Object-Centric Perception and Description of Video



Junyu Xie

Keble College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2026

Abstract

In this thesis, we explore video understanding from an object-centric perspective, treating *objects* as the fundamental units of computation. These objects persist over time, move through the scene, occlude and interact with one another, and, in edited media, often act out a story. We develop this perspective through methods for both low-level object perception and high-level object understanding.

The first direction of this thesis concerns the *discovery* of objects from motion, motivated by the Gestalt principle of common fate, which suggests that the perception of objectness can emerge from coherent motion. Following this intuition, we study moving object segmentation (MOS), the task of discovering, segmenting, and tracking multiple objects from motion cues. Our first work relies on optical flow as the primary motion signal, developing a Sim2Real MOS framework that transfers effectively to real-world videos and is further enhanced through appearance-based refinement. We then leverage large pretrained segmentation models and feed-forward 3D reconstruction backbones to achieve more accurate and geometrically grounded MOS, extending the task to temporally fine-grained, per-frame motion-state prediction.

The second direction focuses on *understanding* videos through the lens of characters, which are often the primary narrative objects in edited media. We first study the task of Audio Description (AD) generation for visually impaired audiences, leveraging advances in large multimodal models through training-free pipelines. Character information and film grammar are incorporated through structured prompting, enabling more accurate and narratively grounded descriptions. Finally, we address the problem of streaming character naming, developing a framework that infers character identities progressively from audiovisual cues and dialogue, providing a foundation for future character-centric video understanding.

Keywords – object-centric video understanding, edited media understanding, moving object segmentation, Audio Description generation, automatic character naming.

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.

Junyu Xie, June 2026.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisors, Andrew Zisserman and Weidi Xie. Your patient guidance, insightful advice, and trust in my judgement gave me the freedom to pursue my own research questions. Working under your supervision has taught me not only how to conduct research, but also how to identify questions that are meaningful and worth pursuing.

Each chapter of this thesis bears the contributions of many co-authors. Charig Yang, Zhongrui Gui, Minghao Chen, Eshika Khandelwal, and Runjia Li were collaborators in the fullest sense, and I am grateful for the countless discussions, problem-solving sessions, and deadlines we navigated together. Skanda Koppula and Guillaume Le Moing, who also hosted me during my internship, showed me how ambitious research is built through effective collaboration and a shared commitment to scientific rigour. Senior collaborators who shaped these projects, including Tengda Han, Chuhan Zhang, Yan Xia, Max Bain, Ignacio Rocco, Liliane Momeni, Arsha Nagrani, Shuyang Sun, Junlin Zhang, Gül Varol, Mehdi S. M. Sajjadi, Iro Laina, and Andrea Vedaldi, were as much mentors as collaborators. I owe a great deal of my growth as a researcher to their guidance, generosity, and expertise.

Co-organising the SLoMO workshop on Story-Level Movie Understanding and Audio Description at ICCV 2025 was one of the highlights of my DPhil. Thank you to Ridouane Ghermi, Divy Kala, Xi Wang, Makarand Tapaswi, and Ivan Laptev for the shared effort and for helping build a community around this exciting research area.

Beyond my immediate collaborators, VGG has been an exceptionally generous place to spend the past years. For their discussions, encouragement, and friendship, I thank Jianyuan Wang, Guanqi Zhan, Jaesung Huh, Stanislaw Szymanowicz, Laurynas Karazija, Zeren Jiang, Zihang Lai, Ruining Li, Orest Kupyn, Sindhu Hegde, Hala Lamdouar, Chuanxia Zheng, Jilan Xu, Yushi Lan, Shangzhe Wu, and Kai Han. I am especially grateful to Ashish Thandavan, Abhishek Dutta, and David Pinto for keeping the in-

frastructure running and for patiently answering my endless server questions over the years.

Outside Oxford, conversations with Zihui Xue, Letian Wang, Chao Huang, and Joao Carreira helped me see my work from different perspectives, and I am grateful for the insight they shared.

I am indebted to Keble College for providing a welcoming home throughout my DPhil, and to the Clarendon Scholarship for making it possible.

Above all, I thank my parents. Their love, encouragement, and unwavering support at every stage of this journey have made everything else possible.

Contents

1 Introduction **12**

1.1 Motivation and key ideas 12

1.2 Background and related work 13

1.2.1 Object-centric representation learning 14

1.2.2 Object detection, segmentation, and tracking in videos . . . 16

1.2.3 High-level object understanding in videos 17

1.3 Thesis outline and contributions 19

1.4 Publications 21

I Object Discovery through Motion in Videos **24**

2 Segmenting Moving Objects via an Object-Centric Layered Representation **25**

2.1 Introduction 26

2.2 The object-centric layered representation (OCLR) model 28

2.2.1 Layered representation 29

2.2.2 Architecture 30

2.2.3 Training objectives 31

2.3 Synthetic dataset generation . . . 32
2.4 Experiments . . . 34
2.4.1 Datasets . . . 34
2.4.2 Evaluation metrics . . . 35
2.4.3 Implementation details . . . 35
2.4.4 Ablation study . . . 36
2.4.5 Single object video segmentation . . . 37
2.4.6 Multiple object video segmentation . . . 39
2.5 Related work . . . 41
2.6 Discussion . . . 42

3 Appearance-Based Refinement for Object-Centric Motion Segmentation 45
3.1 Introduction . . . 46
3.2 Related work . . . 49
3.3 Appearance-based selection and correction . . . 51
3.3.1 Problem scenario . . . 51
3.3.2 Appearance-based mask selection . . . 52
3.3.3 Appearance-based mask correction . . . 54
3.3.4 Sim2Real training procedure . . . 57
3.4 Experiments . . . 59
3.4.1 Datasets and metrics . . . 59
3.4.2 Implementation details . . . 60
3.4.3 Multiple object video segmentation . . . 60
3.4.4 Single object video segmentation . . . 62
3.5 Discussion . . . 63

4 Moving Object Segmentation: All You Need Is SAM (and Flow) 65
4.1 Introduction . . . 66

4.2 Related work . . . 68
4.3 SAM preliminaries . . . 70
4.4 Frame-level segmentation I: Flow as input . . . 71
4.5 Frame-level segmentation II: Flow as prompt . . . 72
4.6 Sequence-level mask association . . . 74
4.7 Experiments . . . 75
4.7.1 Datasets . . . 75
4.7.2 Evaluation metrics . . . 76
4.7.3 Implementation details . . . 77
4.7.4 Ablation study . . . 78
4.7.5 Quantitative results . . . 79
4.7.6 Qualitative visualisations . . . 81
4.8 Discussion . . . 83

5 Grounding Moving Object Segmentation in 3D Space and Time 85
5.1 Introduction . . . 87
5.2 Method: GMOS . . . 89
5.2.1 GMOS proposer . . . 89
5.2.2 GMOS propagator . . . 92
5.3 Dataset: GMOS-2K . . . 93
5.4 Experiments . . . 95
5.4.1 Training datasets . . . 95
5.4.2 Benchmarks for evaluation . . . 96
5.4.3 Implementation details . . . 97
5.4.4 Quantitative results . . . 98
5.4.5 Qualitative results . . . 99
5.5 Related works . . . 100
5.6 Conclusion . . . 102

II Video Understanding and Audio Description 104

6 AutoAD-Zero: A Training-Free Framework for Zero-Shot Audio Description 105

6.1 Introduction . . . 106
6.2 Related work . . . 108
6.3 Prompting VLM with character information . . . 110
6.3.1 Character recognition by face detection . . . 111
6.3.2 Visual prompting VLM by circling character faces . . . 112
6.4 Two-stage training-free AD generation . . . 113
6.4.1 Stage I: VLM-based video description . . . 114
6.4.2 Stage II: LLM-based AD summary . . . 115
6.5 TV-AD dataset . . . 116
6.5.1 Dataset details . . . 116
6.5.2 Dataset formulation: Aligning ADs with TV series . . . 117
6.6 Experiments . . . 119
6.6.1 Datasets . . . 119
6.6.2 Evaluation metrics . . . 120
6.6.3 Implementation details . . . 120
6.6.4 Character recognition results . . . 121
6.6.5 Audio Description predictions . . . 121
6.7 Discussion and extensions . . . 123

7 Shot-By-Shot: Film-Grammar-Aware Training-Free Audio Description Generation 125

7.1 Introduction . . . 126
7.2 Related work . . . 129
7.3 Training-free AD generation framework . . . 132
7.3.1 Leveraging shot-based temporal context . . . 133

7.3.2 Leveraging thread structure . . . 134
7.3.3 Leveraging shot scale information . . . 137
7.4 Action score . . . 137
7.5 Experiments . . . 139
7.5.1 AD generation datasets and metrics . . . 139
7.5.2 Implementation details . . . 139
7.5.3 Thread structure prediction . . . 140
7.5.4 Shot scale classification . . . 141
7.5.5 Human alignment with action scores . . . 141
7.5.6 AD generation – evaluation of components . . . 142
7.5.7 AD generation – comparison with SotA . . . 144
7.5.8 Assisted AD generation . . . 146
7.6 Discussion – summary and limitations . . . 147

8 As the Story Unfolds: Watching a Film and Identifying Characters as a Human Does 148

8.1 Introduction . . . 149
8.2 Movie character naming (MCN) dataset . . . 151
8.2.1 Data collection and curation . . . 152
8.2.2 Dataset statistics . . . 153
8.2.3 Preprocessing – character-level face-tracks and associated speech . . . 154
8.3 Assigning names to characters . . . 154
8.3.1 MLLM-based character naming . . . 155
8.3.2 Fine-tuning the MLLM for character naming . . . 156
8.4 Streaming pipeline for character bank construction . . . 157
8.4.1 Character bank update . . . 157
8.4.2 Error correction during character bank construction . . . 159

8.5 Experiments . . . 159
8.5.1 Evaluation metrics . . . 160
8.5.2 Results on character naming . . . 161
8.5.3 Results on streaming character bank construction . . . 163
8.6 Related work . . . 164
8.7 Discussion . . . 166

9 Discussion **167**
9.1 Achievement and impact . . . 167
9.1.1 Moving object segmentation . . . 167
9.1.2 Movie understanding and Audio Description . . . 169
9.2 Future work . . . 170
9.2.1 Moving object segmentation . . . 170
9.2.2 Movie understanding and Audio Description . . . 171

References **172**

A Statement of Authorship **208**

Chapter 1

Introduction

1.1 Motivation and key ideas

The visual world, as captured by video, is not merely a stream of pixels. Instead, it is populated by *objects*, entities with persistent shape and identity that move through the scene, interact with one another, and, in films and television, act out a story. Treating these objects, rather than pixels, frames, or clips, as the fundamental units of computation forms the object-centric perspective that underlies this thesis. This perspective spans the full spectrum of video understanding, from low-level perception to high-level reasoning.

At the *low level*, the notion of an object is tightly bound to motion. Humans acquire a sense of objectness early in development by perceptually grouping parts that move together. This bottom-up discovery of objects from motion can be transferred to computer vision systems through increasingly rich motion representations. Optical flow, which captures sub-pixel displacement between consecutive frames, provides a strong segmentation signal at the discontinuities that mark object boundaries, but remains fragile under camera motion, articulation, and stationary intervals. Point trajectories extend motion reasoning over longer temporal horizons, while recent feed-forward 3D reconstruction backbones recover true motion in 3D space, disentangling object motion from camera motion within a unified geometric representation.

At the *high level*, edited media such as film and television are organised around *characters*, the primary narrative objects whose actions and interactions drive the

story. Building character-centric systems therefore requires recognising characters, maintaining their identities over time, and understanding their role in the narrative. While principal characters are immediately apparent to human viewers, distinguishing them from background individuals and assigning them names remains difficult for machines, particularly in crowded scenes and animated content. Beyond recognising who the characters are, a complete understanding of a story also requires understanding how they interact with one another and with their environment. These interactions drive the narrative forward and shape the unfolding plot. Understanding and being able to describe them is central to a broad range of video-understanding tasks, including video question answering, captioning, summarisation, and transcript generation.

The object-centric perspective is not only conceptually appealing but also central to a broad range of applications, particularly those involving object motion. In *perceiving* the world, object-centric reasoning supports the discovery of camouflaged animals in wildlife footage [Lamdouar et al., 2020], motion detection and tracking in autonomous driving and surveillance [Chen et al., 2021], and the analysis of cellular dynamics in biomedical imaging [Ulman et al., 2017]. In *describing* the world, the actions and interactions of objects underpin human action recognition [Simonyan and Zisserman, 2014a], the inference of physical properties from motion [Wu et al., 2015], and the generation of captions, summaries, and descriptions for films and television [Han et al., 2024]. In *interacting* with the world, persistent object identity and motion prediction are fundamental to robotic manipulation [Florence et al., 2018] and immersive AR/VR systems that must reason about how users and environments evolve over time [Runz et al., 2018].

1.2 Background and related work

The notion of "objects" is deeply rooted in human perception, tracing back to the earliest stages of visual development. The *Gestalt principle of common fate* [Wertheimer, 1923] states that visual elements moving together are perceived as belonging to the same object, allowing motion to reveal objects even when appearance alone provides little distinction from the background. A complementary concept is *object permanence*, the understanding that objects continue to exist when occluded,

outside the field of view, or temporarily stationary, enabling stable object identities to be maintained over time [Piaget, 1954; Baillargeon et al., 1985; Spelke, 1990; Baillargeon and DeVos, 1991].

These principles have also profoundly influenced modern computer vision, where object-centric representations provide a natural framework for understanding visual scenes and videos. This section reviews the literature through the lens of the object-centric perspective. Section 1.2.1 discusses how object-centric *representations* are learned, followed by Section 1.2.2, which reviews methods for *localising* objects through detection, segmentation, and tracking. Finally, Section 1.2.3 considers *high-level* video understanding, focusing on characters in edited media as the primary narrative objects that are recognised, described, and named.

1.2.1 Object-centric representation learning

The idea that a visual scene is naturally represented as a collection of discrete, persistent *objects* rather than pixels has long motivated object-centric representation learning. Two broad families of approaches have emerged, differing primarily in the structure imposed on the scene decomposition. *Layered* representations model a scene as an ordered stack of objects with explicit occlusion relationships, whereas *slot-based* representations treat it as an unordered set of latent components discovered through reconstruction.

The layered decomposition, introduced by Wang and Adelson [Wang and Adelson, 1994], models a video as an ordered stack of layers, each encoding an object's appearance, opacity, and motion while explicitly preserving occlusion relationships. Although it predates the deep-learning era, the idea has been revived for neural video decomposition and editing, including layered neural atlases [Kasten et al., 2021] and Omnimatte [Lu et al., 2021], which recover RGBA layers together with associated visual effects, and more recent approaches that cast layered decomposition as a video-diffusion problem [Lee et al., 2025].

A complementary family relaxes this explicit ordering. Rather than assigning objects to predefined layers, slot-based models decompose an image into a fixed set of *unordered* latent components using reconstruction as the sole training signal. In deep learning, this intuition was first operationalised through unsupervised gener-

ative models such as Attend, Infer, Repeat [Eslami et al., 2016], MONet [Burgess et al., 2019], IODINE [Greff et al., 2019], and GENESIS [Engelcke et al., 2020], which infer object-centric latent variables through recurrent or iterative encoding. Slot Attention [Locatello et al., 2020] distilled this paradigm into a simple iterative-attention module in which a set of permutation-invariant *slots* compete to bind to objects.

A limitation of this family of methods is its reliance on appearance reconstruction as the primary grouping signal. While these models perform well on synthetic benchmarks with simple geometry and uniform textures, they often struggle on real-world scenes, where reconstruction is dominated by low-level appearance rather than object structure.

Extending slot-based representations to video both increases the difficulty and introduces a natural source of supervision. Objects must not only be discovered, but also tracked through time, while temporal coherence itself provides a powerful cue for grouping. Early approaches addressed this challenge through explicit factorisation and propagation mechanisms [Veerapaneni et al., 2019; Jiang et al., 2020], later systematically benchmarked by [Weis et al., 2021]. Subsequent methods incorporated motion more directly. SAVi [Kipf et al., 2022] predicts optical flow as a reconstruction target, SAVi++ [Elsayed et al., 2022] incorporates depth to scale to real driving footage, STEVE [Singh et al., 2022] replaces pixel reconstruction with a transformer-based decoder for naturalistic videos, and SlotFormer [Wu et al., 2023c] models slot dynamics for future prediction.

More recently, the reconstruction target has shifted from pixels to self-supervised visual features. DINOSAUR [Seitzer et al., 2023] reconstructs DINO features rather than RGB images and demonstrates that slot-based representations can scale to substantially more complex scenes. Its video counterparts learn object-centric representations by reconstructing or predicting the temporal evolution of such features [Aydemir et al., 2023; Zadaianchuk et al., 2023], while later work introduces slot-level contrastive objectives to improve temporal consistency [Manasyan et al., 2025]. A parallel line of research replaces mixture decoders with diffusion-based generators to better model realistic textures [Wu et al., 2023d], and object-centric latents have increasingly been adopted as state representations for predictive world models in video understanding and control [Jiang et al., 2024;

Villar-Corrales and Behnke, 2025].

1.2.2 Object detection, segmentation, and tracking in videos

Localising objects in images has a much longer history than its video counterpart. Object detection evolved from region-proposal pipelines [Ren et al., 2015] to end-to-end set prediction with transformers [Carion et al., 2020], while dense prediction advanced from Mask R-CNN [He et al., 2017] to unified mask-classification architectures [Cheng et al., 2022a]. These methods establish the per-frame *what* and *where* of visual perception, but treat each frame independently, leaving temporal association to video-based methods.

Video object segmentation (VOS) introduces this temporal dimension under two main protocols. In *semi-supervised* VOS, a first-frame mask is provided and propagated throughout the sequence, a setting dominated by space-time memory matching approaches [Oh et al., 2019; Cheng and Schwing, 2022], with more recent methods introducing object-level memory reading [Cheng et al., 2024a]. In *unsupervised* VOS, no annotation is available at inference time, and the system must discover salient objects automatically [Jain et al., 2017; Lu et al., 2019; Yang et al., 2019d].

Whereas video object segmentation propagates an object that is either specified or automatically selected as salient, *video instance segmentation* (VIS) must detect, segment, classify, and track *all* object instances belonging to a predefined category set. VIS therefore combines object detection, segmentation, and tracking, while also handling an unknown and variable number of objects. Early approaches followed a tracking-by-detection paradigm, segmenting each frame independently before associating instances across time [Yang et al., 2019b], whereas clip-level methods instead propagate instance masks across neighbouring frames to jointly classify, segment, and track [Bertasius and Torresani, 2020]. Transformer-based methods later reformulated VIS using object queries to predict each instance's mask, category, and identity, either over an entire clip [Wang et al., 2021c] or online with contrastive instance association [Wu et al., 2022]. More recently, decoupled designs that combine a strong image-level segmenter with a task-agnostic temporal propagator [Cheng et al., 2023a] have proven particularly effective and have enabled VIS to scale towards large-vocabulary and open-vocabulary settings.

Independently of any semantic class vocabulary, *motion* itself provides a powerful cue for object discovery. Classical motion segmentation clusters long-term point trajectories [Brox and Malik, 2010; Ochs and Brox, 2011; Keuper et al., 2015; Fragkiadaki et al., 2012], models per-pixel optical flow under a moving camera [Bideau and Learned-Miller, 2016b], or detects moving objects through motion boundaries and appearance models [Papazoglou and Ferrari, 2013]. Deep learning approaches segment motion directly from optical flow, for example by reconstructing the flow field using slot-based architectures [Yang et al., 2021a] or by predicting which regions are likely to move [Choudhury et al., 2022; Meunier et al., 2023]. Advances in point tracking [Harley et al., 2022; Doersch et al., 2022; Doersch et al., 2023; Karaev et al., 2025] have extended motion reasoning over longer temporal horizons, enabling segmentation to be learned from dense trajectories [Karazija et al., 2024; Huang et al., 2025a].

