R@5 | R@1 | R@5 | MRR | R@5 |
| VIT-B/16 [Radford et al. 2021] | 42.1 | 50.9 | 64.9 | 81.2 | 38.3 | 51.2 |
| ViT-B/32 [Radford et al. 2021] | 41.6 | 50.4 | 64.7 | 81.1 | 38.3 | 51.0 |
| ViT-L/14 [Ilharco et al. 2021] | 42.4 | 51.0 | 65.4 | 81.4 | 38.6 | 51.5 |
| ViT-H/14 [Ilharco et al. 2021] | 42.7 | 51.5 | 65.5 | 81.4 | 38.9 | 52.0 |
| ViT-SO400M/14 (siglip) [Ilharco et al. 2021; Zhai et al. 2023] | 43.4 | 52.0 | 66.0 | 82.3 | 39.4 | 53.7 |

dataset and find clips from the Condensed Movies dataset [Bain et al. 2020b] in which these actors appear (each video clip has manually labelled cast annotations). This leaves us with 697 target video clips. We then ask a large VLM [R. Team et al. 2024; G. Team 2024] to caption these video clips starting with '<actor> is' to obtain video clip-level annotation. We remove clips that overlap with CiA. To avoid potential noise in the data, we also curate our own set of high-quality template images for these actors from `imdb.com`. This is done to avoid potentially noisy or low-quality template images that might arise from sampling in CiA.

Then we use the generated caption as a text-query, and search for the corresponding video clip in the entire dataset.

**Implementation details and metrics:** Visual features are extracted, and search is performed using WISE [Dutta et al. 2024] software (approximate NN-search follows the protocol from [Douze et al. 2024]). We measure actor R@10 and P@10, as well as the clip R@10. Actor R@10= 1 if the actor is present in any of the top-10 retrieved video clips; actor P@10 measures the proportion of video clips containing the target actor in the top-10 retrieved video clips; clip R@10= 1 if the exact video clip is in the top-10 ranked examples. CLIP and CLIPPad used images and text as person-specific queries. Our method uses pre-computed (text) embeddings.

**Results:** We compare a CLIP vision and text benchmark with $\pi$-net's embedding trained from template video clips from CiA as well as a manually curated set of template images. Results can be seen in Tbl 5.7, where we compare the results with zero-shot CLIP retrieval approach [Radford et al. 2021], as well as

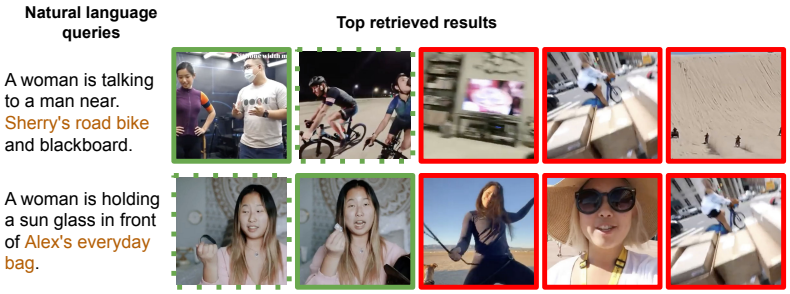
Table 5.7: Quantitative results on 'unseen' retrieval task. We personalise actor embeddings on dataset *a)*, and retrieve clips containing these actors from a disjoint and unseen dataset *b)*. We want to know if we can retrieve the correct video clips despite the dataset gap.

| Method | Training Dset | Actor R@10 | Actor P@10 | Clip R@10 |
|---|---|---|---|---|
| CLIP | NA | 59.6 | 37.5 | 16.8 |
| CLIPPad | VGGFace2 | 78.4 | 55.8 | 32.1 |
| Ours | CiA | 71.8 | 53.2 | 49.1 |
| Ours | Manual | 77.7 | 62.4 | 54.0 |

with the pre-trained CLIPPad model [Korbar and Zisserman 2022b]. We show that our model outperforms the CLIP benchmark by a large margin. The performance of our model almost matches that of CLIPPad – a model trained on a person-identification dataset [Cao et al. 2018b]. Interestingly, although our model retrieves the correct actor less often than CLIPPad (−0.7 R@10 points), if the actor is correctly recognised, it happens with higher precision (+6.6 percentage points). Finally, our model is significantly better at combining textual prompts with personalised embeddings (+21.9 R@10 points). We postulate this is due to our method not training the text-encoder, hence preserving the semantic information of text embedding better.

### 5.7.4 More Qualitative Examples

In Figure 5.7, we present more qualitative examples from the this-is-my [Yeh et al. 2023] and CiA [Korbar and Zisserman 2022b] datasets. A web gallery is attached with examples from the retrieved examples described in Section 5.7.3.




(a)


(b)

Figure 5.7: Additional qualitative examples of retrieval on this-is-my ($a$), and CiA ($b$) datasets. The green frame denotes correctly retrieved instances and captions. The dotted green frame denotes a correctly retrieved instance (e.g. 'Harrison Ford') in an incorrect setting. The red frame denotes incorrectly retrieved examples.

# Chapter 6

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

*The paper has been accepted at ICAASP 2024.*

In this chapter, we employ methods from Chapters 4 and 5, with advances in diarization and speech-to-text to develop a pipeline for named subtitling. Understanding who said what and when is a crucial building block for holistic story-understanding.

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

Bruno Korbar* Jaesung Huh* Andrew Zisserman

* denotes equal contribution

Visual Geometry Group, University of Oxford

March 29, 2026

**Abstract**

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

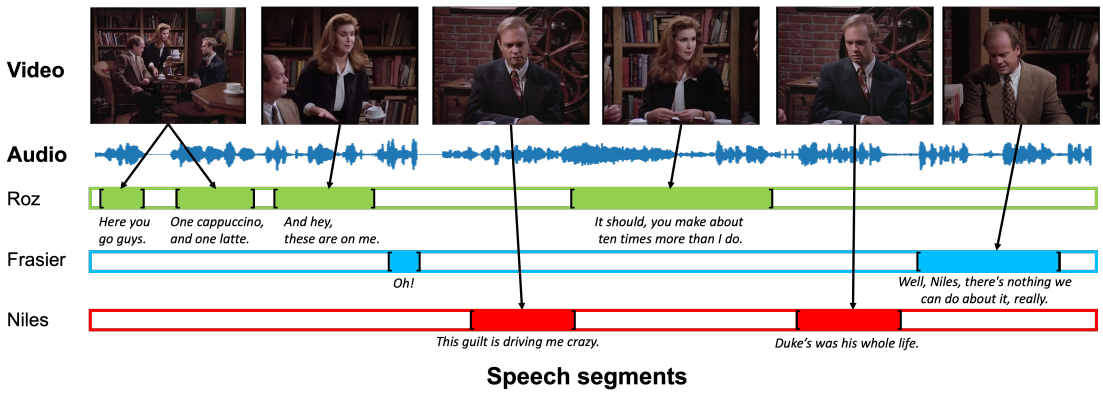



Figure 6.1: Character-aware audio-visual subtitling. The generated data covers *what* is said, *when* it said, and by *whom* it is said.

## 6.1 Introduction

With the rise of streaming platforms that allow watching videos "on-demand", more video content is made available to the general public and researchers than ever in history. With more than 80% of users of one such platform relying on subtitles [*Netflix player control tests* n.d.], automatic subtitle generation and captioning has become an important research topic in the community [Radford et al. 2022; Bain et al. 2023]. Unfortunately, many subtitles, whether automatically generated or not, do not comply with the standards for Subtitles for Deaf and Hard-of-hearing (SDH): namely, they do not include information about speaker identification, nor do they contain sound effects and music.

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

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

We make the following four contributions: (i) we propose a new task, character-aware audio-visual subtitling, which aims to generate the *what*, *when* and by *whom* for subtitles, with minimal required metadata.

(ii) we develop an automatic pipeline for this task that does not require face detection or tracking (Section 6.1.1); (iii) we curate an evaluation dataset that includes subtitles labelled with characters individually for three different sitcom series: Fraiser, Scrubs and Seinfeld (Section 6.2); and (iv) we assess the method on the evaluation dataset and report the performance (Section 6.3).