Two recent developments have substantially reshaped this landscape. First, promptable foundation segmenters, including SAM [Kirillov et al., 2023], its video successor SAM2 [Ravi et al., 2024], and its more recent concept-promptable variant SAM3 [Carion et al., 2026], provide strong object masks that can be guided by points, boxes, or text. These models have also been coupled with language models for referring segmentation [Yuan et al., 2025]. Second, feed-forward 3D reconstruction backbones [Wang et al., 2024c; Leroy et al., 2024; Wang et al., 2025b; Wang et al., 2026] estimate scene geometry directly from images, enabling motion to be reasoned about in 3D rather than solely in the image plane. Recent methods recover moving regions alongside temporally consistent reconstructions [Zhang et al., 2025] or isolate them to reconstruct a clean static scene [Xu et al., 2024], exploit camera and depth estimates to separate moving from static content [Li et al., 2025b; Goli et al., 2025], or disentangle object motion from camera motion using geometric latent representations [Chen et al., 2025; He et al., 2026].

1.2.3 High-level object understanding in videos

Semantic video understanding was initially driven by action recognition, progressing from two-stream networks [Simonyan and Zisserman, 2014b] through inflated 3D convolutions [Tran et al., 2015; Carreira and Zisserman, 2017] and their two-pathway refinements [Feichtenhofer et al., 2019] to video transformers [Arnab et

al., 2021; Bertasius et al., 2021]. These methods typically treat a video clip as a holistic unit. An object-centric perspective instead views an action as the evolving configuration of the participating objects. Action Genome [Ji et al., 2020] represents actions through temporal scene graphs over person and object nodes, video visual-relation detection predicts subject–predicate–object tubes [Shang et al., 2017], and compositional approaches built on hand–object interaction graphs generalise to unseen verb–noun combinations [Materzynska et al., 2020]. Object-region transformers further make objects explicit architectural units by injecting detected regions and their trajectories into the video backbone [Herzig et al., 2022; Wu and Krahenbuhl, 2021].

This perspective naturally extends to compositional and causal reasoning, where discrete object representations serve as the substrate for video question answering [Yi et al., 2020; Wu et al., 2021]. With the emergence of Video-LLMs [Zhang et al., 2023b; Maaz et al., 2024], which typically process video as sequences of frame-level tokens, recent work [Feng et al., 2025b; Wu et al., 2026] has reintroduced explicit object structure by tokenising detected objects alongside the global visual stream.

A complementary direction grounds the model's output in specific objects localised across space and time. Spatio-temporal video grounding localises the trajectory of a queried object [Zhang et al., 2020], while dense video object captioning generates descriptions for individual object tracks [Zhou et al., 2025e]. In the Video-LLM era, this grounding has become increasingly fine-grained, with multimodal models interleaving object masks or bounding boxes into generated responses [Munasinghe et al., 2025; Yuan et al., 2025], making objects the addressable units of both queries and outputs.

In edited media such as film and television, the principal narrative objects are the *characters*, whose recurrence binds together stories that unfold over hours. This perspective is reflected in datasets that place characters at the centre of annotation and structure [Huang et al., 2020; Vicol et al., 2018; Bain et al., 2020]. A central challenge for long-form video understanding is therefore maintaining character identity across scenes and over extended temporal horizons [Ji et al., 2025; He et al., 2025].

Beyond identifying who the characters are, a complete understanding of a story also requires describing their actions and interactions. One of the canonical character-centric description tasks is *Audio Description* (AD) generation. It aims to narrate visual content for visually impaired audiences by complementing the original audio and dialogue signals, making it a naturally visually grounded task with clear societal value. Building on large-scale AD datasets [Soldan et al., 2022; Han et al., 2024], AD generation has progressed from context-conditioned models [Han et al., 2023b] to systems that explicitly reason about character identity through character banks [Han et al., 2023a; Han et al., 2024]. Subsequent work has incorporated richer narrative context [Wang et al., 2025a; Fang et al., 2024] and developed dedicated evaluation protocols for assessing character and action understanding in generated descriptions [Kala et al., 2025].

Underlying these developments is the long-standing problem of character naming. Early work aligned scripts with subtitles and used active-speaker detection to label face tracks [Everingham et al., 2006; Sivic et al., 2009]. Subsequent methods integrated facial, vocal, and clothing cues within probabilistic frameworks [Tapaswi et al., 2012], incorporated coreference information [Ramanathan et al., 2014], and eventually dispensed with scripts by inferring identities directly from dialogue or cast lists [Haurilet et al., 2016; Nagrani and Zisserman, 2018; Cour et al., 2010]. As characters are frequently occluded, off-screen, or viewed from challenging viewpoints, more recent approaches perform multimodal clustering across face, body, and voice representations [Brown et al., 2021], or recognise speaking characters from audio-visual cues without requiring face detection [Korbar et al., 2024].

1.3 Thesis outline and contributions

Figure 1.1 provides an overview of the thesis structure, organised around two complementary directions. The first, *low-level perception* of objects in videos, develops methods for discovering, segmenting, and tracking moving objects directly from visual signals across the four chapters of Part I. The second, *high-level understanding* of objects in videos, develops methods for identifying, naming, and describing movie characters in films and television across the three chapters of Part II.

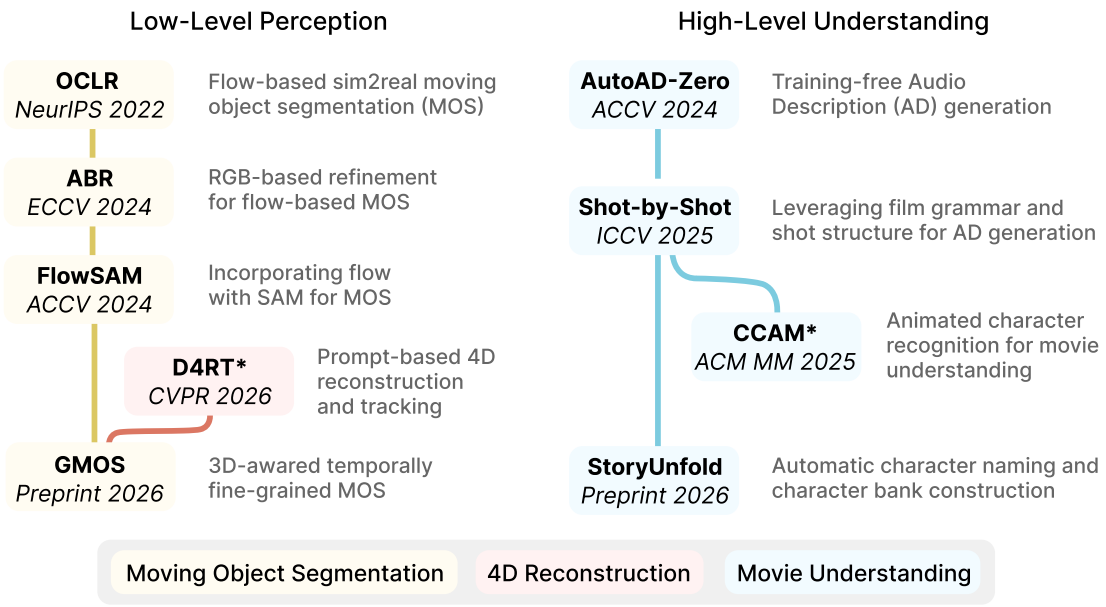


Figure 1.1: **Overview of the thesis structure.** The thesis is organised along two parallel lines of work, with each line proceeding chronologically across its chapters. Works marked with * are related but not included in the thesis.

Part I: Object Discovery through Motion in Videos

Part I progressively grounds moving object segmentation in richer visual representations, evolving from optical flow, through vision foundation models, to 3D geometric representations.

In Chapter 2, we introduce *OCLR*, an object-centric layered representation that discovers and segments multiple moving objects from optical flow alone through a Sim2Real training framework. In Chapter 3, we improve these motion-based predictions through appearance-guided refinement. In Chapter 4, we adapt the Segment Anything Model (SAM) for moving object segmentation through two complementary designs, *FlowI-SAM* and *FlowP-SAM*, together with a sequence-level association module for temporally consistent object tracking. Finally, in Chapter 5, we ground motion segmentation in 4D geometry through a proposer-and-propagator architecture built upon feed-forward 3D reconstruction and SAM2, and introduce the *GMOS-2K* dataset and *MOS-I* protocol for temporally fine-grained motion segmentation evaluation.

Part II: Video Understanding and Audio Description

Part II focuses on understanding characters in long-form video. We first study AD generation, progressively incorporating character identity and film grammar into the generation process. We then address the more fundamental problem of streaming character naming, enabling character identities to emerge as a story unfolds.

In Chapter 6, we introduce *AutoAD-Zero*, a training-free framework that combines character-aware visual prompts with a two-stage VLM–LLM pipeline for AD generation. Rather than fine-tuning on movie datasets, AutoAD-Zero prompts off-the-shelf models, thereby improving naturally as stronger foundation models become available. We also introduce the *TV-AD* dataset, which extends AD generation from films to the previously underexplored setting of television series. In Chapter 7, we make AD generation aware of *film grammar* by taking the *shot* as the unit of temporal context and leveraging thread structure and shot scale to guide VLM prompting, and introduce the character-free action score as a recall-oriented evaluation metric. Finally, in Chapter 8, we remove the standard assumption that an external character bank is available a priori and instead identify and name characters *as the story unfolds* from the film's own dialogue and visual cues.

1.4 Publications

Chapters 2 to 8 each contain a research paper. Most of them have been peer-reviewed and accepted for publication at a conference, with the exception of Chapter 5 and Chapter 8, which are currently under review. We make no modifications to the published papers except for formatting changes. For each paper, we also provide a statement of authorship in Appendix A. The papers included in the thesis are listed below.

Chapter 2: "Segmenting Moving Objects via an Object-Centric Layered Representation" **Junyu Xie**, Weidi Xie, Andrew Zisserman. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Chapter 3: "Appearance-Based Refinement for Object-Centric Motion Segmentation" **Junyu Xie**, Weidi Xie, Andrew Zisserman. In *European Conference on*

Computer Vision (ECCV), 2024.

Chapter 4: "Moving Object Segmentation: All You Need Is SAM (and Flow)" **Junyu Xie**, Charig Yang, Weidi Xie, Andrew Zisserman. In *Asian Conference on Computer Vision (ACCV)*, 2024.

Chapter 5: "Grounding Moving Object Segmentation in 3D Space and Time" **Junyu Xie**, Tengda Han, Weidi Xie, Andrew Zisserman. Submitted to *Advances in Neural Information Processing Systems (NeurIPS)*, 2026. Under review.

Chapter 6: "AutoAD-Zero: A Training-Free Framework for Zero-Shot Audio Description" **Junyu Xie**, Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, Andrew Zisserman. In *Asian Conference on Computer Vision (ACCV)*, 2024.

Chapter 7: "Shot-by-Shot: Film-Grammar-Aware Training-Free Audio Description Generation" **Junyu Xie**, Tengda Han, Max Bain, Arsha Nagrani, Eshika Khandelwal, Gül Varol, Weidi Xie, Andrew Zisserman. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Chapter 8: "As the Story Unfolds: Watching a Film and Identifying Characters as a Human Does" Zhongrui Gui, **Junyu Xie**, Tengda Han, Weidi Xie, Andrew Zisserman. Submitted to *Advances in Neural Information Processing Systems (NeurIPS)*, 2026. Under review.

Publications not included. The following works completed during the DPhil are not included in the thesis.

"SHAP-EDITOR: Instruction-guided Latent 3D Editing in Seconds" Minghao Chen, **Junyu Xie**, Iro Laina, Andrea Vedaldi. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

"Character-Centric Understanding of Animated Movies" Zhongrui Gui, **Junyu Xie**, Tengda Han, Weidi Xie, Andrew Zisserman. In *ACM International Conference on Multimedia (ACM MM)*, 2025.

"Efficiently Reconstructing Dynamic Scenes One Token at a Time" Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, **Junyu Xie**, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, Mehdi S. M. Sajjadi. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Part I

Object Discovery through Motion in Videos

Chapter 2

Segmenting Moving Objects via an Object-Centric Layered Representation

This paper was published at the Advances in Neural Information Processing Systems (NeurIPS), 2022.

Abstract

The objective of this paper is a model that is able to discover, track and segment multiple moving objects in a video. We make four contributions: First, we introduce an object-centric segmentation model with a depth-ordered layer representation. This is implemented using a variant of the transformer architecture that ingests optical flow, where each query vector specifies an object and its layer for the entire video. The model can effectively discover multiple moving objects and handle mutual occlusions; Second, we introduce a scalable pipeline for generating multi-object synthetic training data via layer compositions, that is used to train the proposed model, significantly reducing the requirements for labour-intensive annotations, and supporting Sim2Real generalisation; Third, we conduct thorough ablation studies, showing that the model is able to learn object permanence and temporal shape consistency, and is able to predict amodal segmentation masks; Fourth, we evaluate our model, trained only on synthetic data, on standard video segmentation benchmarks, DAVIS, MoCA, SegTrack, FBMS-59, and achieve state-of-the-art performance among existing methods that do not rely on any manual annotations. With test-time adaptation, we observe further performance boosts.

2.1 Introduction

Humans have the ability to discover and segment moving objects in videos. This ability is present even in young infants, together with notions of object permanence through occlusions, and temporal shape constancy [Piaget, 1954; Baillargeon et al., 1985; Spelke, 1990; Baillargeon and DeVos, 1991]. Achieving this ability by machine has long been a goal of the field [Brox and Malik, 2010; Ochs and Brox, 2011; Keuper et al., 2015; Bideau and Learned-Miller, 2016b; Tokmakov et al., 2017; Bideau et al., 2018; Xie et al., 2019; Tokmakov et al., 2019; Dave et al., 2019; Yang et al., 2019c; Lamdouar et al., 2020; Yang et al., 2021a], and modern methods are now able to reliably discover and segment single moving objects using both appearance (RGB) and motion (optical flow) streams of the video. However, success is limited in the case of multiple moving objects and occlusion.

Our goal in this paper is to develop and study a model that can discover multiple

moving objects and also handle their mutual occlusions. To this end, we propose a *layered* object-centric model that predicts *amodal* segmentation masks [Zhu et al., 2017] for each object throughout the video. Note that, given amodal segmentations, the depth ordering of the object layers can be determined from occlusions in the observed (modal) frames when the objects overlap. See Figure 2.1. Our innovation is to build this mechanism into the representation: when objects overlap in a frame, the model has to maintain the amodal shapes through temporal consistency, and can thus infer depth ordering.

To achieve this goal we use only optical flow as our primary information stream. There are benefits and drawbacks to this choice. The benefits are two fold: first, the information that we are after, the segmentation masks, are directly available from discontinuities in the flow field; second, a key advantage of using solely optical flow is that the domain adaptation problem is largely avoided [Doersch and Zisserman, 2019; Lamdouar et al., 2021] – and this means that the model can be trained on synthetic sequences and applied directly to real sequences without a significant Sim2Real disparity. However, the drawbacks are also two fold: first, unlike the appearance stream, objects can be invisible in flow when they are not moving, thereby requiring the model to build a concept of object permanence; and second, but more subtly, flow is not like appearance in that we cannot expect constancy even when the object is moving (for example, due to camera shake that affects the flow but not the appearance), and thus the model cannot strongly use flow coherence as a cue but must instead learn to model the segmentation mask shape.

We make four contributions: (i) we introduce an **O**bject-**C**entric **L**ayered **R**epresentation (**OCLR**) model for discovering and segmenting multiple moving objects and inferring their mutual occlusions from optical flow alone. The architecture of the model is based on a transformer, with DETR like learnable queries, where each query determines the amodal segmentation mask for an object and its layer order throughout the video. (ii) we introduce a layer-based synthetic pipeline to generate multi-object training sequences that cover the variability of the target real sequences in terms of: multiple deforming objects, overlapping objects in some frames, moving cameras, and non-static backgrounds. (iii) We demonstrate that the model trained on flow from the synthetic videos is able to learn

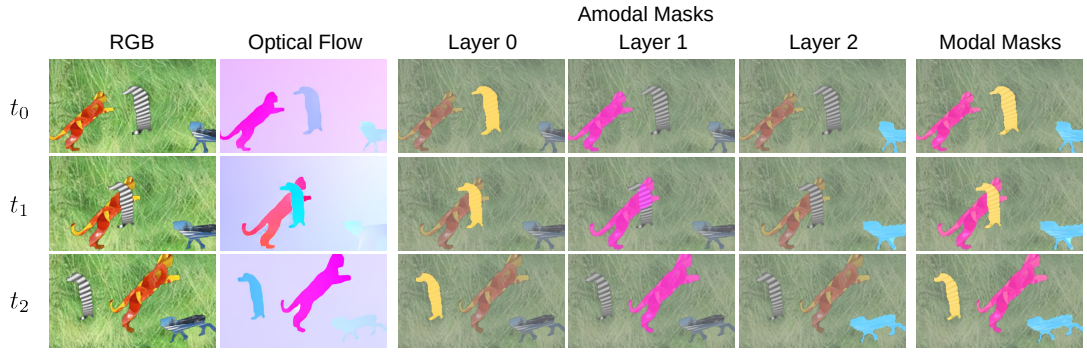


Figure 2.1: **Multiple-object discovery and segmentation.** We develop a layered model for segmenting and tracking multiple objects by their motion, even under mutual occlusion. The model takes only optical flow as its input, and predicts an **amodal** segmentation (whole physical structure) for each object, and associating an object to the same layer throughout the video. The resultant **modal** masks (visible parts) can be constructed by overlaying layers based on the inferred depth ordering (in this case, layer 0-1-2 from front to back). From left to right: RGB sequence, estimated (imperfect) optical flows, amodal mask inferences from our model and resultant modal masks.

object permanence and temporal shape constancy – even if an object is partially occluded in many frames, the model is still able to predict an amodal mask. (iv) Finally, we evaluate the model (trained only on synthetic sequences) on multiple real video segmentation benchmarks, DAVIS, MoCA, SegTrack, and FBMS-59; and demonstrate strong Sim2Real generalisation, achieving superior segmentation performance compared to other human-label-free methods. The addition of a test-time adaptation to include appearance from a self-supervised DINO model boosts the performance further, even outperforming some supervised approaches that are finetuned on real data sequences.

2.2 The object-centric layered representation (OCLR) model

This section describes our object-centric layered representation model that ingests optical flow for the frames of a video, and outputs the segmentation masks and tracks of the moving objects it contains. The model also determines the number of moving objects and their depth order. Specifically, the model predicts amodal (*i.e.* complete) segmentation masks and associates each object with a layer. The layers are composed to form the modal segmentations in the observed optical flow. The

model is trained on synthetic sequences to learn temporal coherence and maintain the object shape even under occlusion, and as a consequence, develops a notion of object permanence.

2.2.1 Layered representation

For a given video with T frames, we can represent its motion by optical flows, $\mathcal{V}_{\text{flow}} = \{F_1, F_2, \ldots, F_T\}$, we aim to segment the moving objects in videos by adopting a layered model, where each object is segmented and consistently associated to one layer throughout the video:

$$\{(\hat{\mathcal{A}}_1, \hat{r}_1), \ldots, (\hat{\mathcal{A}}_T, \hat{r}_T)\} = \Phi(\mathcal{V}_{\text{flow}}) \tag{2.1}$$

where $\hat{\mathcal{A}}_i \in \mathbb{R}^{H \times W \times K}$ refers to the binary **amodal** segmentation of the i-th frame, together with a predicted layer depth $\hat{r}_i \in \mathbb{R}^K$. H, W refer to the input height and width, and K denotes the number of object layers. Note that, the layer depth value here does not have a true meaning, but only indicates the layer ordering. By definition, at all time steps, the same layer would remain empty or always bind to the **same** object throughout the video, *i.e.* objects can be effectively tracked by the considered layered representation.

Modal mask generation. Given the predicted amodal segmentation and layer depth, we describe the composition procedure to generate the **modal** segmentation for moving objects. In specific, we permute the index of layers (**amodal** segmentations) based on their depth values, such that, $\hat{\mathcal{A}}^{k-1}$ is always in front of $\hat{\mathcal{A}}^k$. Note that, pixels in amodal masks defined here are binary-valued, *i.e.* either 0 or 1. For the i-th frame, its modal segmentation can thus be computed via an iterative, front-to-back blending procedure :

$$\hat{\mathcal{M}}_i^k = (1 - \alpha_i^k) \odot \hat{\mathcal{A}}_i^k \tag{2.2}$$

where $\alpha_i^k = \text{clip}(\alpha_i^{k-1} + \hat{\mathcal{M}}_i^{k-1}, 0, 1)$, denoting an accumulated opacity layer with $\alpha_i^0 = 0$.

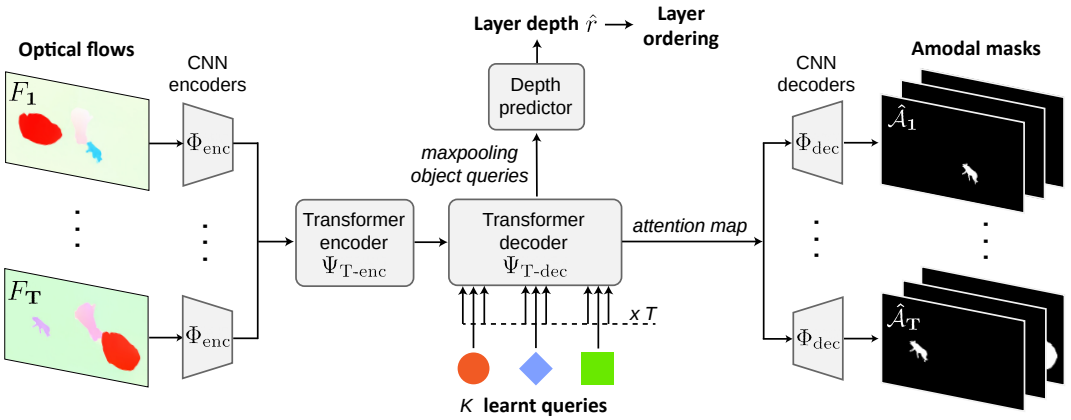


Figure 2.2: **OCLR model architecture.** The model is a U-Net architecture with a Transformer bottleneck. It takes optical flow as input and extracts spatial features by CNN encoders. A transformer encoder jointly processes spatio-temporal features across all frames, followed by a transformer decoder that determines the layer representations. Each learnable query vector in the transformer decoder is associated to one object and used to infer its layer depth. Additionally, the cross-attention maps from the last transformer decoder are extracted and upsampled by CNN decoders to infer amodal segmentation for the moving objects. Note, the skip connections from the CNN encoders to decoders are not shown.

2.2.2 Architecture

In order to accommodate the layered representation, we introduce an architecture that takes optical flows as input, and outputs a set of layers and their depths, see Figure 2.2. We adopt a U-Net architecture with a Transformer-based bottleneck, the queries in transformer decoder play the role as layers in our proposed representation, each is either empty or encoding the amodal shape of one moving object along the video.

CNN backbone. Given a sequence of optical flow inputs, a U-Net encoder is used to compute frame-wise features, $\mathcal{V} = \{v_1, \ldots, v_T\} = \{\Phi_{\text{enc}}(F_1), \ldots, \Phi_{\text{enc}}(F_T)\}$, where $v_i \in \mathbb{R}^{h \times w \times c}$ refers to the feature maps, with height h, width w and number of feature channels c. Following [Yang et al., 2021a], we use Instance Normalisation (IN) [Ulyanov et al., 2016] after each convolution, which encourages the separation of foreground motions from background.

Transformer encoder. We flatten the outputs from the ConvNets in both temporal and spatial dimensions, and inject the positional information by adding a set of learnable spatio-temporal embeddings. As a consequence, the video features are converted into a sequence of vectors, and passed into Transformer Encoder to

model the temporal relationship across multiple frames.

Transformer decoder. We pass the vector sequence to transformer decoder, where K learnable embeddings are used as object queries for individual layers. We broadcast the queries along the temporal dimension, and supply positional information with learnable temporal embeddings, $\mathcal{Q}_{\text{obj}} \in \mathbb{R}^{TK \times c}$. The queries that only differ in temporal embeddings will be associated to the same moving object along the video.

As outputs from transformer decoder, we extract the last cross-attention maps and recover their spatial dimensions, $\rho \in \mathbb{R}^{TK \times h \times w \times \text{nheads}}$, which will be progressively upsampled with skip connections from the ConvNets encoders, resulting in binary amodal segmentation for the object queries in each frame, $\{\hat{\mathcal{A}}_1, \ldots, \hat{\mathcal{A}}_T\} = \Phi_{\text{dec}}(\rho)$.

Layer ordering. Apart from amodal segmentations, the model will also infer the depth of the layers. We assume a global layer ordering that remains fixed across frames. As shown in Figure 2.2, output object queries from transformer decoder are maxpooled along the temporal dimension and passed through two feed-forward layers, to get the output $\hat{r} \in \mathbb{R}^{1 \times K}$, indicating the depth for each of the K layers. Intuitively, this is equivalent to reason the layer depth based on one selected key occlusion frame from the whole sequence. By defining the lowest depth value as corresponding to the top layer, we then obtain a depth order prediction between layers. Note that, for layers that are empty or contain non-interacting objects, the ordering is not strictly defined, thus any prediction should not be penalised.

2.2.3 Training objectives

In this section, we describe the training procedure for the proposed architecture, with groundtruth supervision on both amodal segmentation and layer ordering.

Amodal segmentation loss. In addition to the conventional pixelwise binary classification ($\mathcal{L}_{\text{bce}}$) on each of the K layers, we also adopt a pixelwise classification loss defined only on a strip region around object boundaries, termed as the boundary loss $\mathcal{L}_{\text{bound}}$. This additional loss further emphasises the boundary regions, forcing the model to maintain shape information. During training, since the predicted amodal segmentations are permutation invariant, we use Hungarian

matching to match their corresponding ground truth [Carion et al., 2020], and then compute the amodal loss :

$$\mathcal{L}_{\text{amodal}} = \frac{1}{KT} \sum_{k=1}^{K} \sum_{i=1}^{T} \left(\lambda_{\text{bce}} \cdot \mathcal{L}_{\text{bce}}(\hat{\mathcal{A}}_i^k, \mathcal{A}_i^k) + \lambda_{\text{bound}} \cdot \mathcal{L}_{\text{bound}}(\hat{\mathcal{A}}_i^k, \mathcal{A}_i^k) \right) \tag{2.3}$$

where λ_{bce} and λ_{bound} are loss weights.

Layer ordering loss. For layers with mutual occlusions, we train the order prediction module with ranking losses on all $K!$ pairs of layers, formally,

$$\mathcal{L}_{\text{order}} = \frac{1}{K!} \sum_{i \neq j} -\log \left(\sigma \left(\frac{\hat{r}_i - \hat{r}_j}{\tau} \right) \right) \tag{2.4}$$

where σ is the sigmoid function with a temperature factor τ, and the ground-truth ordering indicates a relative depth relationship $\hat{r}_i > \hat{r}_j$ between layer i and j.

Total loss. The overall loss for training is a combination of amodal $\mathcal{L}_{\text{amodal}}$ and ordering loss $\mathcal{L}_{\text{order}}$:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{amodal}} + \lambda_{\text{order}} \cdot \mathcal{L}_{\text{order}} \tag{2.5}$$

where λ_{order} is the weight factor for the layer ordering loss.

Discussion. As discussed in early sections, our model is trained to infer amodal (*i.e.* complete) object segmentations and layer depth, even though the object itself may sometimes be invisible in the flow field due to being temporarily static, or partially occluded by other objects. As a result, the model is forced to always maintain the object shape internally, *i.e.* a notion of object permanence, as well as infer mutual occlusions from the visible flows. In the next section, we will introduce a pipeline for simulating video sequences, where groundtruth amodal segmentations and layer orderings can be generated at scale to train our model.

2.3 Synthetic dataset generation

We introduce a scalable pipeline to synthesize videos to train the proposed layered object-centric model. The pipeline builds on the method of Lamdouar *et al.* [Lamdouar et al., 2021], but extends it to simulating videos with multiple objects and complex inter-object occlusions through a layer composition.

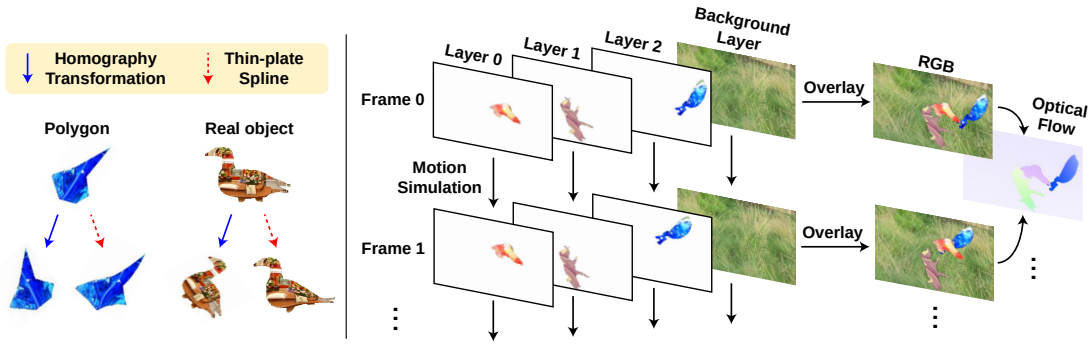


Figure 2.3: **Overview of the synthetic data pipeline.** Left: Motion simulations of polygon and real object sprites based on homography transformations (including spatial translations) and thin-plate spline. Right: Layer composition of frames with independently moving objects.

As shown in Figure 2.3, we first simulate independent motions for foreground objects and backgrounds within each layer. To construct a frame, we overlay the layers via an iterative binary blending procedure, and an occlusion occurs when objects in different layers reside in the same spatial position, similar to the idea discussed in Sect. 2.2.1. The proposed pipeline yields rich information at frame level, including RGB appearances, groundtruth optical flows, and modal and amodal object segmentations. Additionally, we can also compute optical flows for the RGB sequences with an off-the-shell flow estimator, for example, RAFT [Teed and Deng, 2020]. In the following sections, we describe some key procedures for video simulation, and leave more technical details to the Supplementary Material.

Foreground objects. We generate opaque 2D sprites with random shapes and RGB textures as foreground objects. In detail, we adopt two major shapes, namely polygons, and real object sprites. The polygon shapes can be either convex or non-convex, with different numbers of sides. A minimum distance between vertices is also defined to better support vertex-based deformations. On the other hand, the real object masks are directly sourced from the silhouettes in the YouTubeVOS dataset [Xu et al., 2018]. These generated objects are then applied with textures sampled from the PASS dataset [Asano et al., 2021], which is a large-scale image dataset with human identifiable information removed.

Non-rigid object transformations. To simulate non-rigid motions, we apply both homography transformations and thin plate splines to the objects. Thin plate splines define a non-linear coordinate transformation with a set of control

points. For polygon-shaped objects, we consider their vertices as control points, and spatially perturb a subset of the vertices by distances $d_i \in (0, D_i/2)$, where $D_i = \min(|\mathbf{P}_i - \mathbf{P}_j|, \forall j)$, defines the minimum distance between the perturbed point $\mathbf{P}_i$ and any other points $\mathbf{P}_j$. For real object silhouettes, a uniform grid is distributed over the object, and the grid points are defined as control points, with D_i being the distance between adjacent grid points.

Backgrounds and artificial occluders. To simulate the camera motions, we randomly sample images from the PASS dataset, and apply random homography transformations and color jitterings. Additionally, we replace backgrounds with real videos cropped from copyright-free sources to simulate the complex background motions and depth variations. For cases with objects being occluded, we introduce artificial occluders superimposed as the top layer. These occluders are designed to have the same motion as the background.

Summary. The proposed simulation pipeline enables an arbitrary number of videos to be generated, together with multiple types of groundtruth annotation; for example, amodal segmentation and layer ordering. We prepare 4k synthetic sequences (around 120k frames) for training, the video sequences contain 1, 2, or 3 objects in equal proportions. In the following, we train all models on this synthetic video dataset unless otherwise specified.

2.4 Experiments

2.4.1 Datasets

To evaluate our multi-layer model, we benchmark on several popular datasets for video object segmentation tasks. A brief overview of the datasets is given below, with full details in the Supplementary Material.

For single object video segmentation, we evaluate the model on DAVIS2016 [Perazzi et al., 2016], SegTrackv2 [Li et al., 2013], FBMS-59 [Ochs et al., 2014] and MoCA [Lamdouar et al., 2020]. Note that, despite the fact that multiple objects may be annotated, the community often treats SegTrackv2 and FBMS-59 as a benchmark for single object segmentation [Jain et al., 2017; Yang et al., 2019c] by

grouping objects in the foreground.

To benchmark motion-based segmentation for multiple objects, we introduce a synthetic validation dataset (Syn-val) and a curated dataset (DAVIS2017-motion). The former is generated with the same parameters as our synthetic training set (Sect. 2.3), containing over 300 multi-object sequences (around 10k frames) with $1, 2, 3$ objects at equal proportions, and controllable occlusions for evaluating modal and amodal segmentations in the ablation studies. Moreover, as objects in common motion cannot be distinguished purely from motion cues, we re-annotate the original DAVIS2017 [Pont-Tuset et al., 2017] dataset by grouping jointly moving objects to form a new DAVIS2017-motion dataset. We will release our modified annotations for future research.

2.4.2 Evaluation metrics

Following the common practice [Perazzi et al., 2016], we adopt region similarity ($\mathcal{J}$), *aka.* intersection-over-union, in single object segmentation tasks. While for multiple object segmentation, we additionally consider the contour similarity ($\mathcal{F}$) [Martin et al., 2004]. As for MoCA dataset, since only bounding box annotations are provided, we follow the same metric used in [Lin et al., 2014; Yang et al., 2021a], and report a mean object detection success rate averaged over different IoU thresholds $\{0.5, 0.6, 0.7, 0.8, 0.9\}$.

2.4.3 Implementation details

Training details. As for pre-processing, we use RAFT to estimate optical flows at ± 1 frame gaps (except for FBMS, where ± 3 frame gaps are adopted to compensate for small motion), and resize the obtained flows to 128×224. During training, we split the video sequences into 30 frames per sample, each input frame is first encoded by a U-Net encoder into a feature map with $1/16$ of its original spatial resolution, and passed to the transformer bottleneck. We use $K = 3$ learnable object queries, associating to 3 independent foreground layers. The model is trained by the Adam optimizer [Kingma and Ba, 2015] with a learning rate linearly warmed up to 5×10^{-5} during 40k iterations, and decreased by half every 80k iterations.

RGB-based test-time adaptation. To alleviate the drawbacks from using purely flow information (*e.g.* stationary objects over a long period of time), we investigate the benefits of test-time adaptation with appearance features. For this we use the self-supervised DINO-pretrained vision transformer [Caron et al., 2021] (ViT-S/8, patch sizes 8×8). The model is adapted by using the per-frame object masks obtained from the flow-based model predictions as noisy annotations. These annotations are used to finetune the last two layers of the vision transformer by supervised contrastive learning.

The finetuned model for each sequence is then applied for mask propagation in the same manner as [Jabri et al., 2020]. Instead of propagating from the first frame of the sequence, we formulate a heuristic selection process that picks a single key frame based on temporal coherence between OCLR segmentation predictions. Object masks in this selected frame are then bi-directionally propagated across the whole sequence. Apart from the starting frame information, flow-predicted masks in other frames are also selectively introduced during the mask propagation process as a form of dynamic refinement. Finally, we adopt CRF as a post-processing step to refine the resultant mask predictions. More technical details can be found in the Supplementary Material.

2.4.4 Ablation study

Here, we present a series of ablation studies on training details and pipeline for data simulation however, due to the space limitation, we can only summarise some key findings, we refer the reader to Supplementary Materials for all details.

Table 2.1: **Settings for training parameters.** **IN**: Instance Normalisation; **Amodal**: Training on amodal mask (vs. modal mask); λ_{bound}: weight on boundary loss; T: number of input frames. Syn-Val ($\mathcal{M}|\mathcal{A}$) corresponds to modal and amodal results on synthetic dataset.

	Training Settings				$\mathcal{J}$ (Mean) ↑		
Model	IN	Amodal	λ_{bound}	T	Syn-Val ($\mathcal{M}$ | $\mathcal{A}$)	DAVIS2016	DAVIS2017-motion
Ours-A	✗	✓	0.2	30	83.5 | 83.0	67.6	48.7
Ours-B	✓	✗	0.2	30	81.1 | 76.9	69.2	50.5
Ours-C	✓	✓	0.2	30	**85.6** | **84.7**	**72.1**	**54.5**
Ours-D	✓	✓	0.2	15	82.8 | 83.0	71.3	53.5
Ours-E	✓	✓	0	30	80.9 | 81.6	71.5	54.1

As shown in Table 2.1, we can make the following observations: *First*, Instance

Normalisation in CNN encoder is indispensable for training the network, as indicated by Ours-A vs. Ours-C; *Second*, training on only modal mask degrades model performance on all datasets, especially on Syn-Val with multiple objects and occlusions, suggesting that explicit amodal supervision are critical for learning object permanence, as indicated by Ours-B vs. Ours-D; *Third*, a longer temporal input (Ours-C vs. Ours-D) tends to result in slightly higher overall performance; *Fourth*, applying boundary loss (Ours-C vs. Ours-E) leads to a noticeable performance boost. This validates our assumption that focusing on object boundaries can help the model to learn the information regarding object shapes and layer orders from optical flows.

2.4.5 Single object video segmentation

In this section, we compare our model with state-of-the-art methods on various single object segmentation benchmarks. Note that, we mainly compare with the self-supervised approaches, as both lines share the same spirit in the sense that training *does not* require any manual annotation, nor fine-tuning on the target dataset. As shown in Table 2.2, our flow-only model demonstrates superior performance over all other human-label-free approaches. Figure 2.4 provides qualitative illustrations of the model, in comparison with other state-of-the-art methods. It can be seen that our predictions are temporally consistent and not affected by noticeable background distractors in flow signals. Furthermore, our model is capable of handling complex scenarios including heavy object deformations and occlusions. The other methods are not able to maintain the object shape consistently, and consequently their performance is weaker.

The RGB-based test-time adaptation gives a further performance boost on most of the benchmarks, and is even competitive to some supervised approaches that have been finetuned on the target video data. Note that the test-time adaptation is actually detrimental to performance on MoCA. This is not unexpected though, as MoCA has many camouflage sequences where the objects are not visually distinguishable in appearance from their background environment, and motion provides crucial cues for discovering them.

Table 2.2: **Quantitative comparison on single object video segmentation benchmarks.** "HA" stands for human annotations, and "SR" refers to the detection success rate on MoCA. In column Sup. (supervision), "None", "Syn.", "Real" represent self-supervision, synthetic data supervision, and real data supervision, respectively. ***Bold*** represents the state-of-the-art performance (excluding our test-time adaptation results, which are labelled as *blue* instead). For FBMS, results outside/inside brackets correspond to flow inputs with $\pm 1/\pm 3$ frame gaps.