### 6.1.1 Related work

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

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

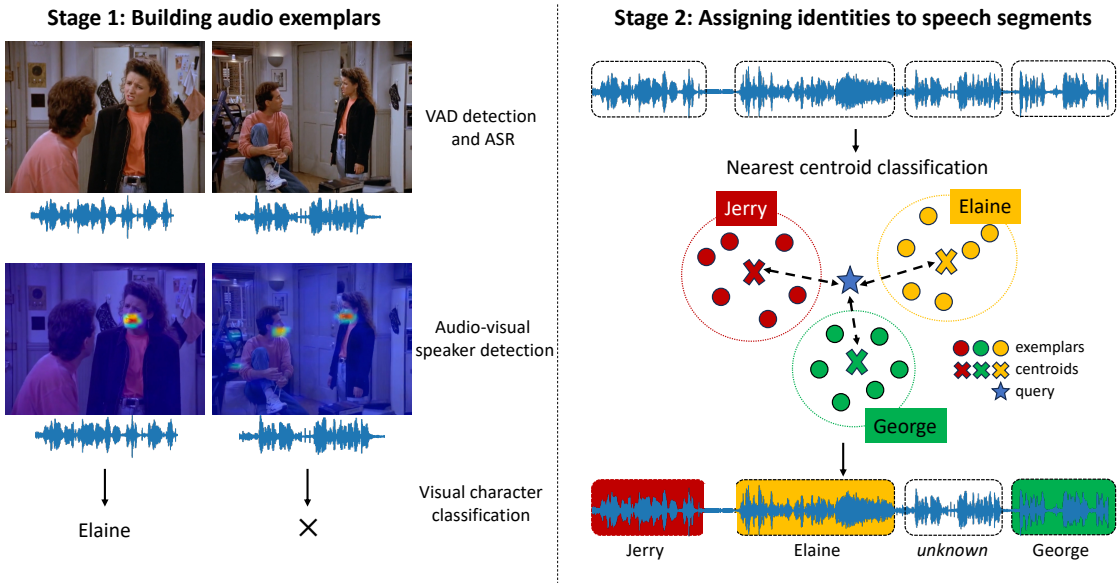



Figure 6.2: Overview of our method. We first build a database of audio exemplars for each character by filtering speech segments until only a high precision set remains (left). Each speech segment is then assigned to a character by comparing its voice embedding to the exemplar embeddings (right).

the character.

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

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

This section explains our approach to creating subtitles for the video and attributing speakers to each speech segment.

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

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

### 6.1.2 Stage 1: building audio exemplars

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

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

We end up with a set of speech segments for each character that are recognised with high precision, and refer to these as *speech exemplars*. The building of these exemplars is illustrated in Figure 6.2, and we give details of each sub-step below.

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

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

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

**2. Audio-visual speaker detection.** The goal of this stage is to take speech segments from the previous stage and select only those with a single visible speaker. This will produce a subset of speech segments where we can recognise the speaker. To achieve this, we localise the speaker with an audio-visual synchronisation model [Afouras et al. 2020] which produces a spatial location of the audible objects and has been shown to detect speakers well. In practice, it generates an audio-guided heatmap over each video frame. We average the heatmaps over the length of each speech segment to avoid unnecessary noise and detect peaks in the heatmap through a combination of maximum filtering and non-maximum suppression. Example heatmap outputs can be seen in Figure 6.2. When a single peak is visible throughout the video clip, we can assume that only one speaker is present. If there are no detected peaks, or there are multiple ones, we discard that speech segment from the candidates of exemplars.

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

Our classification model [Korbar and Zisserman 2022b] compares a visual embedding of the frames of a speech segment to a combination of actor visual embedding and actor name (details are given below). We select the best match or discard

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

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

### 6.1.3 Stage 2: Assigning characters to speech segments

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

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

### 6.1.4 Implementation details

We detect speech and perform ASR with an off-the-shelf WhisperX [Bain et al. 2023] model, and the sentences are tokenized with NLTK [Bird 2006] tokenizer. We use the laughter detector by [Gillick et al. 2021] with a detection threshold of 0.8. All voice embeddings are encoded with ECAPA-TDNN [Desplanques et al.

2020], which is pretrained with VoxCeleb [Nagrani et al. 2019].

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

## 6.2 Evaluation Dataset

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

### 6.2.1 Annotation procedure

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

**Aligning transcripts and timestamps.** To associate character names with corresponding temporal timestamps, we leverage two readily accessible source of textual video annotation: original transcripts and subtitles with word-level timestamps. Transcripts are obtained from multiple online sources [*The Frasier Archives* n.d.; *Seinfeld scripts dot com* n.d.; *Scrubs fandom* n.d.]. They include spoken lines and information about who is speaking. However, they do not provide any

Table 6.1: Evaluation dataset statistics. **# episode**: number of episodes, **duration**: total duration of the dataset, **#IDs**: total number of characters, **speech %**: percentage of video time that is speech and **# spks**: min / mean / max of number of speakers per video.

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

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

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

### 6.2.2 Dataset statistics

Three TV series datasets are used to evaluate our method. We annotate the first six episodes of Season 2 of **Frasier**, Season 2 of **Scrubs** and Season 3 of **Seinfeld**. We utilise the sixth episode in each season as our validation set, while the remaining episodes serve as our test set. The detailed statistics are shown in Table 6.1.

## 6.3 Results

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

Table 6.2: Exemplar yield after steps in Stage 1 (on Seinfeld).

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

Table 6.3: Exemplar recognition performance for named characters in Stage 1 in Seinfeld. 'others' is a group of 21 characters, all named correctly.

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

### 6.3.1 Detailed analysis of Stage 1 and 2

**Performance evaluation of Stage 1.** We evaluate the yield and classification accuracy of the speech exemplars on the five episodes of Seinfeld in our test set. In Table 6.2, it can be seen that **19.3%** of voice activity segments can be considered as exemplars. We also evaluate the performance quantitatively by manually inspecting the exemplars. The results, shown in Table 6.3, demonstrate that the accuracy of Stage 1 is almost perfect, being **100%** correct for most characters. There are 11 characters for which we have no exemplars in the 5 episodes of Seinfeld. They cover only 1.8% of speech segments – most of them speak less than five sentences in the episodes.

**Performance evaluation of Stage 2.** We demonstrate the trade-off between the Proportion of Classified Segments (POCS) and overall precision by varying the threshold $d$ used in the nearest centroid voice classification to assign speech segments as "unknown". True positives are the segments that overlap with the ground truth segments and the character is correctly identified. Figure 6.3 shows the result. It can be seen that precision decreases as we classify more segments. Also, long segments show higher precision in all three TV series at any given POCS, which shows that the speaker model produces better representations for longer segments.

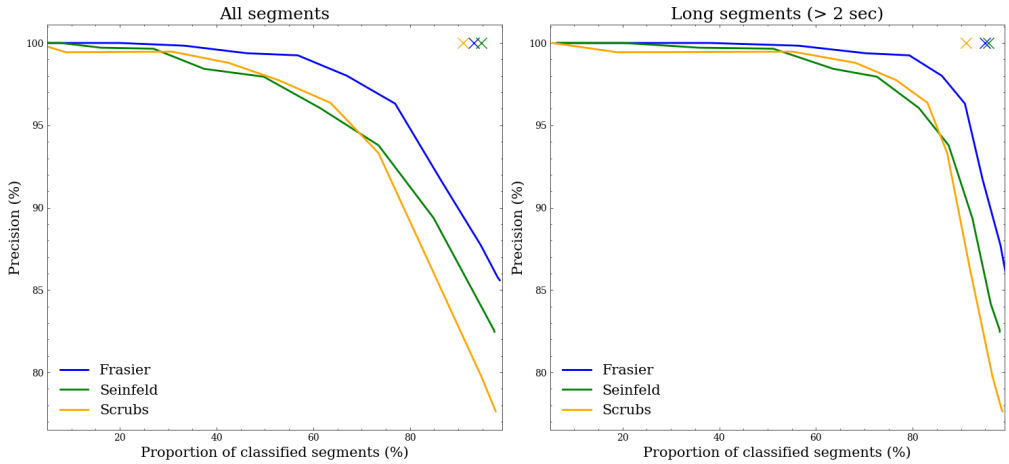



Figure 6.3: Stage 2 Precision-POCS Curves for the test set of the three TV series, obtained by varying the threshold $d$ (for classification as "unknown"). The left figure shows the performance using all detected speech segments. The right figure shows the performance only for the long segments ($> 2$ sec). We also show the oracle points ('x' in each graph) for each TV series. The oracle point is where all segments for which there are character exemplars are correctly classified, and other segments are classified as "unknown".