Model	Training Settings				$\mathcal{J}$ (Mean) ↑			SR (Mean) ↑
	HA	Sup.	RGB	Flow	DAVIS2016	SegTrackv2	FBMS-59	MoCA
NLC [Faktor and Irani, 2014]	✗	None	✓	✓	55.1	67.2	51.5	–
CIS (w. post-process.) [Yang et al., 2019c]	✗	None	✓	✓	71.5	62.0	63.5	0.363
Motion Grouping [Yang et al., 2021a]	✗	None	✗	✓	68.3	58.6	53.1	0.484
SIMO [Lamdouar et al., 2021]	✗	Syn.	✗	✓	67.8	62.0	–	0.566
OCLR (flow-only)	✗	Syn.	✗	✓	**72.1**	**67.6**	**65.4 (70.0)**	**0.599**
OCLR (test. adap.)	✗	Syn.	✓	✓	*80.9*	*72.3*	*69.8 (72.7)*	0.559
FSEG [Jain et al., 2017]	✓	Real	✓	✓	70.7	**61.4**	68.4	–
COSNet [Lu et al., 2019]	✓	Real	✓	✗	80.5	49.7	75.6	0.417
MATNet [Zhou et al., 2020]	✓	Real	✓	✓	82.4	50.4	**76.1**	**0.544**
D^2Conv3D [Schmidt et al., 2022]	✓	Real	✓	✗	**85.5**	–	–	–

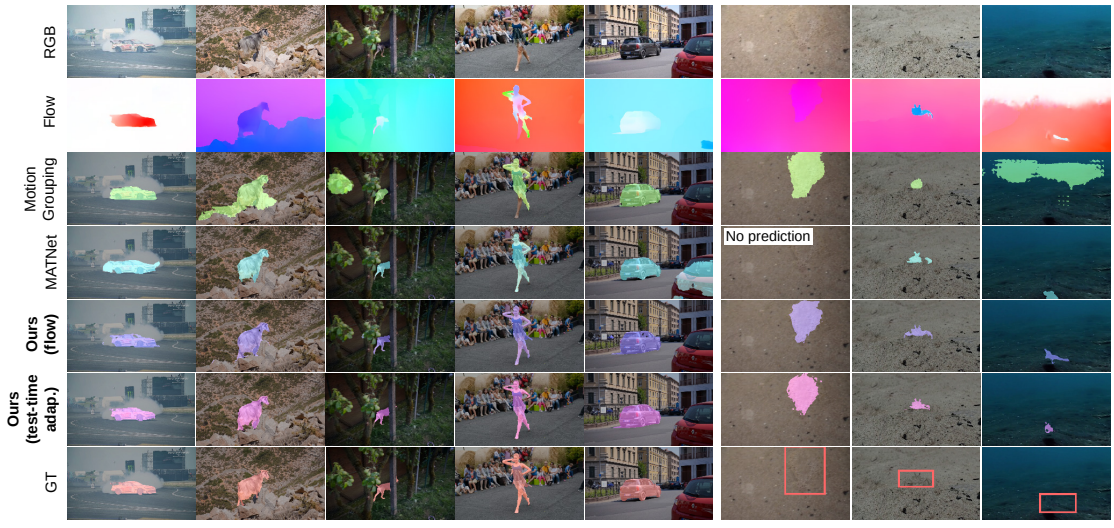


Figure 2.4: **Qualitative results of single object video segmentation on DAVIS2016 and MoCA.**

2.4.6 Multiple object video segmentation

To the best of our knowledge, no existing unsupervised approaches have reported performance segmenting multiple objects purely based on optical flow. Apart from re-running the original self-supervised Motion Grouping [Yang et al., 2021a] method (with three foreground queries) as a baseline on DAVIS2017-motion, we also train a variant of the Motion Grouping model directly supervised by our synthetic dataset, namely Motion Grouping (sup.). As an additional baseline, we train a Mask R-CNN [He et al., 2017] model with only optical flow inputs on our synthetic data, and refer to this as Mask R-CNN (flow-only). We also compare with existing semi-supervised approaches, where the first-frame ground-truth segmentation is provided, and the model only needs to propagate the masks through the video, thus an easier task than ours which requires to simultaneously discover the objects and track them.

As shown in Table 2.3, our proposed flow-only model outperforms both the Motion Grouping and Mask R-CNN baselines. This is because our OCLR model exploits a layered representation to maintain the object shape through the video, which enables to segment the objects that are under occlusion or having unnoticeable motion. In contrast, Motion Grouping only predicts per-frame segmentations,*i.e.* not forced to preserve temporal relations between objects. This conjecture can also be confirmed by the qualitative results in Figure 2.5, for example, despite the person and one dog (4th column) are not visible in the flow, our model still correctly recovers them. Moreover, after RGB-based test-time adaptations, the performance can be further boosted both quantitatively and qualitatively, for example, the boundary of the pig mask in 1st column, the person in the 5th column.

Table 2.3: **Quantitative comparison of multi-object video segmentation on DAVIS2017-motion.** Note that, the compared methods here are trained without using any human annotations during training, in particular, Motion Grouping (sup.), Mask R-CNN (flow-only) and OCLR models are supervised by only synthetic data, and other approaches are trained with self-supervision. ***Bold*** represents the state-of-the-art performance (excluding our test-time adaptation results, which are labelled as *blue* instead).

Model	Training settings			DAVIS2017-motion performance		
	1st-frame-GT	RGB	Flow	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}$ (Mean) $\uparrow$	$\mathcal{F}$ (Mean) $\uparrow$
Motion Grouping [Yang et al., 2021a]	✗	✗	✓	35.8	38.4	33.2
Motion Grouping (sup.)	✗	✗	✓	39.5	44.9	34.2
Mask R-CNN (flow-only)	✗	✗	✓	50.3	50.4	50.2
OCLR (flow-only)	✗	✗	✓	**55.1**	**54.5**	**55.7**
OCLR (test. adap.)	✗	✓	✓	64.4	65.2	63.6
CorrFlow [Lai and Xie, 2019]	✓	✓	✗	54.0	54.2	53.7
UVC [Li et al., 2019a]	✓	✓	✗	65.5	66.2	64.7
MAST [Lai et al., 2020]	✓	✓	✗	70.9	71.0	70.8
CRW [Jabri et al., 2020]	✓	✓	✗	73.4	72.9	74.1
DINO [Caron et al., 2021]	✓	✓	✗	**78.7**	**77.7**	**79.6**

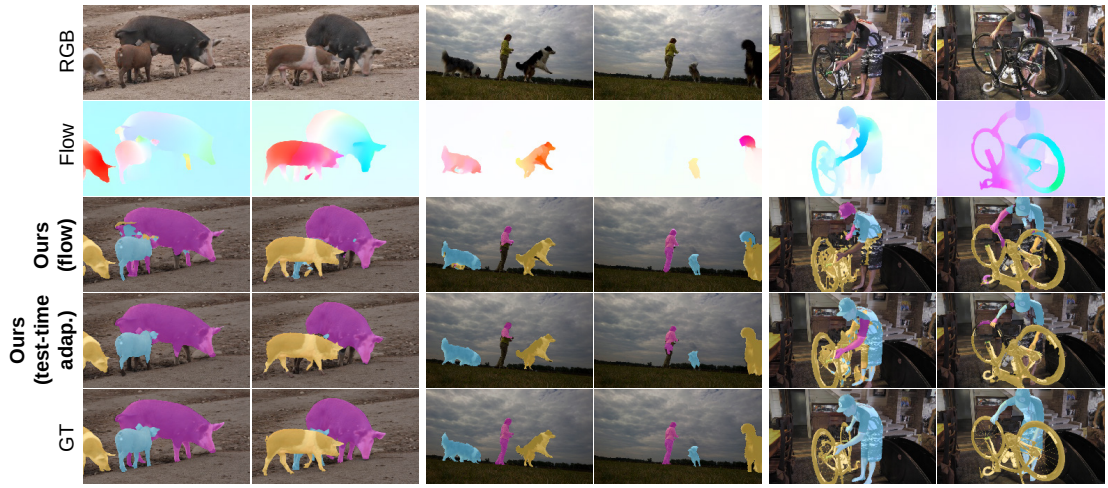


Figure 2.5: **Qualitative results of multi-object video segmentation on DAVIS2017-motion.**

2.5 Related work

Video Object Segmentation has been a longstanding task in computer vision, two protocols have attracted increasing interest in the recent literature [Brox and Malik, 2010; Ochs and Brox, 2011; Fragkiadaki et al., 2012; Papazoglou and Ferrari, 2013; Keuper et al., 2015; Bideau and Learned-Miller, 2016a; Jun Koh and Kim, 2017; Jain et al., 2017; Caelles et al., 2017; Pont-Tuset et al., 2017; Xu et al., 2018; Tokmakov et al., 2019; Dave et al., 2019; Oh et al., 2019; Voigtlaender et al., 2019; Vondrick et al., 2018; Maninis et al., 2018; Fan et al., 2019; Yang et al., 2019d; Lai and Xie, 2019; Lai et al., 2020], namely, semi-supervised video object segmentation (**semi-supervised VOS**), and unsupervised video object segmentation (**unsupervised VOS**). The former aims to re-localize one or multiple targets that are specified in the first frame of a video with pixel-wise masks, and the latter considers automatically segmenting the object of interest (usually the most salient one) from the background in a video sequence. Despite being called **unsupervised VOS**, in practice, the popular methods address such problems by supervised training on large-scale external datasets, this is in contrast to our proposed approach that does not rely on human annotations whatsoever.

Motion Segmentation focuses on discovering the *moving* objects in videos. In the literature, [Brox and Malik, 2010; Ochs and Brox, 2011; Keuper et al., 2015; Xie et al., 2019] proposed to cluster the pixels with similar motion patterns; [Tokmakov et al., 2017; Tokmakov et al., 2019; Dave et al., 2019] train deep networks to map the motions to segmentation masks. In [Yang et al., 2019c], adversarial training was adopted to leverage the independent motions between the moving object and its background; In [Bideau and Learned-Miller, 2016b; Bideau et al., 2018; Lamdouar et al., 2020], the authors propose to highlight the independently moving object by compensating for the background motion, either by registering consecutive frames, or explicitly estimating camera motion. In [Yang et al., 2021a], a Transformer-like architecture is used to reconstruct the input flows, and the segmentation masks can be generated as a side product. In contrast to the existing approaches that primarily focus on single moving object segmentation, we focus on segmenting **multiple** moving objects, even under mutual occlusions.

Layered representation was originally proposed by Wang and Adelson [Wang and Adelson, 1994], to represent a video as a composition of layers with simpler motions. Recently, the layer decomposition ideas have been adopted for novel view synthesis [Zhou et al., 2018c; Srinivasan et al., 2019], separating reflections and other semi-transparent effects [Alayrac et al., 2019b; Alayrac et al., 2019a; Gandelsman et al., 2019; Lu et al., 2020; Lu et al., 2021], or foreground/background estimation [Gandelsman et al., 2019]. Unlike these approaches that primarily focus on graphics applications, we propose a layered representation to handle the occlusion problems in multi-object discovery purely from their motions.

Object-centric representation decomposes the scenes into "objects", normally, visual frame reconstruction has been widely used as training objective, for example, IODINE [Greff et al., 2019] uses iterative variational inference to infer a set of latent variables recurrently, with each representing one object in an image. Similarly, MONet [Burgess et al., 2019] and GENESIS [Engelcke et al., 2020] also adopt multiple encoding-decoding steps, [Locatello et al., 2020; Kipf et al., 2022] propose a slot attention mechanism, which enables the iterative binding procedure. In this paper, we also adopt a Transformer variant, but focus on discovering objects in videos by motions.

Amodal segmentation refers to segmenting the complete shape of objects, including the invisible parts due to possible occlusions. Some existing approaches address this problem by applying human-estimated [Qi et al., 2019; Xiao et al., 2021; Sun et al., 2021] or synthetic [Hu et al., 2019] supervision, while other works generate training datasets with synthetic occluders [Li and Malik, 2016; Zhan et al., 2020; Ling et al., 2020]. Although not specifically targeting the amodal segmentation task, our work leverages the idea of amodal perception and grants the trained model a notion of object permanence via synthetic amodal mask supervision.

2.6 Discussion

We have achieved our design goal, in that the model is able to handle multiple objects, and their occlusions, and to predict depth ordering of their layers. Also, the

model demonstrates superior performance, both quantitatively and qualitatively, over prior methods that rely on zero human annotations. Nevertheless, there are of course some limitations and room for further improvements: *First*, the current method may fail in very challenging real-world scenarios such as heavy object deformations, complex mutual interactions, *etc.* Probably a more sophisticated data simulation pipeline with highly articulated objects or even 3D sprites would help to further reduce this Sim2Real domain gap. *Second*, our test-time adaptation has demonstrated a remarkable performance improvement benefiting from the combination between flow inferences and RGB correlations. This overcomes, to an extent, the problem of the flow-only model 'forgetting' the shape of objects when they are stationary for multiple frames. A further direction could naturally be to incorporate RGB information into our flow-based network in pre-training. *Third*, the model currently uses a global depth ordering for the sequence, so it cannot handle situations such as order altering and mutual occlusions. Potential future studies could work on a more sophisticated layered model by focusing on pair-wise relationships between objects.

Despite these limitations, the approach has convincingly demonstrated both the value of inferring amodal segmentation masks, in order to handle occlusions, and the possibility of training such models entirely on synthetic sequences.

Acknowledgements

This research is funded by EPSRC Programme Grant VisualAI EP/T028572/1, a Royal Society Research Professorship RP\R1\191132, and a Clarendon Scholarship. We thank Tengda Han for proof-reading.

Appendices

Appendices for this chapter can be found in the online version of the paper.[1]

[1] https://www.robots.ox.ac.uk/~vgg/research/oclr/

Statement of authorship

A statement of authorship for this paper is provided in Appendix A.

Chapter 3

Appearance-Based Refinement for Object-Centric Motion Segmentation

This paper was published at the European Conference on Computer Vision (ECCV), 2024.

Abstract

The goal of this paper is to discover, segment, and track independently moving objects in complex visual scenes. Previous approaches have explored the use of optical flow for motion segmentation, leading to imperfect predictions due to partial motion, background distraction, and object articulations and interactions. To address this issue, we introduce an appearance-based refinement method that leverages temporal consistency in video streams to correct inaccurate flow-based proposals. Our approach involves a sequence-level selection mechanism that identifies accurate flow-predicted masks as exemplars, and an object-centric architecture that refines problematic masks based on exemplar information. The model is pre-trained on synthetic data and then adapted to real-world videos in a self-supervised manner, eliminating the need for human annotations. Its performance is evaluated on multiple video segmentation benchmarks, including DAVIS, YouTubeVOS, SegTrackv2, and FBMS-59. We achieve competitive performance on single-object segmentation, while significantly outperforming existing models on the more challenging problem of multi-object segmentation. Finally, we investigate the benefits of using our model as a prompt for the per-frame Segment Anything Model.

3.1 Introduction

Given a video of a complex visual scene, we consider the task of discovering, segmenting and tracking the independently moving objects. Recent approaches for this task have built on two key advantages of optical flow inputs [Yang et al., 2019c; Yang et al., 2021a; Lamdouar et al., 2021; Xie et al., 2022]. *First*, segmentation masks can be directly obtained from discontinuities in the flow field; *Second*, optical flow eliminates the domain gap between synthetic and real videos, making it possible to train models on synthetic sequences and apply them directly to real sequences without incurring a significant Sim2Real penalty.

While optical flow has these advantages, there are also some drawbacks to relying solely on flow. *First*, multi-object motion patterns in real-world videos *are complex* – they can include objects that have no motion relative to the camera in some frames (and so no discontinuities), and objects can be articulated, can

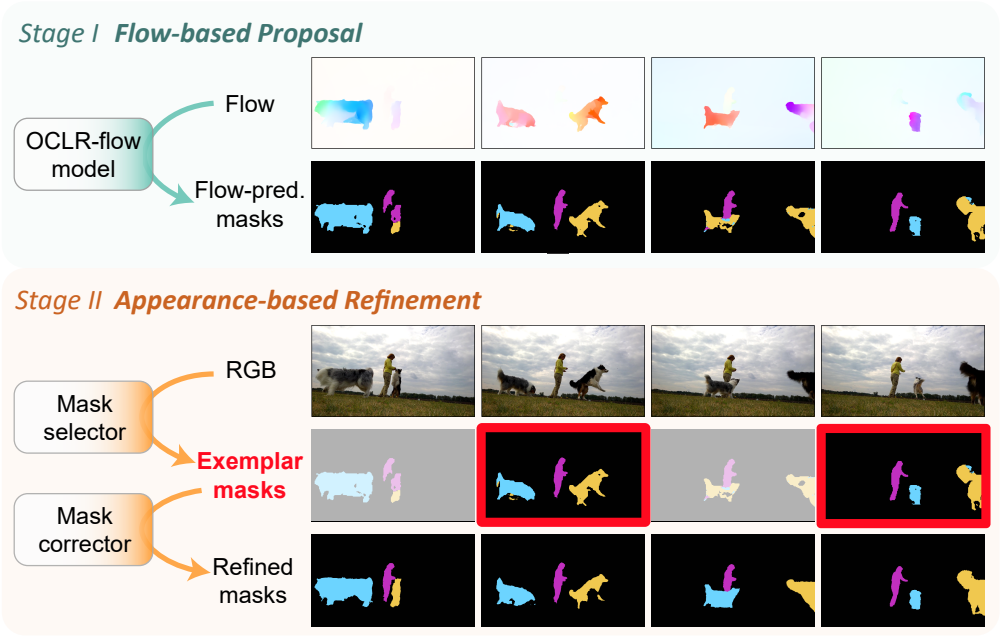


Figure 3.1: **Overview of our Multi-Object Motion Segmentation method.** The method starts with proposing object masks based on optical flow inputs (*i.e.* **flow-based proposals**), followed by an **appearance-based refinement** of flow-predicted masks. The latter stage relies on a selection-correction procedure, where high-quality exemplar masks are selected to guide the correction of other masks. Mask selection involves picking high-quality proposal masks based on both temporal coherence across mask shapes and semantic consistency with object appearances. Mask correction involves using the selected masks (the exemplars) to guide an appearance-based mask correction process in an object-centric model. The success of the method can be seen in the bottom row where correct segmentation masks are predicted, overcoming the deficiencies of the flow-predicted masks in Stage 1.

interact with other objects, and be partly or entirely occluded. Additionally, the background can have distractor motions, which can lead to ambiguous groupings and segmentations. *Second*, unlike the appearance stream where there is a consistency of texture and colour over the video, video constancy is not guaranteed in flow fields. This makes it challenging to establish long-term correlations and build object permanence, which can be crucial in tracking objects over time.

In this paper, to address the drawbacks of flow, we incorporate temporally-consistent appearance information by using a *two-stream* approach [Simonyan and Zisserman, 2014b; Feichtenhofer et al., 2016; Faktor and Irani, 2014] for the task of object discovery, segmentation, and tracking in videos. The motion stream (flow) is used for object discovery, while the appearance stream is used for maintaining consistency in segmentation and tracking, leveraging long-term correlations that arise from the persistence of appearance.

Our approach is depicted in Figure 3.1 and consists of two stages. In the first stage, we use an off-the-shelf flow-based motion segmentation model, OCLR [Xie et al., 2022]. As noted above, the flow can be used to discover moving objects (and has little Sim2Real gap), but this is as far as OCLR-flow can go – the model generates proposals for segmentation masks in *each* frame, but these vary in quality due to indistinct and partially occluded object motion. Appearance can give stronger segmentations, yet faces a significant Sim2Real gap. The novelty in this paper lies in our solution to this problem – "How can appearance information be used to improve flow-based segmentation?" – with three distinct innovations that form the second stage of the approach: (i) we propose a sequence-level mask *selection* mechanism, with DINO features [Caron et al., 2021] employed for establishing temporal consistency; (ii) the selected masks (that we treat as exemplars) support mask *correction*, where we employ a combination of direct mask supervision and motion-initialised semantic reconstruction for each frame; (iii) a self-supervised adaptation technique is introduced to effectively address the Sim2Real gap (that has some similarities with [Wang et al., 2023a]). This selection-correction procedure enables accurate mask predictions under complex motion scenarios, including partial object motion, background distraction, object articulation, and interaction, *etc.*

In summary, we make the following five contributions:

1. We introduce **a sequence-level selection mechanism** designed to globally identify exemplar object masks from flow-based segmentation predictions over a long temporal range.

2. We propose **an object-centric mask correction network** that incorporates both local (*i.e.* short-term) and global (*i.e.* exemplar) information into motion-initialised object queries. Additionally, we investigate a unique loss formulation that combines direct mask supervision with motion-aware semantic reconstruction.

3. We follow **a Sim2Real training procedure** that relies on *zero human annotation*: The framework is synthetically pre-trained, followed by a self-supervised training scheme designed for simple yet effective domain adaptation to real-world video sequences.

4. Our model is evaluated on multiple motion segmentation benchmarks, including DAVIS, YouTubeVOS, SegTrackv2, and FBMS-59, demonstrating competitive performance on single-object segmentation, while significantly outperforming existing approaches under challenging multi-object scenarios.

5. In further exploration, we use our video-level predictions as prompts for the Segment Anything Model (SAM) [Kirillov et al., 2023]. This reveals that the object-centric consistency surpasses direct per-frame segmentation, also highlighting the complementary improvements that SAM could bring to our results.

3.2 Related work

Motion Segmentation aims to discover and segment objects from motion cues in videos. Many prior works [Brox and Malik, 2010; Ochs and Brox, 2011; Keuper et al., 2015; Xie et al., 2019] approach this problem by clustering pixels based on similar motion patterns; Other studies [Bideau and Learned-Miller, 2016b; Bideau et al., 2018; Mahendran et al., 2018; Meunier et al., 2023] apply different motion models to explain the composition of optical flows and yield segmentation predictions. Innovations in model architectures [Tokmakov et al., 2019; Tokmakov et al., 2017; Dave et al., 2019] and training strategies [Yang et al., 2019c] also lead to significant progress in motion segmentation accuracy. Yang et al. [Yang et al., 2021a] apply the Slot Attention mechanism [Locatello et al., 2020] to reconstruct optical flows based on object slots. Lamdouar et al. [Lamdouar et al., 2021] and Xie et al. [Xie et al., 2022] explore a Sim2Real protocol to train a Transformer-based architecture that aggregates flow information in a temporally consistent manner.

Recent research [Faktor and Irani, 2014; Bao et al., 2022; Ye et al., 2022] has also focused on incorporating appearance information with motion cues. Choudhury et al. [Choudhury et al., 2022] and Ponimatkin et al. [Ponimatkin et al., 2023] leverage spectral clustering on appearance features to improve segmentation accuracy. DyStaB [Yang et al., 2021c] trains dynamic and static models via an iterative bootstrapping strategy. Our method follows a similar two-stream idea and targets a more challenging multi-object motion scenario.

Video Object Segmentation (VOS) focuses on obtaining pixel-level masks of one or multiple salient objects in videos. As defined by several benchmark datasets [Perazzi et al., 2016; Caelles et al., 2019], video object segmentation tasks can be further divided into two main protocols, namely semi-supervised VOS [Vondrick et al., 2018; Lai and Xie, 2019; Lai et al., 2020; Miao et al., 2022; Wang et al., 2019b; Jabri et al., 2020; Caron et al., 2021] and unsupervised VOS [Lu et al., 2019; Ventura et al., 2019; Cho et al., 2019; Li et al., 2018; Luiten et al., 2020]. The former provides first-frame ground-truth annotation and aims to recover segmentation masks for the remaining frames, while the latter automatically discovers and tracks salient objects from raw videos. Albeit referred to as "unsupervised", some previous studies rely on models that are pre-trained using manual labels, whereas our proposed method does not rely on any human annotations.

Sequence-level frame selection aims to pick the most informative frames from a video sequence, enhancing efficiency in downstream applications. This idea is widely explored across various computer vision domains, including action recognition [Raptis and Sigal, 2013; Zhu et al., 2016; Price and Damen, 2020], video retrieval [Zhao et al., 2022; Buch et al., 2022], *etc.* In video segmentation, BubbleNet [Griffin and Corso, 2019] employs this mechanism to select a single RGB frame that initiates mask propagation for semi-supervised VOS, while Yin et al. [Yin et al., 2021] extends this idea to interactive VOS. Wu et al. [Wu et al., 2020] introduce a Memory Selection Network for memory-based mask propagation. Recent studies [Kuznetsova et al., 2021; Delatolas et al., 2024; Bekuzarov et al., 2023] also utilise temporal selection to propose key frames for efficient segmentation annotations. Unlike prior works that focus on RGB frame selections via confidence score estimations, our selection network regards error map predictions as a proxy task to guide the exemplar **mask** selection for **each object**.

Two-stream approaches. The idea originates from the two-stream hypothesis in neuroscience [Schneider, 1969; Goodale and Milner, 1992; Norman, 2002], which suggests that the human visual cortex processes high temporal frequency motion information in the dorsal stream, and slower temporal frequency appearance information in the ventral stream. This two-stream (motion & appearance) approach

has been successfully applied to tasks such as action recognition [Simonyan and Zisserman, 2014b; Feichtenhofer et al., 2016; Ma et al., 2018; Feichtenhofer et al., 2019] and VOS [Faktor and Irani, 2014; Yang et al., 2019d; Zhou et al., 2020; Yang et al., 2021c; Yu et al., 2022; Miao et al., 2022; Liu et al., 2023].

Object-centric representation decomposes complex scenes as objects (and the background components) based on reconstruction cues. Recently, unsupervised object-centric representation learning for both static scenes [Greff et al., 2016; Eslami et al., 2016; Greff et al., 2019; Burgess et al., 2019; Engelcke et al., 2020; Bear et al., 2020] and videos [He et al., 2019; Kosiorek et al., 2018; Kabra et al., 2021] has drawn increasing research interest. Slot Attention [Locatello et al., 2020] proposes a fixed number of slots that compete for explaining parts of the input image through an iterative binding process. This idea is later widely adopted in video object representation learning [Veerapaneni et al., 2019; Jiang et al., 2020; Weis et al., 2021; Kipf et al., 2022; Elsayed et al., 2022].

Early unsupervised object-centric learning research mainly focuses on synthetic datasets, while more recent works achieve object segmentation and tracking on real-world videos. Fan et al. [Fan et al., 2023] and Aydemir et al. [Aydemir et al., 2023] achieve an object-centric video representation by employing reconstruction of the self-supervised visual features, thereby extending the method of Seitzer et al. [Seitzer et al., 2023] from real-world images to videos.

3.3 Appearance-based selection and correction

Given an off-the-shelf multi-object motion segmentation model, we aim to refine the imperfect predictions from flow-only signals, by exploiting the temporal consistency in video appearance streams.

3.3.1 Problem scenario

We begin with a video sequence of T frames and their corresponding optical flow, denoted as $\mathcal{V}_{\text{rgb}} = \{I_1, I_2, \ldots, I_T\}$ and $\mathcal{V}_{\text{flow}} = \{F_1, F_2, \ldots, F_T\}$, respectively. We adopt OCLR-flow [Xie et al., 2022] as an off-the-shelf multi-object discovery model, that takes in optical flow as input and outputs multi-object segmentation across

different frames. The model is defined as follows:

$$\{\hat{\mathcal{M}}_1^k, \ldots, \hat{\mathcal{M}}_T^k\}_{k=1}^K = \Phi_{\text{OCLR-flow}}(\mathcal{V}_{\text{flow}}) \tag{3.1}$$

$\hat{\mathcal{M}}_i^k \in \mathbb{R}^{H \times W \times 1}$ represents the flow-predicted segmentation mask for the k^{th} object in frame i, where H and W denote the height and width of the output mask. The masks with the same superscript either correspond to a unique object along the video sequence (*i.e.* tracking one object), or are empty.

Our goal is to refine motion segmentation by leveraging the video appearance stream, specifically, it involves selecting a set of high-quality exemplar segmentation masks and refining them by exploiting the temporal consistency in the appearance stream. To achieve this, we use the flow-predicted masks as a given prior and apply the following model:

$$\{\tilde{\mathcal{M}}_1^k, \ldots, \tilde{\mathcal{M}}_T^k\}_{k=1}^K = \Phi_{\text{corrector}} \circ \Phi_{\text{selector}}(\mathcal{V}_{\text{RGB}}; \{\hat{\mathcal{M}}_1^k, \ldots, \hat{\mathcal{M}}_T^k\}_{k=1}^K) \tag{3.2}$$

where $\tilde{\mathcal{M}}_i^k \in \mathbb{R}^{H \times W \times 1}$ denotes the refined mask output for the k^{th} object in frame i. The selection network (Φ_{selector}) is applied to identify high-quality exemplar masks, followed by a correction network ($\Phi_{\text{corrector}}$) to conduct further mask refinements.

3.3.2 Appearance-based mask selection

We propose a mask selector that takes the flow-predicted masks and RGB frames as inputs, to identify a set of high-quality exemplar masks **independently** for each object k. Formally,

$$(\hat{\mathcal{E}}_1^k, \ldots, \hat{\mathcal{E}}_T^k) = \Phi_{\text{selector}}(\mathcal{V}_{\text{RGB}}; \hat{\mathcal{M}}_1^k, \ldots, \hat{\mathcal{M}}_T^k) \longrightarrow (\hat{\mathcal{M}}_{e1}^{\ k}, \ldots, \hat{\mathcal{M}}_{et}^{\ k}) \tag{3.3}$$

where $\hat{\mathcal{E}}_1^k, \ldots, \hat{\mathcal{E}}_T^k$ are predicted error maps, from which a set of t **exemplar masks** $\hat{\mathcal{M}}_{e1}^{\ k}, \ldots, \hat{\mathcal{M}}_{et}^{\ k}$ can be determined for **each** object by comparing the size of predicted erroneous regions.

By considering multiple frames and their associated flow-based mask proposals, the selection procedure is able to employ two potential cues, as illustrated in Figure 3.2: (i) the temporal consistency between neighbouring masks – objects

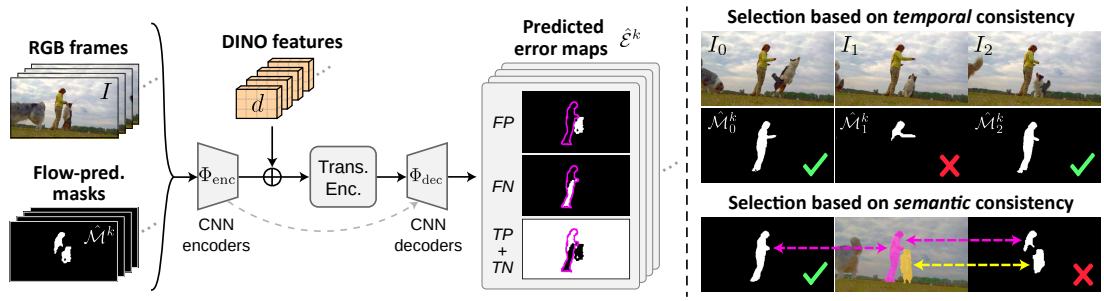


Figure 3.2: **Appearance-based mask selection. Left:** The mask selector takes in RGB frames, DINO features, and flow-predicted masks, generating three-channel error maps for each object (in this case the standing person). The error maps are used to select the exemplar masks for each object. The false positive (FP) channel highlights the over-segmented areas (the dog), while the false negative (FN) channel predicts the missing masks (the lower body of the person). The remaining channel encompasses a combination of true positive and true negative (TP + TN) regions, where the flow-predicted masks align accurately with the groundtruth, as indicated by white regions. The groundtruth annotation boundaries (purple contours) are provided for illustration purposes only. **Right:** Two potential cues underlying the mask selection process: (i) temporal consistency of mask shapes across consecutive frames, and and (ii) the consistency of semantic content – the mask (including all its parts) should only correspond to one semantic class.

undergo smooth and continuous changes in position and shape, and the selector can filter out the frames with empty or incomplete masks due to stationary objects (*i.e.* no flow) or abrupt occlusions; (ii) the consistency of semantic content within each RGB frame – as a high-quality object mask normally corresponds to only one semantic class and this does not switch throughout the video.

Architecture. Figure 3.2 illustrates a simple architecture for the mask selector that considers a joint input of RGB frames and flow-predicted masks at all time steps. To incorporate semantic priors, visual features from the frozen DINO ViT [Caron et al., 2021] are concatenated channel-wise with the CNN-encoded features in the model bottleneck. These DINO features have been shown to be effective for encoding frame-to-frame correspondence [Siméoni et al., 2021; Melas-Kyriazi et al., 2022; Wang et al., 2022; Shin et al., 2022], and thus enable the transformer to use the two potential cues for the selection. Full-resolution error maps are obtained from a UNet-like upsampling process, where the original RGB frames help by providing sharp boundaries.

Training objective. In contrast to prior approaches [Griffin and Corso, 2019; Yin et al., 2021] that directly compute confidence scores, we task the selector to predict an error map, *i.e.* the difference map between the input flow-predicted mask and the groundtruth segmentation, as an intermediate result. Specifically, for each frame i, the prediction $\hat{\mathcal{E}}_i \in \mathbb{R}^{H\times W\times 3}$ consists of three channels corresponding to the false positive (FP), false negative (FN) and a combination of true positive and true negative (TP + TN) regions. We train the mask selector by applying a binary cross-entropy loss on each of the predicted regions:

$$\mathcal{L}_{\text{selector}} = \frac{1}{T}\sum_{i=1}^{T}\Big(\mathcal{L}_{\text{bce}}(\hat{\mathcal{E}}_i, \mathcal{E}_i)\Big) \tag{3.4}$$

The groundtruth error map, $\mathcal{E}_i$, is derived from a synthetic video dataset (as will be detailed in Section 3.3.4), which serves as the training data for the model. By comparing the size of erroneous regions (*i.e.* FP and FN regions) across the sequence, we can compute a score that indicates the mask quality for each frame, and subsequently keep t exemplar masks with high scores for each object.

3.3.3 Appearance-based mask correction

Given the high-quality exemplars from the selector, the object-centric corrector aims to refine the flow-predicted object masks across T contiguous **target frames** of the sequence. This correction process essentially uses strong framewise correlations to facilitate generalisation of the sparse exemplar mask information to other (target) frames. Additionally, it recovers the fine-grained details by leveraging the input appearance information and the sharp boundaries in RGB frames and flow-predicted masks. The correction can be written as:

$$\{\tilde{\mathcal{M}}_1^k, \ldots, \tilde{\mathcal{M}}_T^k\}_{k=1}^K = \Phi_{\text{corrector}}(\mathcal{V}_{\text{RGB}}; \{\hat{\mathcal{M}}_1^k, \ldots, \hat{\mathcal{M}}_T^k; \hat{\mathcal{M}}_{e1}^k, \ldots, \hat{\mathcal{M}}_{et}^k\}_{k=1}^K) \tag{3.5}$$

where $\tilde{\mathcal{M}}_i^k$ refers to the refined mask for the k^{th} object at frame i.

In overview, a variant of a transformer decoder is used, as illustrated in Figure 3.3. A query vector is provided for each object, initialised from the exemplar masks for that object, and subsequently the object query vectors are jointly processed, and attend to the target and exemplar mask frames to establish temporal correlations

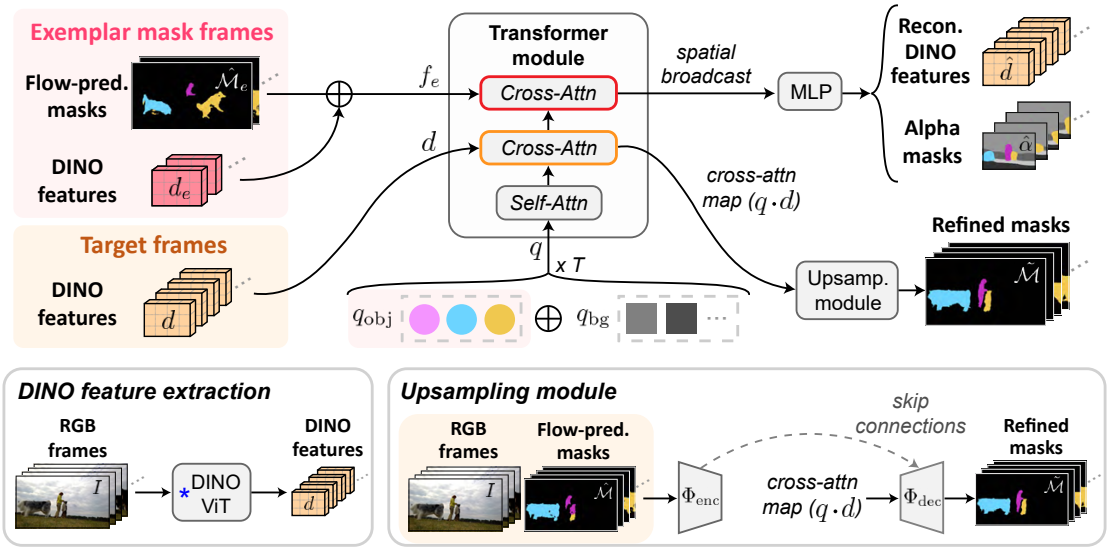


Figure 3.3: **Appearance-based mask corrector.** The object-centric architecture first utilises the exemplar mask information to initialise independent object queries. The transformer decoder module then refines the query vectors with two cross-attention layers using dense target frame features for one, and exemplar information for the other. There are two major outputs: (i) the refined queries are spatially broadcast and projected to recover DINO features in target frames, and (ii) the cross-attention maps between queries and target frame DINO features are extracted, followed by a CNN-based upsampling module to yield final refined masks. Note, "Exemplar mask frames" refer to frames from which the exemplar masks are selected. For illustration, we present exemplar masks for different objects in the same frame. However, in practice, different objects may have distinct sets of exemplar mask frames.

in the visual stream. Again, DINO features are used to enable semantic grouping and temporal correspondence. In the following, we give more details for each element of this architecture.

Query initialisation. Foreground object-wise query vectors ($q_{\text{obj}} \in \mathbb{R}^{K \times D}$) are initialised from the visual features for the set of exemplar mask frames, by weighted pooling the dense feature maps with the exemplar masks along both spatial and temporal dimensions. An additional set of K_{bg} learnable embedding vectors ($q_{\text{bg}} \in \mathbb{R}^{K_{\text{bg}} \times D}$) models the background regions. As a result, there are a total of $K + K_{\text{bg}}$ query vectors to represent the entire video scene. To account for the temporal dimension in videos, we broadcast these query vectors to T target frames, leading to a total of $(K + K_{\text{bg}}) \times T$ initialised query vectors, *i.e.* $q \in \mathbb{R}^{(K+K_{\text{bg}}) \times T \times D}$.

Transformer module. Once initialised, the queries q undergo iterative refinements through a series of transformer-like layers, with two differences to the stan-

dard transformer decoder: (i) we insert an additional multi-head cross-attention layer; (ii) we apply a Slot-Attention-like mechanism [Locatello et al., 2020] within each cross-attention module, encouraging competition between foreground objects and background regions.

In the first cross-attention layer, the query vectors attend to the visual features of all adjacent **target** frames, whereas in the second cross-attention layer, the key and value entries are generated by combining **exemplar** masks with the visual features of corresponding frames. These layers enable the incorporation of short-term local information and global exemplar information, respectively.

Refined mask outputs. Following a similar idea to [Cheng et al., 2021; Xie et al., 2022], we extract the cross-attention maps between query vectors and visual features of target frames from the last layer of the transformer module. Notably, we only retain the attention maps that are associated with K foreground object queries. These filtered attention maps are then upsampled by a CNN decoder to yield refined masks $\{\tilde{\mathcal{M}}_1^k, \ldots, \tilde{\mathcal{M}}_T^k\}_{k=1}^K$. During the upsampling process, information of flow-predicted masks and object appearances in target frames are passed through a lightweight CNN encoder, acting as skip layers for fine-grained feature compensation.