### 6.3.2 Overall performance on the test set

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

The results are given in Table 6.4. We can see that the model performs best on Frasier and worst on Scrubs in all metrics. This is due to the difference in size of the casts in each dataset. Scrubs has more characters than Frasier ($48 > 29$) for a similar total duration (see Table 6.1). Thus, Scrubs provides more potential assignments for each segment, making identification more challenging.

We also report the diarisation performance with and without the overlapping speech in Table 6.4. The difference in DER for these two categories is small in Seinfeld and Frasier, meaning that there is not much overlapping speech within these two shows.

**Speech transcription performance.** Our method uses the WhisperX ASR model which also produces the speech transcription results. We compare the performance with the state-of-the-art models in Table 6.5. Word Error Rate (WER)

Table 6.4: Performance on the test set. We report the Diarisation Error Rate both with and without consideration of the overlapping regions, **DER(O)** and **DER** respectively. **Acc** denotes a character recognition accuracy for the segments that overlap with the groundtruth. **Ppc** and **Rpc** are the average per-character precision and recall, respectively.

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

Table 6.5: Word Error rate (WER) (%) on each dataset.

| Model | Version | Seinfeld | Frasier | Scrubs |
|---|---|---|---|---|
| Wav2Vec2.0 [Baevski et al. 2020] | `ASR_BASE_960H` | 45.0 | 36.9 | 36.3 |
| Whisper [Radford et al. 2022] | `medium.en` | 13.2 | 13.5 | 10.6 |
| WhisperX [Bain et al. 2023] | `medium.en` | **11.8** | **11.2** | **9.2** |

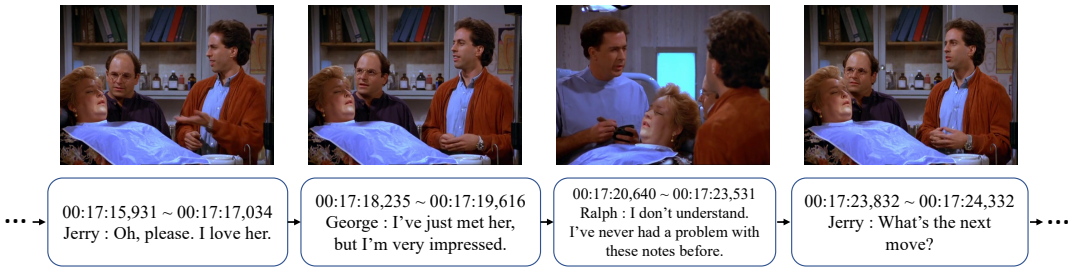



Figure 6.4: Qualitative example. Our method produces the speech segments with timestamps, and assigns the character who spoke it.

is computed after applying the Whisper text normaliser to both ground truth and predictions which can be found in the original paper [Radford et al. 2022]. We see that WhisperX outperforms both Wav2vec2.0 and Whisper. This is because the VAD Cut & Merge preprocessing reduces the hallucination of Whisper, which is also mentioned in the original paper [Bain et al. 2023].

**Qualitative example.** We show a qualitative example of our results in Figure 6.4. As can be seen, our method assigns the character for each speech segment, as well as timestamps and the transcription.

## 6.4 Conclusions

In this work, we show promising first steps towards a model for character-aware subtitling, which we hope will be beneficial for improving accessibility, and facilitating further research in video understanding. Our method is not perfect, however. Our recognition efforts fail on short segments such as exclamations and

also do not deal with overlapping speech – though the latter does not appear to be a serious limitation in practice. Furthermore, to generate the true SDH subtitles, we would need to classify and categorise every sound, not just speech – something our model is not yet capable of.

# Chapter 7

# Discussion

In this chapter, the reader can find a summary of the accomplishments, with an emphasis on the significance of the work for the broader research community. This is followed by potential future directions for this research.

## 7.1 Achievements and Impact

**Open Vocabulary Tracking and Instance-Level Retrieval** In recent literature, multi-object tracking has predominantly relied on class-dependent classifiers. In other words, objects could only be tracked if they could be classified using a trained deep neural network. In Chapter 2, we relax this constraint by using a tracking query that is initialised from the initial object (i.e. the first frame) and then tracked by maintaining similarity to the original object embedding. This allowed us to achieve results comparable to those of class-specific methods on the TAO dataset without the need for a classification loss. Our method, together with that of [Meinhardt et al. 2021], became a foundation for what other methods refer to as 'tracking-by-query' methods [De Plaen et al. 2024; Ruopeng Gao et al. 2024].

Similarly, class-agnostic instance-level retrieval is a relatively new task involving the retrieval of a specific named instance from a larger data corpus. Inaugural works on the topic compute the query for a named instance as a function of known traits of various object categories [Cohen et al. 2022; Yeh et al. 2023]. This limits the queries they can learn to the size of their "category dictionary". Our work uses a pre-trained language model's innate ability to discern fine semantic differences

in text, allowing us to use a seemingly endless vocabulary to learn personalised queries. This then removes the limitation of vocabulary size for personalised retrieval. We believe this line of work will have many practical applications in search and retrieval systems.

**Long Video Processing** In Chapter 3, we explore a task-specific sampling mechanism that enables the use of much longer videos without sacrificing performance in large visual-language models. Most existing visual-language models are limited to processing 8 or 16 frames due to memory constraints [J. Li et al. 2023a; Alayrac et al. 2022]. Our work extends this capability to over 100 frames, allowing us to achieve higher accuracy on semantic tasks compared to previous approaches. This research has facilitated advancements in integrating long videos into corporate machine learning systems.

**Personalisation of Video Understanding** The identity-aware compound retrieval method we presented in Chapter 4 serves as a seminal work in the field of compound video retrieval, and the *Celebrities in Action* dataset introduced in this chapter is rapidly becoming an important benchmark for the community [Baldrati et al. 2023]. Additionally, the technical approach we presented has inspired analogous developments in related fields, such as movie description [T. Han et al. 2023b].

Moreover, our work on audio-visual character identification, as proposed in Chapter 6, has the potential to significantly extend and simplify the automatic annotation of video datasets. This advancement could prove invaluable for gathering conversational datasets tailored to embodied artificial intelligence.

## 7.2 Extensions

**RingAttention for Long-video Understanding** The work presented in Chapter 3, along with other concurrent and related studies, has demonstrated that only a small proportion of video tokens need to be passed from the vision encoder to the large language model [Maaz et al. 2023; J. Li et al. 2023a]. However, a practical limitation remains in terms of memory consumption, which still restricts the absolute number of frames that the model can observe. Recent advancements, such

as RingAttention, hold promise for significantly increasing the context available to samplers like ours, potentially to an almost infinite extent [H. Liu et al. 2023]. This approach would maintain the benefits of the information bottleneck while further expanding the number of frames that can be processed.

**Further Efforts in Personalisation** This thesis presents an overview of what we believe are the most readily available and important tasks for personalisation: complex retrieval tasks and audio-visual diarisation. These are only some of the plethora of video understanding tasks that would benefit from personalisation. For instance, personalizing audio-visual movie description tasks would enable models to automatically generate narratives for individual actors, independent of the scene. Additionally, exploring the personalization of large-scale multi-modal language models directly (i.e., without additional adapter strategies) would represent a logical next step in this domain.

## 7.3 Conclusion

In this thesis, we explore, overcome, and introduce various challenges in making video understanding tasks identity-aware. We begin by addressing technical capabilities in two key areas: tracking (Chapter 2) and long video understanding (Chapter 3). Subsequently, we delve into challenges and develop tasks that enhance machine learning systems' identity-awareness, including retrieval in Chapters 4 and 5, and audio-visual diarisation (Chapter 6).

By overcoming these challenges and introducing new ones for the community to tackle, we contribute a small step towards developing human-like perception capabilities for artificial intelligence.

# References

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

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

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

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Fei-Fei Li, and Silvio Savarese (2016). "Social LSTM: Human Trajectory Prediction in Crowded Spaces". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*.