Mask prediction loss. At training time, we apply both binary cross-entropy loss and boundary loss (adopted from [Xie et al., 2022]) on the resultant refined masks

$$L_{\text{mask}} = \frac{1}{KT} \sum_{i=1}^{T} \sum_{k=1}^{K} \left(\lambda_{\text{bce}} L_{\text{bce}}(\tilde{\mathcal{M}}_i^k, \mathcal{M}_i^k) + \lambda_{\text{bound}} L_{\text{bound}}(\tilde{\mathcal{M}}_i^k, \mathcal{M}_i^k) \right) \tag{3.6}$$

with $\lambda_{\text{bce}}, \lambda_{\text{bound}}$ denoting weight factors. The groundtruth segmentation masks $\{\mathcal{M}_i^k\}$ are provided by synthetic video sequences, as will be detailed in Section 3.3.4.

Feature reconstruction loss. In addition to direct supervision, we incorporate feature reconstruction as an auxiliary task. Specifically, the output query vectors from the transformer module are spatially broadcast and projected by MLP layers to reconstruct the DINO feature maps d of input frames. This reconstruction process is inspired from [Yang et al., 2021a; Seitzer et al., 2023], where the feature

predictions by each query $\hat{d}^k$ are linearly combined, weighted by the predicted alpha channels $\hat{\alpha}^k$, *i.e.* $\hat{d} = \sum_k \hat{\alpha}^k \hat{d}^k$.

The overall loss for this feature reconstruction process is averaged over all target frames:

$$L_{\text{feat-recon}} = \frac{1}{T}\sum_{i=1}^{T}\left(\lambda_{\text{recon}}\text{MSE}(\hat{d}_i, d_i) + \frac{1}{K}\sum_{k=1}^{K}\lambda_{\text{alpha}}L_{\text{bce}}(\hat{\alpha}_i^k, \alpha_i^k)\right) \tag{3.7}$$

where the first term (weighted by λ_{recon}) denotes an unsupervised feature reconstruction loss, and the second term (weighted by λ_{alpha}) corresponds to a supervised alpha channel prediction loss. This supervision signal is only applied to the alpha masks generated from foreground object queries, with the groundtruth $\{\alpha_i^k\}$ obtained by downsampling the groundtruth segmentation masks $\{\mathcal{M}_i^k\}$.

3.3.4 Sim2Real training procedure

Training proceeds in two steps: first, the selector and corrector are pre-trained with synthetic videos; and second, a self-supervised adaptation process is used to bridge the domain gap to real-world videos. Both steps are described below. Notably, this entire training process does not require any manual annotations.

Pre-training on synthetic videos.

We adopt the simple simulation engine proposed by [Xie et al., 2022] to generate synthetic training video sequences with RGB frames, optical flows and groundtruth annotations.

Synthetic training of mask selector. Given one synthetic video sequence, we feed its optical flows into the OCLR-flow multi-object motion segmentation model, to yield flow-predicted masks as the selector inputs. The groundtruth error maps for training (as introduced in Equation 3.4) can be obtained by comparing the flow-predicted masks with groundtruth segmentation maps.

Synthetic training of mask corrector. By combining loss functions introduced in the refined mask prediction (Equation 3.6) and feature reconstruction

(Equation 3.7) processes, the overall loss for training the mask corrector is

$$L_{\text{corrector}} = \lambda_{\text{mask}} L_{\text{mask}} + \lambda_{\text{feat-recon}} L_{\text{feat-recon}} \tag{3.8}$$

where λ_{mask} and $\lambda_{\text{feat-recon}}$ are corresponding weight factors, and the supervision is provided by the groundtruth annotations in the synthetic sequences.

Both the selector and corrector models, trained on synthetic data, are able to generalise to real-world videos, as will be demonstrated in our ablation analysis.

Self-supervised adaptation to real-world videos.

The Sim2Real domain gap is mitigated by self-supervised training on target real-world datasets. Specifically, for each real-world video dataset, we first employ our synthetically-trained selection-correction model to generate segmentation masks starting from the OCLR-flow proposals. The resultant masks are then treated as pseudo-labeling to fine-tune the mask corrector and the OCLR-flow segmentation model. We provide more details on each fine-tuning process below, and verify the effectiveness of each stage via ablation analysis in in the Appendix.

Adapting the mask correction model. The overall loss for fine-tuning remains the same as for synthetic training (Equation 3.8). However, for adaptation, the refined masks generated by the synthetically-trained corrector (*i.e.* $\{\tilde{\mathcal{M}}_i^k\}$) are treated as pseudo-GT, and become the target masks in the loss. Note that, we only apply pseudo-labeling at the frames where exemplar masks present due to their high quality. Apart from the self-supervised training at the dataset level, we also employ a test-time per-sequence adaptation to further boost the performance. A detailed description is given in the Appendix.

Adapting the flow segmentation model. Similarly, the OCLR-flow model (which is also originally trained only on synthetic data) is adapted to real-world videos by fine-tuning with the pseudo-GT provided by the refined segmentation masks $\{\tilde{\mathcal{M}}_i^k\}$. This adapts the model to complex real-world flow inputs, consequently reducing Sim2Real domain gap.

The updated flow-predicted masks can then form the basis for a further appearance-based selector-corrector refinement. In principle, it would be possible to repeat

this refinement-adaptation process. However, in practice, we find that one round of adaptation is sufficient and there is little gain in iterating further.

3.4 Experiments

3.4.1 Datasets and metrics

Training dataset. To pre-train our models, we use a synthetic dataset generated by the simple pipeline introduced in [Xie et al., 2022], where motions of multiple object sprites (*i.e.* shapes with random textures) are simulated on a common moving background across time. Each sequence contains 1 to 3 moving sprites, with information such as RGB frames, optical flows, groundtruth modal and amodal annotations provided.

Multi-object benchmark. We adopt the DAVIS2017-motion dataset (short form "DAVIS17-m") that is proposed in [Xie et al., 2022] with jointly moving objects re-annotated. We further formulate a YouTubeVOS2018-motion dataset (short form "YTVOS18-m"), which is a subset of YTVOS18 [Xu et al., 2018], containing 120 sequences with one or more dominant moving objects. Visualisations of multi-object sequences can be found in Figure 3.4. Additional information regarding this new benchmark is provided in the Appendix.

Single-object benchmark. We report performance on popular single-object video segmentation benchmarks including DAVIS2016 [Perazzi et al., 2016], SegTrackv2 [Li et al., 2013] and FBMS-59 [Ochs et al., 2014]. Though some sequences in SegTrackv2 and FBMS-59 consist of multiple objects, we follow the common practice [Jain et al., 2017; Yang et al., 2019c] and group foreground objects as a whole for later evaluation.

Evaluation metrics. We benchmark our single object video segmentation performance using region similarity ($\mathcal{J}$). For multi-object video segmentation, we additionally report the contour similarity ($\mathcal{F}$) [Martin et al., 2004].

Table 3.1: **Quantitative comparison on multi-object video segmentation benchmarks.** "H.A." indicates that human annotations are involved during training. † denotes that we reproduce the results on the benchmark datasets using the official code released by the original authors. In the semi-supervised setting, the first-frame ground-truth annotation is provided.

	Model settings					YTVOS18-m			DAVIS17-m		
Model	VOS	H.A.	RGB	Flow	Input Res.	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
†MG [Yang et al., 2021a]	un-sup.	✗	✗	✓	128 × 224	33.3	37.0	29.6	35.8	38.4	33.2
OCLR-flow [Xie et al., 2022]	un-sup.	✗	✗	✓	128 × 224	45.3	46.5	44.1	55.1	54.5	55.7
Safadoust et al. [Safadoust and Güney, 2023]	un-sup.	✗	✓	✗	128 × 224	–	–	–	59.2	59.3	59.2
OCLR-TTA [Xie et al., 2022]	un-sup.	✗	✓	✓	480 × 854	50.6	52.7	48.6	64.4	65.2	63.6
†VideoCutLER [Wang et al., 2023b]	un-sup.	✗	✓	✗	480 × 854	57.0	59.0	55.1	57.3	57.4	57.2
Ours	un-sup.	✗	✓	✓	128 × 224	**65.2**	**67.1**	**63.2**	**66.2**	**67.0**	**65.4**
OCLR-flow [Xie et al., 2022] + SAM [Kirillov et al., 2023]	un-sup.	✓	✓	✓	576 × 1024	58.5	57.0	60.0	64.2	62.0	66.4
Ours + SAM [Kirillov et al., 2023]	un-sup.	✓	✓	✓	576 × 1024	**70.6**	**71.1**	**70.2**	**71.5**	**70.9**	**72.1**
†CorrFlow [Lai and Xie, 2019]	semi-sup.	✗	✓	✗	480 × 854	55.2	60.0	50.4	54.0	54.2	53.7
†UVC [Li et al., 2019a]	semi-sup.	✗	✓	✗	480 × 854	66.9	70.4	63.4	65.5	66.2	64.7
†CRW [Jabri et al., 2020]	semi-sup.	✗	✓	✗	480 × 854	72.9	75.5	70.4	73.4	72.9	74.1
†DINO [Caron et al., 2021]	semi-sup.	✗	✓	✗	480 × 854	**75.4**	**76.6**	**74.3**	**78.7**	**77.7**	**79.6**

3.4.2 Implementation details

For pre-processing, we apply RAFT [Teed and Deng, 2020] to estimate optical flow with frame gaps ±1 (except for FBMS and YTVOS18-m with frame gaps ±3 owing to relatively slow video motions). All optical flows and RGB frame inputs are resized to 128 × 224. The motion segmentation model proposes $K = 3$ objects for each sequence, from which $t = 10$ exemplar masks are selected for each object. Also, we use self-supervised pre-trained DINO transformers [Caron et al., 2021], ViT-S/16 for mask selector, ViT-S/8 for corrector, resulting in DINO feature resolutions 8 × 14 and 16 × 28, respectively. As a final post-processing step, we apply Conditional Random Field (CRF) segmentation to further refine the output masks. We refer the reader to the Appendix.

3.4.3 Multiple object video segmentation

Comparison to unsupervised models. Table 3.1 compares the performance of our motion segmentation model against several baselines on unsupervised video object segmentation benchmarks. Interestingly, most prior baseline methods exhibit degraded performance on the YTVOS18-m dataset compared to the DAVIS17-m dataset. This is understandable, as the former dataset involves more complicated motion patterns, posing difficulties for motion-based segmentation. However, our method effectively handles this problem and demonstrates superior performance on both benchmark datasets.



Figure 3.4: **Qualitative comparison on multi-object video segmentation benchmarks**, including YTVOS18-m (left) and DAVIS17-m (right).

Our findings are further supported by comprehensive ablation studies provided in the Appendix.

Comparison to the segment anything model (SAM) [Kirillov et al., 2023]. Recent studies highlight SAM as a robust general segmentor with diverse prompt options. To assess its efficacy for refining flow-based segmentation, we prompt SAM with masks predicted by the OCLR-flow model in a per-frame manner. As shown in Table 3.1, the resultant refinements (OCLR-flow + SAM) are not as effective as the refinements provided by our selection-correction model (Ours), owing to the limited utilisation of temporal consistency in the former approach. On the other hand, prompting SAM with masks predicted by our model does lead to improvements (Ours + SAM). This demonstrates that the SAM-based refinement is complementary to our method built on the persistence of appearance streams. Visualisations of SAM-based refinements are given in the Appendix.

Comparison to semi-supervised models. Table 3.1 also includes a list of existing semi-supervised methods that were re-run on motion segmentation benchmarks for reference. Interestingly, the issue of stationary frames in YTVOS18-m actually benefits semi-supervised tasks, as objects that move infrequently are easier to track using first-frame ground truth. Despite this advantage, our unsupervised-VOS method still achieves competitive results compared to some semi-supervised-

Table 3.2: **Quantitative comparison on single object video segmentation benchmarks.** "H.A." indicates that human annotations are involved during training. ‡ corresponds to methods that rely on human-label-free supervision on synthetic data.

Model	Model settings				DAVIS16	STv2	FBMS59
	H.A.	RGB	Flow	Input Res.	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$
MG [Yang et al., 2021a]	✗	✗	✓	128 × 224	68.3	58.6	53.1
‡OCLR-flow [Xie et al., 2022]	✗	✗	✓	128 × 224	72.1	67.6	70.0
DS [Ye et al., 2022]	✗	✓	✓	240 × 426	79.1	72.1	71.8
DystaB [Yang et al., 2021c]	✗	✓	✓	192 × 384	80.0	74.2	73.2
GWM [Choudhury et al., 2022]	✗	✓	✗	128 × 224	80.7	78.9	78.4
‡OCLR-TTA [Xie et al., 2022]	✗	✓	✓	480 × 854	80.9	72.3	72.7
LOCATE [Singh et al., 2023]	✗	✓	✗	480 × 848	80.9	**79.9**	68.8
‡**Ours**	✗	✓	✓	128 × 224	**81.1**	76.6	**81.9**
OCLR-flow [Xie et al., 2022] + SAM [Kirillov et al., 2023]	✓	✓	✓	576 × 1024	80.6	71.5	79.2
MATNet [Zhou et al., 2020]	✓	✓	✓	473 × 473	82.4	50.4	76.1
DystaB [Yang et al., 2021c]	✓	✓	✓	192 × 384	82.8	74.2	75.8
Ours + SAM [Kirillov et al., 2023]	✓	✓	✓	576 × 1024	86.6	**81.3**	**85.7**
DPA [Cho et al., 2024]	✓	✓	✓	512 × 512	**87.1**	–	81.0

VOS baselines.

Qualitative results. Figure 3.4 presents visualisations of multi-object sequences with complex flow signals. It is evident that the method relying solely on flow inputs (*i.e.* OCLR-flow) sometimes fails to extract object shapes from static flow fields and articulated motion signals. OCLR-TTA addresses this issue by using RGB-based dense mask propagations, but still struggles with completing the shape of objects, such as the large white tiger in the first column. In contrast, our method accurately recovers object segmentation even under extreme motion scenarios, thanks to its object-centric representation and reconstruction-based scene clustering.

3.4.4 Single object video segmentation

We also summarise our performance on various single-object segmentation benchmarks. These comparisons with other self-supervised approaches are considered reasonable since both setups rely on unsupervised learning schemes on targeted data distributions, and no manual annotations are used during the training process. While our method is not specialised for the task of single object segmentation, as demonstrated in Table 3.2, it still achieves the top performance on two out of three datasets among methods that do not use human annotations. It does less well on the SegTrack dataset since the low-quality sequences (that have watermarks and video compression artefacts) result in noisy flow estimates that compromise our



Figure 3.5: **Qualitative comparison on single object video segmentation benchmarks**, including DAVIS16 (left), SegTrackv2 (middle), and FBMS-59 (right).

performance.

Furthermore, applying per-frame SAM to refine mask predictions, gives noticeable improvements, and leads to the top or matching performance among supervised methods. Figure 3.5 provides corresponding visualisations on single-object benchmarks. Similar to the multi-object case, our method exhibits outstanding competence in handling challenging motion situations.

3.5 Discussion

In this paper, we introduce an appearance-based refinement method to address inaccuracies in flow-predicted masks. Our approach involves synthetic pre-training, followed by self-supervised adaptation to real-world videos, without relying on any human annotations. By formulating a new motion segmentation benchmark, we have demonstrated the efficacy of our method in handling complicated motion patterns such as moving objects that are motionless for a time, environment motions, multi-object interactions, *etc.*

Instead of pursuing an end-to-end design, we have explored the potential of employing multiple stages: flow-based mask proposals, appearance-based mask selection, and appearance-based mask correction, with each stage utilising a lightweight interpretable model. More importantly, this approach allows us to efficiently harness the advantages of both flow and appearance modalities: the object discovery and accurate boundary information from optical flow; and the temporal consistency observed in appearance streams.

Nevertheless, some limitations have also been revealed under extremely complex motion scenarios: *First*, when a significant majority of problematic flow-predicted masks are present (*e.g.* due to a large proportion of stationary frames in the sequence), selecting meaningful exemplar masks becomes challenging, consequently affecting the correction process. This problem could be potentially mitigated by incorporating additional appearance priors into the flow-based proposal stage. *Second*, the adopted frozen DINO ViT mainly encodes semantic-level information, which could result in difficulties in discerning entangled objects with similar appearances at the instance level. This highlights a possible research direction, aiming to develop instance-level appearance representations by leveraging flow-predicted information in complex multi-object scenes.

Acknowledgements

This research is supported by the UK EPSRC Programme Grant Visual AI (EP/T028572/1), a Royal Society Research Professorship RP\R1\191132, and a Clarendon Scholarship.

Appendices

Appendices for this chapter can be found in the online version of the paper.[1]

Statement of authorship

A statement of authorship for this paper is provided in Appendix A.

[1] https://www.robots.ox.ac.uk/~vgg/research/appear-refine/

Chapter 4

Moving Object Segmentation: All You Need Is SAM (and Flow)

This paper was published at the Asian Conference on Computer Vision (ACCV), 2024.

Abstract

The objective of this paper is motion segmentation – discovering and segmenting the moving objects in a video. This is a much studied area with numerous careful, and sometimes complex, approaches and training schemes including: self-supervised learning, learning from synthetic datasets, object-centric representations, amodal representations, and many more. Our interest in this paper is to determine if the Segment Anything model (SAM) can contribute to this task.

We investigate two models for combining SAM with optical flow that harness the segmentation power of SAM with the ability of flow to discover and group moving objects. In the first model, we adapt SAM to take optical flow, rather than RGB, as an input. In the second, SAM takes RGB as an input, and flow is used as a segmentation prompt. These surprisingly simple methods, without any further modifications, outperform all previous approaches by a considerable margin in both single and multi-object benchmarks. We also extend these frame-level segmentations to sequence-level segmentations that maintain object identity. Again, this simple model achieves outstanding performance across multiple moving object segmentation benchmarks.

4.1 Introduction

Recent research in image segmentation has been transformative, with the Segment Anything Model (SAM) [Kirillov et al., 2023] emerging as a significant breakthrough. Leveraging large-scale datasets and scalable self-labelling, SAM enables flexible image-level segmentation across many scenarios [Ma et al., 2024; Wu et al., 2023b; Ren et al., 2024; Tang et al., 2023; Zhang et al., 2023d; Chen et al., 2023c], facilitated by user prompts such as boxes, texts, and points. In videos, optical flow has played an important and successful role for *moving object segmentation* – in that it can (i) discover moving objects, (ii) provide crisp boundaries for segmentation, and (iii) group parts of objects together if they move together. It has formed the basis for numerous methods of moving object discovery by self-supervised learning [Yang et al., 2021a; Meunier and Bouthemy, 2023b; Xie et al., 2022; Lamdouar et al., 2021; Meunier et al., 2023; Choudhury et al., 2022; Safadoust and Güney, 2023]. However, it faces segmentation challenges if objects

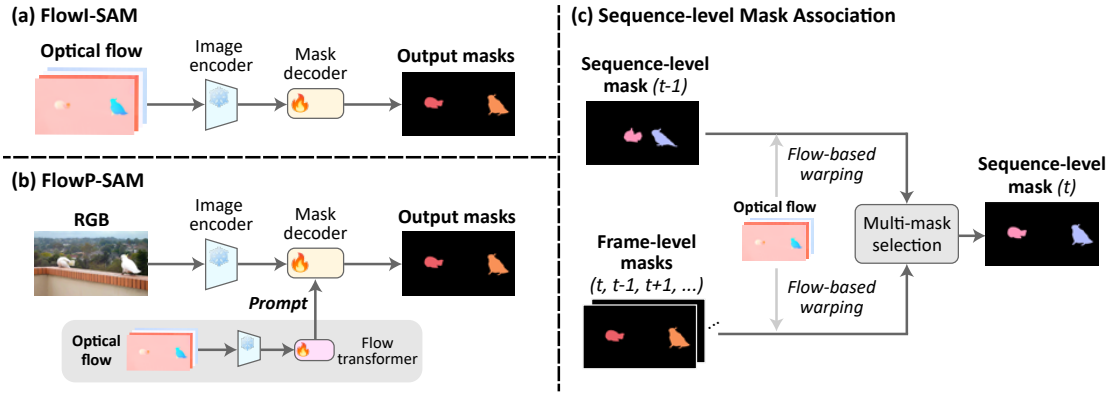


Figure 4.1: **Adapting SAM for Video Object Segmentation by incorporating flow. (a) Flow-as-Input**: `FlowI-SAM` takes in optical flow *only* and predicts frame-level segmentation masks. **(b) Flow-as-Prompt**: `FlowP-SAM` takes in RGB and applies flow information as a prompt for frame-level segmentation. **(c) Sequence-level mask association**: as a post-processing step, the multi-mask selection module autoregressively transforms *frame-level* mask outputs from `FlowI-SAM` and/or `FlowP-SAM` and produces *sequence-level* masks in which object identities are consistent throughout the sequence.

are momentarily motionless, and in distinguishing foreground objects from background 'noise'. This naturally raises the question: "How can we leverage the power of SAM with flow for moving object segmentation in videos?".

To this end, we explore two simple variants to effectively tailor SAM for motion segmentation. *First*, we introduce `FlowI-SAM` (Figure 4.1a), an adaption of the original SAM that directly processes optical flow as a three-channel *input* image for segmentation, where points on a uniform grid are used as prompts. This approach leverages the ability of SAM to accurately segment moving objects against the static background, by exploiting the distinct textures and clear boundaries present in optical flow fields. However, it has less success in scenes where the optical flow arises from multiple interacting objects as the flow only contains limited information for separating them. *Second*, building on the strong ability of SAM on RGB image segmentation, we propose `FlowP-SAM` (Figure 4.1b) where the input is an RGB frame, and flow guides SAM for moving object segmentation as *prompts*, produced by a trainable prompt generator. This method effectively leverages the ability of SAM on RGB image segmentation, with flow information acting as a selector of moving objects/regions within a frame. Additionally, we extend these methods from frame-level to *sequence-level* video segmentation (Figure 4.1c) so that object identities are consistent throughout the sequence. We do this by introducing a matching module that auto-regressively chooses whether to select a new

object or propagate the old one based on temporal consistency.

In summary, this paper introduces and explores two models to leverage SAM for moving object segmentation in videos, enabling the principal moving objects to be discriminated from background motions. Our contributions are threefold:

- The `FlowI-SAM` model, which utilises optical flow as a three-channel input image for precise *per-frame* segmentation and moving object identification.
- The `FlowP-SAM` model, a novel combination of dual-stream (RGB and flow) data, that employs optical flow to generate prompts, guiding SAM to identify and localise the moving objects in RGB images.
- New state-of-the-art unsupervised video object segmentation performance by a large margin on moving object segmentation benchmarks, including DAVIS16, DAVIS17-m, YTVOS18-m, and MoCA.

4.2 Related work

Video Object Segmentation (VOS) is an extensively studied task in computer vision. The objective is to segment the primary object(s) in a video sequence. Numerous benchmarks are developed for evaluating VOS performance, catering to both single-object [Perazzi et al., 2016; Li et al., 2013; Ochs and Brox, 2011; Lamdouar et al., 2020] and multi-object [Pont-Tuset et al., 2017; Xu et al., 2018] scenarios. Two major protocols are widely explored in VOS research, namely unsupervised [Yang et al., 2019d; Yang et al., 2021a; Lu et al., 2019; Ventura et al., 2019; Cho et al., 2019; Li et al., 2018; Luiten et al., 2020; Lin et al., 2021] and semi-supervised VOS [Vondrick et al., 2018; Lai and Xie, 2019; Lai et al., 2020; Miao et al., 2022; Wang et al., 2019b; Jabri et al., 2020; Caron et al., 2021; Oh et al., 2019; Yang and Yang, 2022; Cheng and Schwing, 2022; Pan et al., 2022]. Notably, the term "unsupervised" exclusively indicates that no groundtruth annotation is used *during inference time* (*i.e.* no inference-time supervision). In contrast, the semi-supervised VOS employs first-frame groundtruth annotations to initiate the object tracking and mask propagation in subsequent frames. This paper focuses on unsupervised VOS and utilises motion as a crucial cue for object discovery.

Motion Segmentation focuses on discovering objects through their movement and generating corresponding segmentation masks. Existing benchmarks for motion segmentation largely overlap with those used for VOS evaluation, especially in the single-object case. For multi-object motion segmentation, datasets [Xie et al., 2022; Xie et al., 2024b] have been specifically curated from VOS benchmarks to exclusively focus on sequences with dominant locomotion. There are two major setups in the motion segmentation literature: one that relies on motion information *only* to distinguish moving elements from the background through spatial clustering [Yang et al., 2021a; Meunier et al., 2023; Meunier and Bouthemy, 2023b] or explicit supervision [Lamdouar et al., 2021; Xie et al., 2022]; the other [Bideau and Learned-Miller, 2016b; Mahendran et al., 2018; Yang et al., 2021c; Choudhury et al., 2022; Xie et al., 2024b; Ponimatkin et al., 2023] that enhances motion-based object discovery by incorporating appearance information. We term these two approaches "flow-only" and "RGB-based" segmentation, respectively, and explore both setups in this work.

Segment anything model (SAM) [Kirillov et al., 2023] has demonstrated impressive ability on image segmentation across various scenarios. It was trained on the SA-1B datasets with over one billion self-labelled masks and 11 million images. Such large-scale training renders it a strong zero-shot generalisability to unseen domains. Many works adapt the SAM model to perform different tasks, such as tracking [Cheng et al., 2023b], change detection [Zheng et al., 2024b], and 3D segmentation [Cen et al., 2023]. Some other works extend SAM towards more efficient models [Zhang et al., 2023a; Zhao et al., 2023a; Xiong et al., 2023], and more domains [Ma et al., 2024; Wu et al., 2023b; Ren et al., 2024; Chen et al., 2023c]. However, most studies follow the default prompt options in SAM (*i.e.* points, boxes, and masks). Recent works [Zou et al., 2023; Sun et al., 2024] have shown that more versatile prompts, including scribbles and visual references, can lead to improvements. In this paper, we explore a novel route that prompts SAM with optical flow and demonstrate its effectiveness for moving object segmentation.

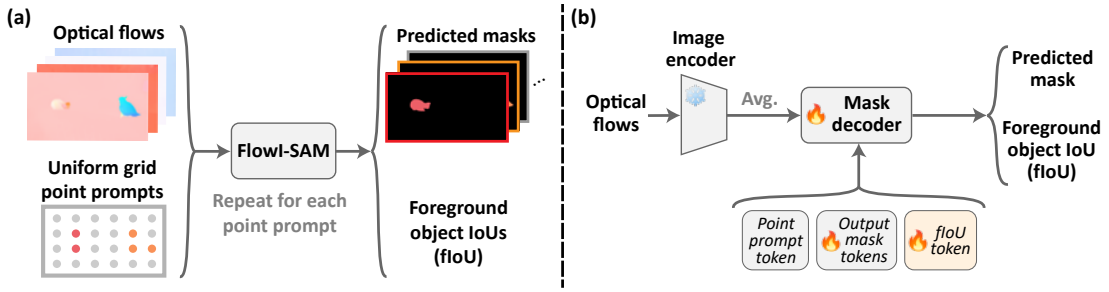


Figure 4.2: **Overview of `FlowI-SAM`. (a)** Inference pipeline of `FlowI-SAM`. **(b)** Architecture of `FlowI-SAM` with trainable parameters labelled. The point prompt token is generated by a frozen prompt encoder.

4.3 SAM preliminaries

The Segment Anything Model (SAM) is engineered for high-precision image segmentation, accommodating both user-specified prompts and a fully autonomous operation mode. When guided by user input, SAM accepts various forms of prompts including points, boxes, masks, or textual descriptions to accurately delineate the segmentation targets. Alternatively, in its automatic mode, SAM uses points on a uniform grid as prompts, to propose all plausible segmentation masks that capture objects and their hierarchical subdivisions—objects, parts, and subparts. In this case, the inference is repeated for each prompt of the grid, generating masks for each prompt in turn, and the final mask selection is guided by the predicted mask IoU scores.

Architecturally, SAM is underpinned by three foundational components: (i) Image encoder extracts strong image features via a heavy Vision Transformer (ViT) backbone, which is pre-trained by the Masked Auto-Encoder (MAE) approach; (ii) The prompt encoder converts the input prompts into positional information which helps with locating the segmentation target; (iii) Mask decoder features a lightweight two-way transformer that takes in a combination of encoded prompt tokens, learnable mask tokens, and an IoU prediction token as input queries. These queries iteratively interact with the dense spatial features from image encoder, leading to the final mask predictions and IoU estimations. In the next sections, we describe two distinct, yet simple variants to effectively tailor SAM for motion segmentation.

4.4 Frame-level segmentation I: Flow as input

In this section, we focus on discovering moving objects from individual frames by exploiting motion information *only*, to yield corresponding segmentation masks. Formally, given the optical flow input $F_t \in \mathbb{R}^{H\times W\times 3}$ at frame t, we aim to predict a segmentation mask $M_t^i \in \mathbb{R}^{H\times W}$ together with a foreground object IoU (fIoU) score $s^i_{\text{fIoU},t} \in \mathbb{R}^1$ for each object i,

$$\{M_t^i, s^i_{\text{fIoU},t}\}_{i=0}^N = \Phi_{\texttt{FlowI-SAM}}(F_t) \tag{4.1}$$

To adapt SAM for this new task, we formulate `FlowI-SAM` ($\Phi_{\texttt{FlowI-SAM}}$) by finetuning it on optical **Flow I**nputs, and re-purpose the original IoU prediction head to instead predict the fIoU, as illustrated in Figure 4.2b. By definition, fIoU is a scalar that measures the "objectness": fIoU is 0 if the mask belongs to the background, and equal to the IoU between predicted and GT object masks for foreground moving object masks. A high fIoU indicates the predicted mask corresponds to the entire object, while a low fIoU might suggest the mask is erroneous or only captures a small part of the object.

Flow inputs with multiple frame gaps. To mitigate the effect of noisy optical flow, *i.e.* complicated flow fields due to stationary parts, articulated motion, and object interactions, *etc.*, we consider multiple flow inputs $\{F_{t,g}\}$ with different frame gaps (*e.g.* $g \in \{(1,\text{-}1), (2,\text{-}2)\}$) for both training and evaluation stages. These multi-gap flow inputs are **independently** processed by the image encoder to obtain dense spatial features $\{d_{t,g}\}$ at a lower resolution $h \times w$, which are then combined by averaging the spatial feature maps across different flow gaps, *i.e.* $d_t = \text{Average}_g(\{d_{t,g}\}) \in \mathbb{R}^{h\times w\times d}$. These averaged spatial features are then treated as keys and values in the mask decoder.

`FlowI-SAM` **inference.** To discover all moving objects from flow input, the `FlowI-SAM` model is prompted by points on a uniform grid. Each point prompt outputs a pair of mask and objectness score predictions. This mechanism is the same as in the original SAM formulation, and is illustrated in Figure 4.2a. The final segmentation is selected using Non-Maximum Suppression (NMS) based on the predicted fIoU and overlap ratio.

`FlowI-SAM` **training.** To adapt the pre-trained SAM model for optical flow inputs, we finetune the lightweight mask decoder, while the image encoder and the prompt encoder remain frozen. The overall loss is formulated as:

$$\mathcal{L}_{\texttt{FlowI-SAM}} = \frac{1}{NT}\sum_{i,t}^{N,T}\left(\mathcal{L}_{\text{BCE}}(M_t^i, \hat{M}_t^i) + \lambda_f \|s_{\text{fIoU},t}^i - \hat{s}_{\text{fIoU},t}^i\|^2\right) \tag{4.2}$$

where $\hat{M}_t^i$ and $\hat{s}_{\text{fIoU},t}^i$ denote the groundtruth segmentation masks and fIoU, and λ_f is a scale factor.

4.5 Frame-level segmentation II: Flow as prompt

In this section, we adapt SAM for video object segmentation by processing RGB frames, with optical flow as a prompt. We term this frame-level segmentation architecture `FlowP-SAM` for **Flow** as **P**rompt SAM. As shown in Figure 4.3b, `FlowP-SAM` encompasses two major modules, namely the flow prompt generator and the segmentation module. The flow prompt generator takes optical flow as inputs, and produces flow prompts that can be used as supplemental queries to infer frame-level segmentation masks M_t^i from RGB inputs I_t. Formally,

$$\{M_t^i,\ s_{\text{fIoU},t}^i,\ s_{\text{MOS},t}^i\}_{i=0}^N = \Phi_{\texttt{FlowP-SAM}}(F_t, I_t) \tag{4.3}$$

where $s_{\text{MOS},t}^i$ indicates the moving object score (MOS) predicted by the flow prompt generator, while $s_{\text{fIoU},t}^i$ denotes the foreground object IoU (fIoU) estimated by the segmentation module. Specifically, MOS measures whether each input point prompt (therefore the resultant mask) is within a moving object region based on observing flow fields. Groundtruth MOS scores are binary (*i.e.* $\hat{s}_{\text{MOS},t}^i = 1$ if the point prompt is within GT annotation, and $\hat{s}_{\text{MOS},t}^i = 0$ otherwise). On the other hand, fIoU follows the same formulation as in `FlowI-SAM`, *i.e.* predicting IoUs for foreground objects and yielding 0 for background regions.

Flow prompt generator consists of (i) a frozen SAM image encoder, where the dense spatial features are extracted from optical flow inputs at different frame gaps, followed by an averaging across frame gaps; (ii) a flow transformer, with the detailed architecture depicted in Figure 4.3c, where we first stack the input point

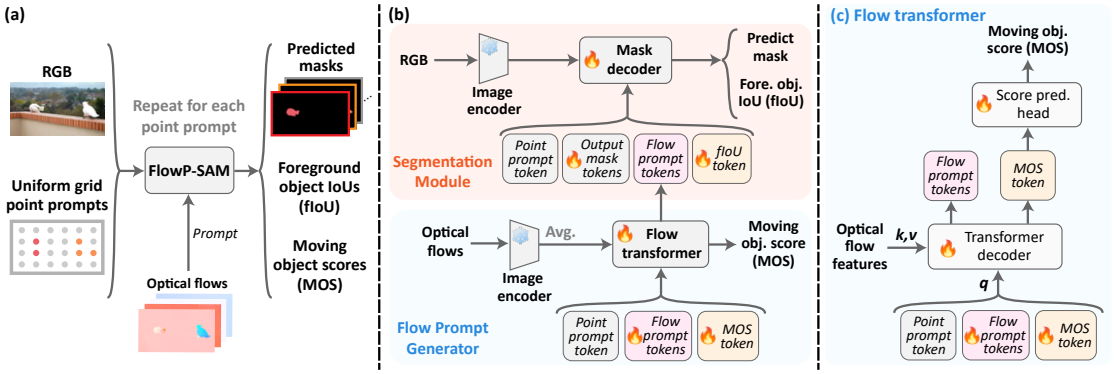


Figure 4.3: **Overview of `FlowP-SAM`. (a)** Inference pipeline of `FlowP-SAM`. **(b)** Architecture of `FlowP-SAM`. The flow prompt generator produces flow prompts to be injected into a SAM-like RGB-based segmentation module. Both modules take in the same point prompt token, which is obtained from a frozen prompt encoder. **(c)** Detailed architecture of the flow transformer. The input tokens function as queries within a lightweight transformer decoder, iteratively attending to dense flow features. The output moving object score (MOS) token is then processed by an MLP-based head to predict a score indicating whether the input point prompt corresponds to a moving object.

prompt (*i.e.* a positional embedding) with learnable flow prompts and moving object score (MOS) tokens to form queries. These queries then iteratively attend to dense flow features in a lightweight transformer decoder. There are two outputs from the flow prompt generator, namely, the refined flow prompts, and an MOS token, which is subsequently processed by an MLP-based head to yield a final moving object score.

Segmentation module has a structure that resembles the original SAM, except for two adaptions: (i) The IoU-prediction head is re-purposed to predict foreground object scores (fIoU) (same as the `FlowI-SAM`); (ii) The outputs tokens from flow prompt generator are injected as additional query inputs.

`FlowP-SAM` inference. Similar to `FlowI-SAM`, we prompt `FlowP-SAM` with single point prompts from a uniform grid to iteratively predict possible segmentation masks, together with MOS and fIoU estimations. These predicted scores are averaged, *i.e.* $(s^i_{\text{MOS},t} + s^i_{\text{fIoU},t})/2$, and then utilised to guide post-processing which involves the NMS and overlay of selected masks.

`FlowP-SAM` training. We train `FlowP-SAM` in an end-to-end fashion while keeping the SAM pre-trained prompt encoder and image encoders frozen. The flow transformer is trained from scratch,

$$\mathcal{L}_{\texttt{FlowP-SAM}} = \frac{1}{NT}\sum_{i,t}^{N,T}\Big(\mathcal{L}_{\text{BCE}}(M_t^i, \hat{M}_t^i) + \lambda_m \mathcal{L}_{\text{BCE}}(s^i_{\text{MOS},t}, \hat{s}^i_{\text{MOS},t}) + \lambda_f \|s^i_{\text{fIoU},t} - \hat{s}^i_{\text{fIoU},t}\|^2\Big) \tag{4.4}$$

where $\hat{M}_t^i$ corresponds to the groundtruth mask. $\hat{s}^i_{\text{MOS},t}$ and $\hat{s}^i_{\text{fIoU},t}$ indicate the groundtruth of two predicted scores, with λ_m and λ_f being the scale factors.

4.6 Sequence-level mask association

In this section, we outline our method for linking the frame-wise predictions for each moving object into a continuous track throughout the sequence. Specifically, we compute two types of masks: *frame-wise* masks M at the current frame using the model (`FlowI-SAM` and/or `FlowP-SAM`); and *sequence-level* masks $\mathcal{M}$, that are obtained by propagating the initial frame prediction with optical flow, we then update the mask of the current frame by making a comparison between the two. The following section details our update mechanism.

Update mechanism. This process aims to associate object masks across frames, as well as to determine whether the sequence-level results at a particular frame should be obtained directly from frame-level predictions at that frame or by propagating from previous frame results.

Specifically, given a sequence-level mask for object i at frame $t-1$ (*i.e.* $\mathcal{M}^i_{t-1}$), we first warp it to frame t using optical flow,

$$\mathcal{M}^i_{t\leftarrow t-1} = \text{warp}(\mathcal{M}^i_{t-1}, F_{t-1}) \tag{4.5}$$

We then consider three sets of masks: (i) the warped masks $\{\mathcal{M}^i_{t\leftarrow t-1}\}$; (ii) the frame-level predictions $\{M^i_t\}$ at frame t; and (iii) the frame-level predictions from neighboring frames (with Δt gap) after aligning them to the current frame using optical flow (*i.e.* $\{M^i_{t\leftarrow t+\Delta t}\}$). For each pair of mask sets, we perform a pairwise Hungarian matching based on the IoU score, resulting in three pairings in total. The Hungarian matched pairs can then reflect the *temporal consistency* across

these predictions based on the *transitivity* principle, i.e. if object i in (i) matches with object j in (ii) and object k in (iii), then the latter two objects must also match with each other. If such transitivity holds, we set the consistency score $c_i = 1$, and $c_i = 0$ otherwise.

This matching process is repeated for $\Delta t \in \{1, 2, -1, -2\}$, resulting in an averaged consistency score $\bar{c}_i$, which guides the following mask update:

$$\mathcal{M}_t^i = \begin{cases} M_t^i & \text{if } \bar{c}_i \geq 0.5 \\ \mathcal{M}_{t \leftarrow t-1}^i & \text{if } \bar{c}_i < 0.5 \end{cases} \tag{4.6}$$

where $\mathcal{M}_t^i$ denotes the sequence-level mask prediction for object i at frame t.