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

Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman (2020). "BSL-1K: Scaling up Co-articulated Sign Language Recognition Using Mouthing Cues". In: *Proc. ECCV*.

Motasem Alfarra, Hani Itani, Alejandro Pardo, Merey Ramazanova, Juan Camilo Perez, Matthias Müller, Bernard Ghanem, et al. (n.d.). "Evaluation of Test-Time Adaptation Under Computational Time Constraints". In: *Forty-first International Conference on Machine Learning*.

Humam Alwassel, Fabian Caba Heilbron, and Bernard Ghanem (2018). "Action Search: Spotting Actions in Videos and Its Application to Temporal Action Localization". In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Cham: Springer International Publishing.

Anton Andriyenko and Konrad Schindler (2011). "Multi-target tracking by continuous energy minimization". In: *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*.

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

Hexin Bai, Wensheng Cheng, Peng Chu, Juehuan Liu, Kai Zhang, and Haibin Ling (2021). "GMOT-40: A Benchmark for Generic Multiple Object Tracking". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*.

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

Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman (2020a). "Condensed Movies: Story Based Retrieval with Contextual Embeddings". In: *Asian Conf. Comput. Vis.*

Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman (2020b). *Condensed Movies: Story Based Retrieval with Contextual Embeddings*. arXiv: 2005.04208 [cs.CV].

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: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman (2022). "A CLIP-Hitchhiker's Guide to Long Video Retrieval". In: *CoRR*.

Alberto Baldrati, Lorenzo Agnolucci, Marco Bertini, and Alberto Del Bimbo (2023). "Zero-shot composed image retrieval with textual inversion". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Jérôme Berclaz, François Fleuret, Engin Türetken, and Pascal Fua (2011). "Multiple Object Tracking Using K-Shortest Paths Optimization". In: *IEEE Trans. Pattern Anal. Mach. Intell.*

Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé (2019). "Tracking Without Bells and Whistles". In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*.

Keni Bernardin and Rainer Stiefelhagen (2008). "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics". In: *EURASIP J. Image Video Process.*

Gedas Bertasius and Lorenzo Torresani (2020). “Classifying, Segmenting, and Tracking Object Instances in Video with Mask Propagation”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani (2021). “Is space-time attention all you need for video understanding?” In: *ICML*.

Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip HS Torr (2016). “Fully-Convolutional Siamese Networks for Object Tracking”. In: *arXiv preprint arXiv:1606.09549*.

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

Guillem Brasó and Laura Leal-Taixé (2020). “Learning a Neural Solver for Multiple Object Tracking”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*.

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: *Multimedia Information Processing and Retrieval (MIPR)*.

Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman (2020). “Smooth-AP: Smoothing the Path Towards Large-Scale Image Retrieval”. In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IX*. Lecture Notes in Computer Science. Springer.

Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles (2017). “SST: Single-Stream Temporal Action Proposals”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*.

Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles (2022). “Revisiting the" video" in video-language understanding”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

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*.

Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto (2022). "MeMOT: Multi-Object Tracking with Memory". In: *CoRR*.

Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman (2018a). "Vggface2: A dataset for recognising faces across pose and age". In: *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*.

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

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko (2020). "End-to-End Object Detection with Transformers". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*.

Joao Carreira and Andrew Zisserman (2017). "Quo vadis, action recognition? a new model and the kinetics dataset". In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Daozheng Chen, Mustafa Bilgic, Lise Getoor, and David Jacobs (2011). "Dynamic processing allocation in video". In: *TPAMI*.

Guo Chen, Sen Xing, Zhe Chen, Yi Wang, Kunchang Li, Yizhuo Li, Yi Liu, Jiahao Wang, Yin-Dong Zheng, Bingkun Huang, Zhiyu Zhao, Junting Pan, Yifei Huang, Zun Wang, Jiashuo Yu, Yinan He, Hongjie Zhang, Tong Lu, Yali Wang, Limin Wang, and Yu Qiao (2022). *InternVideo-Ego4D: A Pack of Champion Solutions to Ego4D Challenges*. arXiv: 2211.09529 [cs.CV].

Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang (2018). "Real-Time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-Identification". In: *2018 IEEE International Conference on Multimedia and Expo, ICME 2018, San Diego, CA, USA, July 23-27, 2018*.

Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton (2021). "Pix2seq: A language modeling framework for object detection". In: *arXiv preprint arXiv:2109.10852*.

Wongun Choi and Silvio Savarese (2010). "Multiple Target Tracking in World Coordinate with Single, Minimally Calibrated Camera". In: *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*. Lecture Notes in Computer Science.

Peng Chu and Haibin Ling (2019). "FAMNet: Joint Learning of Feature, Affinity and Multi-Dimensional Assignment for Online Multiple Object Tracking". In: *2019*

*IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*.

Peng Chu, Jiang Wang, Quanzeng You, Haibin Ling, and Zicheng Liu (2021). "TransMOT: Spatial-Temporal Graph Transformer for Multiple Object Tracking". In: *CoRR*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. (2022). "Scaling instruction-finetuned language models". In: *arXiv preprint arXiv:2210.11416*.

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

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

Niv Cohen, Rinon Gal, Eli A. Meirom, Gal Chechik, and Yuval Atzmon (2022). ""This is my unicorn, Fluffy": Personalizing frozen vision-language representations". In: *European Conference on Computer Vision (ECCV)*.

Timothee Cour, Chris Jordan, Eleni Miltsakaki, and Ben Taskar (2008). "Movie/script: Alignment and parsing of video and text transcription". In: *European Conference on Computer Vision*. Springer.

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: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi (2023). *InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning*. arXiv: 2305.06500 [cs.CV].

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray (2018). "Scaling Egocentric Vision: The EPIC-KITCHENS Dataset". In: *European Conference on Computer Vision (ECCV)*.

Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg (2017). "ECO: Efficient Convolution Operators for Tracking". In: *2017 IEEE Conference*

*on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré (2022). *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness*. arXiv: 2205.14135 [cs.LG].

Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan (2020). "TAO: A Large-Scale Benchmark for Tracking Any Object". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*.

Pierre-François De Plaen, Nicola Marinello, Marc Proesmans, Tinne Tuytelaars, and Luc Van Gool (Jan. 2024). "Contrastive Learning for Multi-Object Tracking With Transformers". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Mostafa Dehghani, Alexey Gritsenko, Anurag Arnab, Matthias Minderer, and Yi Tay (2022). "Scenic: A JAX Library for Computer Vision Research and Beyond". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

V. Delaitre, J. Sivic, and I. Laptev (2011). "Learning person-object interactions for action recognition in still images". In: *NeurIPS*.

Vincent Delaitre, David F. Fouhey, Ivan Laptev, Josef Sivic, Abhinav Gupta, and Alexei A. Efros (2012). "Scene Semantics from Long-Term Observation of People". In: *Computer Vision – ECCV 2012*. Ed. by Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid. Berlin, Heidelberg: Springer Berlin Heidelberg.

Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian D. Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé (2020). "MOT20: A benchmark for multi object tracking in crowded scenes". In: *CoRR*. URL: https://arxiv.org/abs/2003.09003.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck (2020). "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification". In: *Proc. Interspeech*.

Yifan Ding, Yong Xu, Shi-Xiong Zhang, Yahuan Cong, and Liqiang Wang (2020). "Self-supervised learning for audio-visual speaker diarization". In: *Proc. ICASSP*.

Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang (2019). "Dual encoding for zero-example video retrieval". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang, and Meng Wang (2021). "Dual encoding for video retrieval by text". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou (2024). "The Faiss library". In: arXiv: 2401.08281 [cs.LG].

Abhishek Dutta, Horace Lee, and Prasanna Sridhar (2024). *Wise 2.0*. Version 2.0. URL: https://gitlab.com/vgg/wise/wise.

Abhishek Dutta and Andrew Zisserman (2019). "The VIA Annotation Software for Images, Audio and Video". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: ACM. URL: https://doi.org/10.1145/3343031.3350535.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman (2020). "Counting out time: Class agnostic video repetition counting in the wild". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Maksim Dzabraev, Maksim Kalashnikov, Stepan Komkov, and Aleksandr Petiushko (2021). "Mdmmt: Multidomain multimodal transformer for video retrieval". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem (2016). "DAPs: Deep Action Proposals for Action Understanding". In: *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*.

Mark Everingham, Josef Sivic, and Andrew Zisserman (2006). "Hello! My name is... Buffy"–Automatic Naming of Characters in TV Video." In: *BMVC*.

Mark Everingham, Josef Sivic, and Andrew Zisserman (2009). "Taking the Bite out of Automatic Naming of Characters in TV Video". In: *Image and Vision Computing*.

Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman (2017). "Detect to Track and Track to Detect". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*.

Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan (2010). "Object Detection with Discriminatively Trained Part-Based Models". In: *IEEE Trans. Pattern Anal. Mach. Intell.*

Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe (2019). "End-to-end neural speaker diarization with permutation-free objectives". In: *Proc. Interspeech*.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid (2020). "Multi-modal Transformer for Video Retrieval". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV*.

Jiyang Gao, Zhenheng Yang, Chen Sun, Kan Chen, and Ram Nevatia (2017). "TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao (2024). "Clip-adapter: Better vision-language models with feature adapters". In: *International Journal of Computer Vision*.

Ruohan Gao, Rogério Schmidt Feris, and Kristen Grauman (2018). "Learning to Separate Object Sounds by Watching Unlabeled Video". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*.

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*.

Ruopeng Gao, Yijun Zhang, and Limin Wang (2024). "Multiple Object Tracking as ID Prediction". In: *arXiv preprint arXiv:2403.16848*.

Yuying Ge, Ruimao Zhang, Lingyun Wu, Xiaogang Wang, Xiaoou Tang, and Ping Luo (2019). "A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images". In: *CVPR*.

Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun (2021). "YOLOX: Exceeding YOLO Series in 2021". In: *arXiv preprint arXiv:2107.08430*.

Jon Gillick, Wesley Deng, Kimiko Ryokai, and David Bamman (2021). "Robust Laughter Detection in Noisy Environments." In: *Proc. Interspeech*.

Ross B. Girshick (2015). "Fast R-CNN". In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*.

Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu (2022). "X-pool: Cross-modal language-video attention for text-video retrieval". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Shreyank N Gowda, Marcus Rohrbach, and Laura Sevilla-Lara (2021). "Smart frame selection for action recognition". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik (2022). "Ego4D: Around the World in 3,000 Hours of Egocentric Video". In: *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*.

Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman (2020). "Automatically Discovering and Learning New Visual Categories with Ranking Statistics". In: *International Conference on Learning Representations*.

Shoudong Han, Piao Huang, Hongwei Wang, En Yu, Donghaisheng Liu, Xiaofeng Pan, and Jun Zhao (2020). "MAT: Motion-Aware Multi-Object Tracking". In: *CoRR*.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman (2023a). "AutoAD II: The Sequel – Who, When, and What in Movie Audio Description". In: *ICCV*.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman (2023b). "AutoAD: Movie Description in Context". In: *CVPR*.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman (2024). "AutoAD III: The Prequel – Back to the Pixels". In: *CVPR*.

Tengda Han, Weidi Xie, and Andrew Zisserman (2022a). "Temporal Alignment Networks for Long-term Video". In: *Proc. CVPR*.

Tengda Han, Weidi Xie, and Andrew Zisserman (2022b). "Turbo Training with Token Dropout". In: *Brit. Mach. Vis. Conf.*

Monica-Laura Haurilet, Makarand Tapaswi, Ziad Al-Halah, and Rainer Stiefelhagen (2016). "Naming TV characters by watching and analyzing dialogs". In: *Proc. WACV*. IEEE.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask R-CNN". In: *Int. Conf. Comput. Vis.*

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee (2023). *Flax: A neural network library and ecosystem for JAX*. Version 0.7.5. URL: http://github.com/google/flax.

Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem (2016). "Fast Temporal Activity Proposals for Efficient Detection of Human Actions in Untrimmed Videos". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*.

David Held, Sebastian Thrun, and Silvio Savarese (2016). "Learning to Track at 100 FPS with Deep Regression Networks". In: *CoRR*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell (2018). "Localizing Moments in Video with Temporal Language". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*.

Shota Horiguchi, Yusuke Fujita, Shinji Watanabe, Yawen Xue, and Kenji Nagamatsu (2020). "End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors". In: *Proc. Interspeech*.

Huazhang Hu, Sixun Dong, Yiqun Zhao, Dongze Lian, Zhengxin Li, and Shenghua Gao (2022). "Transrac: Encoding multi-scale temporal correlation with transformers for repetitive action counting". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jie Hu, Li Shen, and Gang Sun (2018). "Squeeze-and-Excitation Networks". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*.

Daoji Huang, Otmar Hilliges, Luc Van Gool, and Xi Wang (2023). *Palm: Predicting Actions through Language Models @ Ego4D Long-Term Action Anticipation Challenge 2023*. arXiv: 2306.16545 [cs.CV]. URL: https://arxiv.org/abs/2306.16545.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt (2021). *OpenCLIP*. Version 0.1. If you use this software, please cite it as below. URL: https://doi.org/10.5281/zenodo.5143773.

*International Movie Database* (n.d.). https://www.imdb.com.

Mihir Jain, Jan C. van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees G. M. Snoek (2014). "Action Localization with Tubelets from Motion". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig (2021a). "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision". In: *arXiv:2102.05918 [cs]*. arXiv: 2102.05918. URL: http://arxiv.org/abs/2102.05918 (visited on 03/30/2022).

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig (2021b). "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*.

Hanwen Jiang, Santhosh Kumar Ramakrishnan, and Kristen Grauman (2023). "Single-Stage Visual Query Localization in Egocentric Videos". In: *arXiv preprint arXiv:2306.09324*.

Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie (2021). "Prompting Visual-Language Models for Efficient Video Understanding". In: *CoRR*.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion (2021). "Mdetr-modulated detection for end-to-end multi-modal understanding". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev,

et al. (2017). “The Kinetics Human Action Video Dataset”. In: *arXiv preprint arXiv:1705.06950*.

Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M. Rehg (2015). “Multiple Hypothesis Tracking Revisited”. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*.

Dohwan Ko, Joonmyung Choi, Juyeon Ko, Shinyeong Noh, Kyoung-Woon On, Eun-Sol Kim, and Hyunwoo J Kim (2022). “Video-Text Representation Learning via Differentiable Weak Temporal Alignment”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Bruno Korbar, Du Tran, and Lorenzo Torresani (2019). “Scsampler: Sampling salient clips from video for efficient action recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Bruno Korbar and Andrew Zisserman (2022a). “End-to-end Tracking with a Multi-query Transformer”. In: *arXiv preprint arXiv:2210.14601*.

Bruno Korbar and Andrew Zisserman (2022b). “Personalised CLIP or: how to find your vacation videos”. In: *Brit. Mach. Vis. Conf.*

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles (2017). “Dense-Captioning Events in Videos”. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton (2012a). “Imagenet classification with deep convolutional neural networks”. In: *NeurIPS*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton (2012b). “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*.

Anna Kukleva, Makarand Tapaswi, and Ivan Laptev (June 2020). “Learning Interactions and Relationships Between Movie Characters”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Weicheng Kuo, A. J. Piergiovanni, Dahun Kim, Xiyang Luo, Ben Caine, Wei Li, Abhijit Ogale, Luowei Zhou, Andrew Dai, Zhifeng Chen, Claire Cui, and Anelia Angelova (2023). *MaMMUT: A Simple Architecture for Joint Learning for MultiModal Tasks*. URL: http://arxiv.org/abs/2303.16839 (visited on 10/11/2023).

Youngki Kwon, Hee Soo Heo, Jaesung Huh, Bong-Jin Lee, and Joon Son Chung (2021). "Look who's not talking". In: *2021 IEEE Spoken Language Technology Workshop (SLT)*.

I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld (2008). "Learning realistic human actions from movies". In: *Proc. CVPR*.

Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler (2016). "Learning by Tracking: Siamese CNN for Robust Target Association". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2016, Las Vegas, NV, USA, June 26 - July 1, 2016*.

Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese (2014). "Learning an Image-Based Motion Context for Multiple People Tracking". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*.

Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn (2011). "Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker". In: *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*.

Jie Lei, Tamara L. Berg, and Mohit Bansal (2021a). "QVHighlights: Detecting Moments and Highlights in Videos via Natural Language Queries". In: *CoRR*.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu (2021b). "Less Is More: ClipBERT for Video-and-Language Learning via Sparse Sampling". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg (2018). "TVQA: Localized, Compositional Video Question Answering". In: *EMNLP*.

Paul Lerner, Juliette Bergoënd, Camille Guinaudeau, Hervé Bredin, Benjamin Maurice, Sharleyne Lefevre, Martin Bouteiller, Aman Berhe, Léo Galmant, Ruiqing Yin, et al. (2022). "Bazinga! A Dataset for Multi-Party Dialogues Structuring". In: *LREC*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi (2023a). "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models". In: *ICML*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi (2023b). *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. URL: http://arxiv.org/abs/2301.12597 (visited on 10/11/2023).

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi (2022). "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation". In: *ICML*.

Shuang Li, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, and Igor Mordatch (2022). "Composing ensembles of pre-trained models via iterative consensus". In: *arXiv preprint arXiv:2210.11522*.

Yanwei Li, Chengyao Wang, and Jiaya Jia (2023). *LLaMA-VID: An Image is Worth 2 Tokens in Large Language Models*. arXiv: 2311.17043 [cs.CV].

Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang (2018). "BSN: Boundary Sensitive Network for Temporal Action Proposal Generation". In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick (2014). "Microsoft COCO: Common Objects in Context". In: *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*.

Hao Liu, Matei Zaharia, and Pieter Abbeel (2023). *Ring Attention with Blockwise Transformers for Near-Infinite Context*. arXiv: 2310.01889 [cs.CL].

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. (2023). "Grounding dino: Marrying dino with grounded pre-training for open-set object detection". In: *arXiv preprint arXiv:2303.05499*.

Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman (2019). "Use What You Have: Video retrieval using representations from collaborative experts". In: *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, 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 (2022). "CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning". In: *Neurocomputing*.

Liqian Ma, Siyu Tang, Michael J. Black, and Luc Van Gool (2018). "Customized Multi-person Tracker". In: *Computer Vision - ACCV 2018 - 14th Asian Conference*

*on Computer Vision, Perth, Australia, December 2-6, 2018, Revised Selected Papers, Part II*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan (2023). *Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models*. arXiv: 2306.05424 [cs.CV].

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik (2023). "EgoSchema: A Diagnostic Benchmark for Very Long-form Video Language Understanding". In: *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Marcin Marszałek, Ivan Laptev, and Cordelia Schmid (2009). "Actions in Context". In: *IEEE Conference on Computer Vision & Pattern Recognition*.

Effrosyni Mavroudi, Triantafyllos Afouras, and Lorenzo Torresani (2023). "Learning to Ground Instructional Articles in Videos through Narrations". In: *arXiv preprint arXiv:2306.03802*.

Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixé, and Christoph Feichtenhofer (2021). "TrackFormer: Multi-Object Tracking with Transformers". In: *CoRR*.

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: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*.

Antoine Miech, Ivan Laptev, and Josef Sivic (2018). "Learning a Text-Video Embedding from Imcomplete and Heterogeneous Data". In: *arXiv*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic (2019a). "HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips". In: *ICCV*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic (2019b). "HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips". In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*.

Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler (2016). "MOT16: A Benchmark for Multi-Object Tracking". In: *CoRR*. URL: http://arxiv.org/abs/1603.00831.

Bogdan Mocanu, Ruxandra Tapu, and Titus Zaharia (2019). "Enhancing the accessibility of hearing impaired to video content through fully automatic dynamic captioning". In: *2019 E-Health and Bioengineering Conference (EHB)*.

Ron Mokady, Amir Hertz, and Amit H Bermano (2021). "ClipCap: CLIP Prefix for Image Captioning". In: *arXiv preprint arXiv:2111.09734*.

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 and Andrew Zisserman (2017). "From Benedict Cumberbatch to Sherlock Holmes: Character Identification in TV series without a Script". In: *Brit. Mach. Vis. Conf.*

Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell (2021). "CLIP-It! Language-Guided Video Summarization". In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*.

*Netflix player control tests* (n.d.). https://about.netflix.com/en/news/player-control-tests, 2023.

Edwin G. Ng, Bo Pang, Piyush Sharma, and Radu Soricut (2021). "Understanding Guided Image Captioning Performance across Domains". In.

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, A. Sophia Koepke, Joao F. Henriques, Zeynep Akata, and Samuel Albanie (2021). "Audio Retrieval with Natural Language Queries". In: *INTERSPEECH*. Annual Conference Series. isca-speech.

Aljosa Osep, Wolfgang Mehner, Paul Voigtlaender, and Bastian Leibe (2018). "Track, Then Decide: Category-Agnostic Vision-Based Multi-Object Tracking". In: *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*.

Rameswar Panda, Chun-Fu Richard Chen, Quanfu Fan, Ximeng Sun, Kate Saenko, Aude Oliva, and Rogerio Feris (2021). "Adamml: Adaptive multi-modal learning for efficient video recognition". In: *Proceedings of the IEEE/CVF international conference on computer vision*.

Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu (June 2021). "Quasi-Dense Similarity Learning for Multiple Object Tracking". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman (2015). "Deep Face Recognition". In: *Brit. Mach. Vis. Conf.*

Mandela Patrick, Po-Yao Huang, Yuki Markus Asano, Florian Metze, Alexander G. Hauptmann, João F. Henriques, and Andrea Vedaldi (2021). "Support-set bottlenecks for video-text representation learning". In: *9th*

*International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Alonso Patron-Perez, M. Marszałek, Andrew Zisserman, and Ian D. Reid (2010). "High Five: Recognising Human Interactions in TV Shows". In: *Brit. Mach. Vis. Conf.*

Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool (2009). "You'll never walk alone: Modeling social behavior for multi-target tracking". In: *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*.

Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes (2011). "Globally-optimal greedy algorithms for tracking a variable number of objects". In: *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*.

Jesús Andrés Portillo-Quintero, José Carlos Ortiz-Bayliss, and Hugo Terashima-Marín (2021). "A Straightforward Framework for Video Retrieval Using CLIP". In: *Pattern Recognition - 13th Mexican Conference, MCPR 2021, Mexico City, Mexico, June 23-26, 2021, Proceedings*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. (2021). "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever (2022). "Robust speech recognition via large-scale weak supervision". In: *ICML*.

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*.

Santhosh Kumar Ramakrishnan, Ziad Al-Halah, and Kristen Grauman (2023). "SpotEM: Efficient Video Search for Episodic Memory". In.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*.

Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi (2016). "Performance Measures and a Data Set for Multi-target, Multi-camera

Tracking". In: *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*.

Ergys Ristani and Carlo Tomasi (2018). "Features for Multi-Target Multi-Camera Tracking and Re-Identification". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*.

Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Chris Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele (2017). "Movie Description". In: *International Journal of Computer Vision*.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell (2010). "Adapting visual category models to new domains". In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer.

Pramod Sankar, C. V. Jawahar, and Andrew Zisserman (2009). "Subtitle-Free Movie to Script Alignment". In: *Brit. Mach. Vis. Conf.*

Simon Schrodi, David T Hoffmann, Max Argus, Volker Fischer, and Thomas Brox (2024). "Two Effects, One Trigger: On the Modality Gap, Object Bias, and Information Imbalance in Contrastive Vision-Language Representation Learning". In: *arXiv preprint arXiv:2404.07983*.

Paul Scovanner and Marshall F. Tappen (2009). "Learning pedestrian dynamics from the real world". In: *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*. IEEE Computer Society.

*Scrubs fandom* (n.d.). https://scrubs.fandom.com/wiki/Category:Transcripts.

*Seinfeld scripts dot com* (n.d.). https://www.seinfeldscripts.com/seinfeld-scripts.html.

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: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*.

Jiayi Shao, Xiaohan Wang, Ruijie Quan, and Yi Yang (2023). "Action Sensitivity Learning for the Ego4D Episodic Memory Challenge 2023". In: *arXiv preprint arXiv:2306.09172*.

Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun (2018). "CrowdHuman: A Benchmark for Detecting Human in a Crowd". In: *arXiv preprint arXiv:1805.00123*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut (2018). "Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning". In: *Proceedings of ACL*.

Rahul Sharma and Shrikanth Narayanan (2022). "Audio visual character profiles for detecting background characters in entertainment media". In: *arXiv preprint arXiv:2203.11368*.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer (2022). "How Much Can CLIP Benefit Vision-and-Language Tasks?" In.

Bowen Shi, Wei-Ning Hsu, and Abdelrahman Mohamed (2022). "Robust self-supervised audio-visual speech recognition". In: *Proc. Interspeech*.

Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang (2017). "CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*.

Zheng Shou, Dongang Wang, and Shih-Fu Chang (2016). "Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*.

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi (2023). "What does CLIP know about a red circle? Visual prompt engineering for VLMs". In: *IEEE International Conference on Computer Vision*.

Bing Shuai, Andrew Berneshawi, Xinyu Li, Davide Modolo, and Joseph Tighe (2021). "SiamMOT: Siamese Multi-Object Tracking". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Karen Simonyan and Andrew Zisserman (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang (2023). *MovieChat: From Dense Token to Sparse Memory for Long Video Understanding*. arXiv: 2307.16449 [cs.CV].

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah (2012). "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild". In: *CRCV-TR-12-01*.

Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo (2020). "TransTrack: Multiple-Object Tracking with Transformer". In: *CoRR*.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao (2023). "EVA-CLIP: Improved Training Techniques for CLIP at Scale". In: *arXiv preprint arXiv:2303.15389*.

Shuyang Sun, Runjia Li, Philip Torr, Xiuye Gu, and Siyang Li (2023). *CLIP as RNN: Segment Countless Visual Concepts without Training Endeavor*. arXiv: 2312.07661 [cs.CV].

Reuben Tan, Matthias De Lange, Michael Iuzzolino, Bryan A Plummer, Kate Saenko, Karl Ridgeway, and Lorenzo Torresani (2023). "Multiscale Video Pretraining for Long-Term Activity Forecasting". In: *arXiv preprint arXiv:2307.12854*.

Ran Tao, Efstratios Gavves, and Arnold W. M. Smeulders (2016). "Siamese Instance Search for Tracking". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler (2016). "MovieQA: Understanding Stories in Movies Through Question-Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Graham W. Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler (2010). "Convolutional Learning of Spatio-temporal Features". In: *Computer Vision - ECCV 2010 - 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI*. Ed. by Kostas Daniilidis, Petros Maragos, and Nikos Paragios. Lecture Notes in Computer Science. Springer. URL: https://doi.org/10.1007/978-3-642-15567-3%5C_11.

Gemini Team (2024). *Gemini: A Family of Highly Capable Multimodal Models*. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

Reka Team, Aitor Ormazabal, Che Zheng, Cyprien de Masson d'Autume, Dani Yogatama, Deyu Fu, Donovan Ong, Eric Chen, Eugenie Lamprecht, Hai Pham, Isaac Ong, Kaloyan Aleksiev, Lei Li, Matthew Henderson, Max Bain, Mikel Artetxe, Nishant Relan, Piotr Padlewski, Qi Liu, Ren Chen, Samuel Phua, Yazheng Yang, Yi Tay, Yuqi Wang, Zhongkai Zhu, and Zhihui Xie (2024). *Reka Core, Flash, and Edge: A Series of Powerful Multimodal Language Models*. arXiv: 2404.12387 [cs.CL]. URL: https://arxiv.org/abs/2404.12387.

Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer,

Hisham Cholakkal, Ivan Laptev, Mubarak Shah, Fahad Shahbaz Khan, and Salman Khan (2025). *LlamaV-o1: Rethinking Step-by-step Visual Reasoning in LLMs*. arXiv: 2501.06186 [cs.CV]. URL: https://arxiv.org/abs/2501.06186.

*The Frasier Archives* (n.d.). https://www.kacl780.net/.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang (2022). "Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training". In: *Advances in neural information processing systems*.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri (2015). "Learning Spatiotemporal Features with 3D Convolutional Networks". In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri (2018). "A closer look at spatiotemporal convolutions for action recognition". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017a). "Attention is all you need". In: *Advances in neural information processing systems*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017b). "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*.

Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol (2024). "Covr: Learning composed video retrieval from web video captions". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler (2018). "Moviegraphs: Towards understanding human-centric situations from videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe (2019). "MOTS: Multi-Object Tracking and Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao (2023). "Videomae v2: Scaling video masked autoencoders

with dual masking". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Limin Wang, Yu Qiao, Xiaoou Tang, et al. (2014). "Action recognition and detection by combining motion and appearance features". In: *THUMOS14 Action Recognition Challenge*.

Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopz Moreno (2018). "Speaker diarization with LSTM". In: *Proc. ICASSP*.

Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. (2022). "Internvideo: General video foundation models via generative and discriminative learning". In: *arXiv preprint arXiv:2212.03191*.

Yulin Wang, Zhaoxi Chen, Haojun Jiang, Shiji Song, Yizeng Han, and Gao Huang (2021). "Adaptive focus for efficient video recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang (2020). "Towards Real-Time Multi-Object Tracking". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. (2022). "Dualprompt: Complementary prompting for rehearsal-free continual learning". In: *European Conference on Computer Vision*. Springer.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. (2022). "Robust fine-tuning of zero-shot models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Zuxuan Wu, Caiming Xiong, Yu-Gang Jiang, and Larry S Davis (2019). "Liteeval: A coarse-to-fine framework for resource efficient video recognition". In: *Advances in Neural Information Processing Systems*.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua (2021). "Next-qa: Next phase of question-answering to explaining temporal actions". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Shaoning Xiao, Long Chen, Songyang Zhang, Wei Ji, Jian Shao, Lu Ye, and Jun Xiao (2021). "Boundary proposal network for two-stage natural language video localization". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Eric Zhongcong Xu, Zeyang Song, Satoshi Tsutsui, Chao Feng, Mang Ye, and Mike Zheng Shou (2022). "AVA-AVD: Audio-Visual Speaker Diarization in the Wild". In: MM '22.

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: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics.

Huijuan Xu, Abir Das, and Kate Saenko (2017). "R-C3D: Region Convolutional 3D Network for Temporal Activity Detection". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui (2016). "MSR-VTT: A Large Video Description Dataset for Bridging Video and Language". In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). URL: https://www.microsoft.com/en-us/research/publication/msr-vtt-a-large-video-description-dataset-for-bridging-video-and-language/.

Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda (2021). "TransCenter: Transformers with Dense Queries for Multiple-Object Tracking". In: *CoRR*.

Zhenbo Xu, Wei Zhang, Xiao Tan, Wei Yang, Huan Huang, Shilei Wen, Errui Ding, and Liusheng Huang (2020). "Segment as Points for Efficient Online Multi-Object Tracking and Segmentation". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*.

Kota Yamaguchi, Alexander C. Berg, Luis E. Ortiz, and Tamara L. Berg (2011). "Who are you with and where are you going?" In: *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. IEEE Computer Society.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid (2022a). "TubeDETR: Spatio-Temporal Video Grounding With Transformers". In: *CVPR*.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid (2022b). "Zero-shot video question answering via frozen bidirectional language models". In: *Advances in Neural Information Processing Systems*.

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: *CVPR*.

Fan Yang, Wongun Choi, and Yuanqing Lin (2016). "Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*.

Xun Yang, Fuli Feng, Wei Ji, Meng Wang, and Tat-Seng Chua (2021). "Deconfounded video moment retrieval with causal intervention". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu (2022). "FILIP: Fine-grained Interactive Language-Image Pre-Training". In.

Qinghao Ye, Guohai Xu, Ming Yan, Haiyang Xu, Qi Qian, Ji Zhang, and Fei Huang (2023). "Hitea: Hierarchical temporal-aware video-language pre-training". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Chun-Hsiao Yeh, Bryan Russell, Josef Sivic, Fabian Caba Heilbron, and Simon Jenni (2023). "Meta-Personalizing Vision-Language Models To Find Named Instances in Video". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei (2016). "End-to-end learning of action detection from frame glimpses in videos". In: *CVPR*.

En Yu, Zhuoling Li, Shoudong Han, and Hongwei Wang (2021). "RelationTrack: Relation-aware Multiple Object Tracking with Decoupled Representation". In: *CoRR*.

Keunwoo Peter Yu (n.d.). *VideoBLIP*. URL: https://github.com/yukw777/VideoBLIP.

Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal (2023). "Self-Chained Image-Language Model for Video Localization and Question Answering". In: *arXiv preprint arXiv:2305.06988*.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi (2021). "MERLOT: Multimodal Neural Script Knowledge Models". In: *arXiv:2106.02636 [cs]*. URL: http://arxiv.org/abs/2106.02636 (visited on 04/13/2022).

Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei (2022). "MOTR: End-to-End Multiple-Object Tracking with TRansformer". In: *European Conference on Computer Vision (ECCV)*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer (2023). *Sigmoid Loss for Language Image Pre-Training*. arXiv: 2303.15343 [cs.CV]. URL: https://arxiv.org/abs/2303.15343.

Aonan Zhang, Quan Wang, Zhenyao Zhu, John Paisley, and Chong Wang (2019). "Fully supervised speaker diarization". In: *Proc. ICASSP*.

Chen-Lin Zhang, Jianxin Wu, and Yin Li (2022). "Actionformer: Localizing moments of actions with transformers". In: *European Conference on Computer Vision*. Springer.

Hang Zhang, Xin Li, and Lidong Bing (2023a). "Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding". In: *arXiv preprint arXiv:2306.02858*. URL: https://arxiv.org/abs/2306.02858.

Hang Zhang, Xin Li, and Lidong Bing (2023b). *Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding*. arXiv: 2306.02858 [cs.CL].

Hao Zhang, Aixin Sun, Wei Jing, Guoshun Nan, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh (2021). "Video corpus moment retrieval with contrastive learning". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Li Zhang, Yuan Li, and Ramakant Nevatia (2008). "Global data association for multi-object tracking using network flows". In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*.

Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan (2019). "Hacs: Human action clips and segments dataset for recognition and temporal localization". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Shuai Zhao, Xiaohan Wang, Linchao Zhu, and Yi Yang (2024). "Test-Time Adaptation with CLIP Reward for Zero-Shot Generalization in Vision-Language Models". In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=kIP0duasBb.

Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin (2017). "Temporal Action Detection with Structured Segment Networks". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*.

Yuan Zhi, Zhan Tong, Limin Wang, and Gangshan Wu (2021). "Mgsampler: An explainable sampling strategy for video action recognition". In: *Proceedings of the IEEE/CVF International conference on Computer Vision*.

Yujie Zhong, Relja Arandjelovic, and Andrew Zisserman (2016a). “Faces in Places: compound query retrieval”. In: *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*.

Yujie Zhong, Relja Arandjelović, and Andrew Zisserman (2016b). “Faces in Places: Compound Query Retrieval”. In: *Brit. Mach. Vis. Conf.*

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*.

Luowei Zhou, Chenliang Xu, and Jason J. Corso (2018). “Towards Automatic Learning of Procedures From Web Instructional Videos”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*.

Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl (2020). “Tracking Objects as Points”. In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV*.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai (2021). “Deformable DETR: Deformable Transformers for End-to-End Object Detection”. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

# Appendix A

# Statement of Authorship

A statement of authorship is provided for each multi-authored paper included in this thesis. The statements describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication, there exists a complete statement that is filled out and signed by the candidate and supervisor.

## Statement of Authorship for the paper "End-to-end Tracking with a Multi-query Transformer" in Chapter 2.

| | |
|---|---|
| Paper title | End-to-end Tracking with a Multi-query Transformer |
| Authors | **Bruno Korbar**, Andrew Zisserman |
| Publication status | Unpublished |
| Publication details | Released as a technical report |

### Student Confirmation

| | | |
|---|---|---|
| Student name | Bruno Korbar | |
| Contribution to the paper | First-author contribution:<br>• conception of research ideas<br>• design and implementation of models<br>• running of experiments<br>• writing and presentation of the paper | |
| Signature and Date |  | Oct. 10th 2024 |

### Supervisor Confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

| | | |
|---|---|---|
| Supervisor name | Prof. Andrew Zisserman | |
| Supervisor comments | | |
| Signature and Date |  | Oct. 10th 2024 |

## Statement of Authorship for the paper "Text-Conditioned Resampler For Long Form Video Understanding" in Chapter 3.

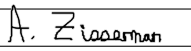
| | |
|---|---|
| Paper title | Text-Conditioned Resampler For Long Form Video Understanding |
| Authors | **Bruno Korbar**, Yongqin Xian, Alessio Tonioni, Andrew Zisserman, Federico Tombari |
| Publication status | Published |
| Publication details | In 2024 IEEE European Conference on Computer Vision |

### Student Confirmation

| | | |
|---|---|---|
| Student name | Bruno Korbar | |
| Contribution to the paper | First-author contribution:<br>• conception of research ideas<br>• design and partial implementation of models<br>• running of large scale experiments<br>• participated in writing and presentation of the paper | |
| Signature and Date |  | Oct. 10th 2024 |

### Supervisor Confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

| | | |
|---|---|---|
| Supervisor name | Prof. Andrew Zisserman | |
| Supervisor comments | | |
| Signature and Date |  | Oct. 10th 2024 |

## Statement of Authorship for the paper "Personalised CLIP or: how to find your vacation videos" in Chapter 4.

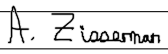
| | |
|---|---|
| Paper title | Personalised CLIP or: how to find your vacation videos |
| Authors | **Bruno Korbar**, Andrew Zisserman |
| Publication status | Published |
| Publication details | In 2022 British Machine Vision Conference |

### Student Confirmation

| | | |
|---|---|---|
| Student name | Bruno Korbar | |
| Contribution to the paper | First-author contribution:<br>• conception of research ideas<br>• design and implementation of models<br>• creation of a benchmark dataset<br>• running of experiments<br>• writing and presentation of the paper | |
| Signature and Date |  | Oct. 10th 2024 |

### Supervisor Confirmation

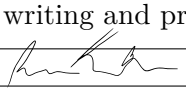
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

| | | |
|---|---|---|
| Supervisor name | Prof. Andrew Zisserman | |
| Supervisor comments | | |
| Signature and Date |  | Oct. 10th 2024 |

## Statement of Authorship for the paper "Personalizing Retrieval using Joint Embeddings; or "the Return of Fluffy"" in Chapter 5.

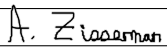
| | |
|---|---|
| Paper title | Personalizing Retrieval using Joint Embeddings; or "the Return of Fluffy" |
| Authors | **Bruno Korbar**, Andrew Zisserman |
| Publication status | Under review |
| Publication details | Under review for IEEE Conference on Computer Vision and Pattern Recognition |

### Student Confirmation

| | | |
|---|---|---|
| Student name | Bruno Korbar | |
| Contribution to the paper | First-author contribution:<br>• conception of research ideas<br>• design and implementation of models<br>• curating sample datasets<br>• running all experiments<br>• writing and presentation of the paper | |
| Signature and Date |  | Oct. 10th 2024 |

### Supervisor Confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

| | | |
|---|---|---|
| Supervisor name | Prof. Andrew Zisserman | |
| Supervisor comments | | |
| Signature and Date |  | Oct. 10th 2024 |

## Statement of Authorship for the paper "Look, Listen and Recognise: Character-Aware Audio-Visual Subtitling" in Chapter 6.

| | |
|---|---|
| Paper title | Look, Listen and Recognise: Character-Aware Audio-Visual Subtitling |
| Authors | **Bruno Korbar**, Jaesung Huh, Andrew Zisserman |
| Publication status | Published |
| Publication details | In 2024 IEEE International Conference on Acoustics, Speech and Signal Processing |

### Student Confirmation

| | | |
|---|---|---|
| Student name | Bruno Korbar | |
| Contribution to the paper | First-author contribution:<br>• cleaning and annotation of datasets<br>• design and implementation of models<br>• running of visual experiments<br>• writing and presentation of the paper | |
| Signature and Date | | Oct. 10th 2024 |

### Supervisor Confirmation

By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

| | | |
|---|---|---|
| Supervisor name | Prof. Andrew Zisserman | |
| Supervisor comments | | |
| Signature and Date | A. Zisserman | Oct. 10th 2024 |