The rationale behind this is that the two methods have their own strengths and drawbacks: propagation is safe in preserving the object identity, but the mask quality degrades over time, whereas updating ensures high-quality masks but comes with the risk of mis-associating object identities. Thus, if the current frame-wise mask is temporally consistent, then we can be reasonably confident to update the mask, if not, then we choose the safer option and propagate the previous mask.

Note, we do this separately for each object $i \in N$, which gets updated or propagated independently. We lastly layer all objects back together (to its original order) and remove any overlaps to obtain the final sequence-level predictions.

4.7 Experiments

4.7.1 Datasets

Single-object benchmarks. For single-object motion segmentation, we adopt standard datasets, including DAVIS2016 [Perazzi et al., 2016], SegTrackv2 [Li et al., 2013], FBMS-59 [Ochs et al., 2014], and MoCA [Lamdouar et al., 2020]. Although SegTrackv2 and FBMS-59 include a few multi-object sequences, following the common practice [Yang et al., 2021a; Lamdouar et al., 2021], we treat them as single-object benchmarks by grouping all moving objects into a single foreground mask. MoCA stands for Moving Camouflaged Animals, designed as a camouflaged object detection benchmark. Following [Yang et al., 2021a; Lamdouar et al., 2021;

Xie et al., 2022], we adopt a filtered MoCA dataset by excluding videos with predominantly no locomotion.

Multi-object benchmarks. In terms of multi-object segmentation, we report the performance on DAVIS2017 [Pont-Tuset et al., 2017], DAVIS2017-motion [Pont-Tuset et al., 2017; Xie et al., 2022], and YouTube-VOS2018-motion [Xu et al., 2018; Xie et al., 2024b], where DAVIS2017 is characterised by predominantly moving objects, each annotated as distinct entities. For example, a man riding a horse would be separately labelled. In contrast, DAVIS2017-motion re-annotates the objects based on their joint movements such that objects with shared motion are annotated as a single entity. For example, a man riding a horse is annotated as a single entity due to their shared motion.

The YouTubeVOS2018-motion [Xie et al., 2024b] dataset is a curated subset of the original YouTubeVOS2018 [Xu et al., 2018]. It specifically excludes video sequences involving common motion, severe partial motion, and stationary objects, making it ideally suited for motion segmentation evaluation. Whereas, the original dataset also annotates many stationary objects and only provides partial annotations for a subset of moving objects.

Summary of evaluation datasets. To investigate the role of motion in object discovery and segmentation, we adopt all aforementioned benchmarks, which consist of only predominantly moving object sequences. Notably, for the evaluation of `FlowI-SAM`, we exclude the class-labelled DAVIS17 dataset, as commonly moving objects cannot be separated solely based on motion cues.

Training datasets. To adapt the RGB pre-trained SAM for moving object discovery and motion segmentation, we train both `FlowI-SAM` and `FlowP-SAM` first on the synthetic dataset introduced by [Xie et al., 2022], as described below, and then on real-world video datasets, including DAVIS16, DAVIS17, and DAVIS17-m.

4.7.2 Evaluation metrics

To assess the accuracy of predicted masks, we report intersection-over-union ($\mathcal{J}$), except for MoCA where only the ground-truth bounding boxes are given, we instead follow the literature [Yang et al., 2021a] and report the detection success

rate (SR). Regarding multi-object benchmarks, we additionally report the contour accuracy ($\mathcal{F}$) in the Appendix.

In this work, we differentiate between the frame-level and sequence-level methods, and adopt two distinct evaluation protocols: (i) Since frame-level methods generate the segmentation *independently* for each frame, we apply per-frame Hungarian matching to match the object masks between predictions with the groundtruth, before the evaluation; (ii) Conversely, sequence-level methods employ an extra step to link object masks across frames. As a result, the Hungarian matching is conducted globally for each sequence, *i.e.* the object IDs between predicted and groundtruth masks are matched once per sequence. Given the added complexity and potential errors during frame-wise object association, sequence-level predictions are often considered a greater challenge.

4.7.3 Implementation details

In this section, we summarise the experimental setting in our frame-level segmentation models. For more information regarding detailed architectures, hyperparameter settings, and sequence-level mask associations, please refer to the Appendix.

Flow computation. We adopt an off-the-shelf method (RAFT [Teed and Deng, 2020]) to estimate optical flow with multiple frame gaps at (1,-1) and (2,-2), except for YTVOS18-m and FBMS-59, where higher frame gaps at (3,-3) and (6,-6) are used to compensate for slow motion. Following common practice [Yang et al., 2021a; Xie et al., 2022], we convert optical flow into 3-channel RGB format using a standard color wheel [Baker et al., 2007].

Model settings. For both `FlowI-SAM` and `FlowP-SAM`, we follow the default SAM setting and adopt the first output mask token (out of four) for mask predictions. For `FlowI-SAM`, we deploy two versions of the pre-trained SAM image encoder, specifically ViT-B and ViT-H, to extract optical flow features.

Regarding `FlowP-SAM`, for efficiency reasons, we utilise ViT-B to encode optical flows and employ ViT-H as the image encoder for RGB frames. We initialise the flow prompt generator with 4 learnable flow prompt tokens, which are subsequently processed by a light-weight two-layer transformer decoder in the flow transformer

module.

Evaluation settings. At inference time, for `FlowI-SAM` with flow-only inputs, we input independent point prompts over a 10×10 uniform grid, while for `FlowP-SAM`, to take account for more complicated RGB textures, we consider a large grid size of 20×20.

Mask selection over multiple point prompts. During post-processing, we utilise the predicted scores (fIoU for `FlowI-SAM`, and an average of MOS and fIoU for `FlowP-SAM`) as guidance throughout the mask selection process: (i) Non-Maximum Suppression (NMS) is applied to filter out repeating masks and keep the ones with higher scores; (ii) The remaining masks are then ranked according to their scores and top-n masks are retained ($n = 5$ for `FlowI-SAM`, and $n = 10$ for `FlowP-SAM`); (iii) These n masks are overlaid by allocating masks with higher scores at the front.

Training settings. The training is performed in two stages, which involves synthetic pre-training on the dataset proposed by [Xie et al., 2022], followed by fine-tuning on the real DAVIS sequences, as detailed in the Appendix. YTVOS is not used for fine-tuning as there is only a low proportion of moving object sequences. We train both models in an end-to-end manner using the Adam Optimiser at a learning rate of $3e^{-5}$. The training was conducted on a single NVIDIA A40 GPU, with each mode taking roughly 3 days to reach full convergence.

4.7.4 Ablation study

In this section, we present a series of ablation studies on key designs in the per-frame `FlowP-SAM` model. We refer the reader to the Appendix.

Ablation studies for `FlowP-SAM`. As illustrated in Table 4.1, we start from the vanilla SAM and progressively add our proposed components. Note that, we adopt the same inference pipeline (*i.e.* the same point prompts and post-processing steps) for all predictions shown. Since the foreground object IoU (fIoU) is not predicted by the vanilla SAM, we instead apply default IoU predictions to guide the mask selection.

Table 4.1: **Ablation analysis of `FlowP-SAM`.** The study starts from the vanilla SAM checkpoint and progressively introduces new components (labelled in blue). "MOS" is short for the moving object score, and "fIoU" indicates the foreground object IoU. The results are shown for frame-level predictions.

	Stage	Predicted scores for post-processing	Flow prompt	FT mask decoder	DAVIS17 $\mathcal{J}\uparrow$	DAVIS16 $\mathcal{J}\uparrow$
		IoU	✗	✗	25.2	30.3
+	Train flow prompt generator	IoU	✓	✗	61.9	80.6
		(MOS+IoU)/2	✓	✗	63.7	81.4
+	Finetune segment -ation module	(MOS+IoU)/2	✓	✓	65.5	81.5
		(MOS+fIoU)/2	✓	✓	69.9	86.1

We *train the flow prompt generator* to simultaneously predict flow prompt tokens and moving object scores (MOS). The injection of flow prompts into the standard RGB-based SAM architecture results in notable enhancements, verifying the value of motion information for accurately determining object positions and shapes. Additionally, employing MOS as additional post-processing guidance yields further improvements.

Upon *finetuning the segmentation module*, we observe a slight enhancement in performance. Finally, substituting the default IoU predictions with fIoU scores achieves more precise mask selection, as evidenced by the improved results.

Discussion on the effectiveness of MOS and fIoU. As outlined in Sect. 4.3, the original SAM framework over-segments images into objects, parts, and sub-parts, which cannot be further distinguished using the default SAM IoU estimations. To adapt this setup for *object-only* discovery, we propose two new scores, *i.e.* MOS and fIoU, as new criteria to filter out non-object masks. These scores effectively assess the "objectness" of masks: MOS determines if the predicted masks represent moving objects, while fIoU evaluates whether the masks depict complete objects and are not background segments. The effectiveness of this adaptation is validated in Table 4.1, where replacing IoU estimations with MOS + fIoU scores leads to noticeable performance boosts.

4.7.5 Quantitative results

Given the distinct evaluation protocols outlined in Section 4.7.2, we report our method separately, with a frame-level analysis for `FlowI-SAM` (introduced in Sec-

Table 4.2: **Frame-level comparison on Video Object Segmentation benchmarks.** "†" indicates models that are trained without human annotations. For the results in the last row, we combine frame-level predictions from `FlowP-SAM` and `FlowI-SAM` (ViT-H).

			Multi-object benchmarks			Single-object benchmarks			
			YTVOS18-m	DAVIS17-m	DAVIS17	DAVIS16	STv2	FBMS	MoCA
Model	Flow	RGB	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	SR $\uparrow$
Flow-only methods									
COD [Lamdouar et al., 2020]	✓	✗	–	–	–	65.3	–	–	0.236
†MG [Yang et al., 2021a]	✓	✗	37.0	38.4	–	68.3	58.6	53.1	0.484
†EM [Meunier et al., 2023]	✓	✗	–	–	–	69.3	55.5	57.8	–
`FlowI-SAM` **(ViT-B)**	✓	✗	56.7	63.2	–	**79.4**	69.0	72.9	**0.628**
`FlowI-SAM` **(ViT-H)**	✓	✗	**58.6**	**65.7**	–	79.1	**70.1**	**75.1**	0.625
RGB-based methods									
†VideoCutLER [Wang et al., 2023b]	✗	✓	59.0	57.4	41.7	–	–	–	–
†Safadoust et al. [Safadoust and Güney, 2023]	✗	✓	–	59.3	–	–	–	–	–
MATNet [Zhou et al., 2020]	✓	✓	–	–	56.7	82.4	50.4	76.1	0.544
DystaB [Yang et al., 2021c]	✗	✓	–	–	–	82.8	74.2	75.8	–
AMC-Net [Yang et al., 2021b]	✓	✓	–	–	–	84.5	–	76.5	–
TransportNet [Zhang et al., 2021]	✓	✓	–	–	–	84.5	–	78.7	–
TMO [Cho et al., 2023]	✓	✓	–	–	–	85.6	–	79.9	–
`FlowP-SAM`	✓	✓	76.9	78.5	69.9	86.1	83.9	87.9	**0.645**
`FlowP-SAM+FlowI-SAM`	✓	✓	**77.4**	**80.0**	**71.6**	**86.2**	**84.2**	**88.7**	**0.645**

tion 4.4) and `FlowP-SAM` (introduced in Section 4.5), followed by a sequence-level evaluation.

Frame-level performance. Table 4.2 distinguishes between flow-only and RGB-based methods, where the former adopts optical flow as the only input modality, and the latter takes in RGB frames with optional flow inputs. Note that, the performance for some recent self-supervised methods is also reported, owing to the lack of the supervised baselines.

For flow-only segmentation, our `FlowI-SAM` (with both SAM image encoders) outperforms the previous methods by a large margin (>10%). For RGB-based segmentation, our `FlowP-SAM` also achieves state-of-the-art performance, particularly excelling at multi-object benchmarks. By combining these two frame-level predictions (`FlowI-SAM+FlowP-SAM`), we observe further performance boosts. This suggests the complementary roles of the flow and RGB modalities in frame-level segmentation, particularly when there are multiple moving objects involved. In particular, we show that using both models in tandem by layering `FlowI-SAM`'s predictions behind that of `FlowP-SAM` allows the model to fill in on missed predictions (such as motion blur, poor lighting, or small objects).

Table 4.3: **Sequence-level comparison on Video Object Segmentation benchmarks.** "†" indicates models that are trained without human annotations. "seq" indicates that our sequence-level predictions with object masks matched across frames. We adopt `FlowP-SAM` and `FlowI-SAM` (ViT-H) to obtain the results in the last row.

			Multi-object benchmarks			Single-object benchmarks			
			YTVOS18-m	DAVIS17-m	DAVIS17	DAVIS16	STv2	FBMS	MoCA
Model	Flow	RGB	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	SR $\uparrow$
Flow-only methods									
†SIMO [Lamdouar et al., 2021]	✓	✗	–	–	–	67.8	62.0	–	0.566
†Meunier et al. [Meunier and Bouthemy, 2023b]	✓	✗	–	–	–	73.2	55.0	59.4	–
†OCLR [Xie et al., 2022]	✓	✗	46.5	54.5	–	72.1	67.6	70.0	0.599
OCLR-real [Xie et al., 2022]	✓	✗	49.5	55.7	–	73.3	65.9	70.5	0.605
`FlowI-SAM` **(seq, ViT-B)**	✓	✗	51.9	60.0	–	**78.4**	66.9	69.0	**0.615**
`FlowI-SAM` **(seq, ViT-H)**	✓	✗	**53.8**	**61.5**	–	78.0	**67.7**	**71.5**	0.604
RGB-based methods									
UnOVOST [Luiten et al., 2020]	✗	✓	–	–	66.4	–	–	–	–
Propose-Reduce [Lin et al., 2021]	✗	✓	–	–	67.0	–	–	–	–
OCLR-flow [Xie et al., 2022] + SAM [Kirillov et al., 2023]	✓	✓	57.0	62.0	–	80.6	71.5	79.2	–
PMN [Lee et al., 2023]	✓	✓	–	–	–	85.6	–	77.8	–
Xie et al. [Xie et al., 2024b] + SAM [Kirillov et al., 2023]	✓	✓	71.1	70.9	–	86.6	**81.3**	**85.7**	–
DEVA [Cheng et al., 2023a]	✗	✓	–	–	70.4	87.6	–	–	–
UVOSAM [Zhang et al., 2024b]	✗	✓	–	–	**77.5**		–	–	–
`FlowP-SAM+FlowI-SAM` **(seq)**	✓	✓	**74.7**	**74.3**	71.0	**87.7**	80.1	82.8	**0.647**

Sequence-level performance. For flow-based segmentation, we apply the mask association technique introduced in Section 4.6 to obtain sequence-level predictions from per-frame `FlowI-SAM` results. To ensure a fair comparison, we additionally finetune the synthetic-trained OCLR [Xie et al., 2022] model on the real-world dataset (DAVIS) with groundtruth annotations provided, resulting in "OCLR-real" results. As shown by the top part of Table 4.3, `FlowI-SAM` (seq) demonstrates superior performance against OCLR-real, benefiting from the robust prior knowledge in pre-trained SAM.

For RGB-based segmentation, we obtain our sequence-level predictions by `FlowI-SAM` +`FlowP-SAM`. As shown in the lower part of Table 4.3, our method achieves outstanding performance across single- and multi-object benchmarks. Note, DAVIS17 annotations are class-based, which could be unclear and inconsistent with the class-agnostic unsupervised VOS methods. More discussion and results are provided in the Appendix.

4.7.6 Qualitative visualisations

In this section, example visualisations are provided across multiple datasets. We refer more visualisations to the Appendix.

Figure 4.4 illustrates the segmentation predictions based on only optical flow in-

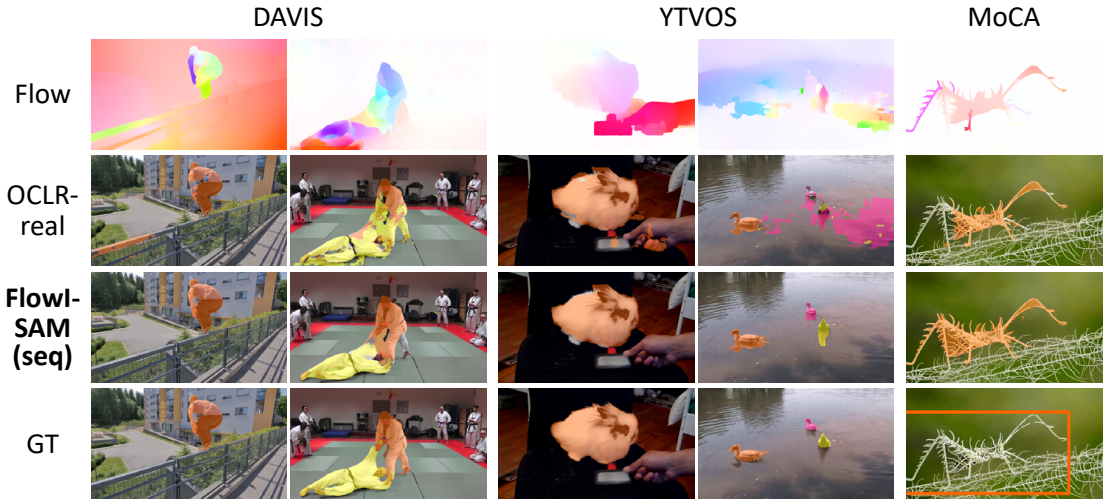


Figure 4.4: **Qualitative comparison of flow-only segmentation methods** on DAVIS (left), YTVOS (middle), and MoCA (right) sequences. Our `FlowI-SAM` (seq) successfully identifies moving objects from noisy optical flow background (*e.g.* the ducks in the fourth column).



Figure 4.5: **Qualitative comparison of RGB-based segmentation methods** on DAVIS (left), YTVOS (middle), and SegTrackv2 (right). While the previous method (the third row) struggles to disentangle multiple moving objects (*e.g.* mixed gold fishes in the second column), our `FlowP-SAM`+`FlowI-SAM` (seq) accurately separates and segments all moving objects.

puts. Compared to OCLR-real, our `FlowI-SAM` accurately identifies and disentangles the moving objects from the noisy backgrounds (*e.g.* the person in the first column and the ducks in the fourth column), as well as extracts fine structures (*e.g.* the camouflaged insect in the fifth column) from optical flow.

Figure 4.5 further provides the visualisations of the RGB-based method, where the

prior work (Xie et al. [Xie et al., 2024b] + SAM [Kirillov et al., 2023]) sometimes fails to (i) identify the moving objects (*e.g.* missing leopard in the fifth column); (ii) distinguish between multiple objects (*e.g.* entangled object segmentation in the second and fourth columns), while our `FlowI-SAM`+`FlowP-SAM` (seq) incorporates RGB-based prediction with flow prompts, resulting in the accurate localisation and segmentation of moving objects.

4.8 Discussion

In this paper, we focus on moving object segmentation in real-world videos, by incorporating per-frame SAM with motion information (optical flow) in two ways: (i) for flow-only segmentation, we introduce `FlowI-SAM` that directly takes in optical flow as inputs; (ii) for RGB-based segmentation (`FlowP-SAM`), we utilise motion information to generate flow prompts as guidance. The former (`FlowI-SAM`) is particularly effective in scenarios with predominant motion and/or where RGB information might introduce confusion, such as in moving object detection and camouflaged object discovery. Additionally, owing to simple texture and a low cross-domain gap in optical flow, it generalises to diverse domains beyond everyday videos. On the other hand, `FlowP-SAM` focuses on common videos where both RGB and motion are informative and can be utilised to resolve ambiguities for independent objects in common motion.

Both approaches deliver state-of-the-art performance in frame-level segmentation across single- and multi-object benchmarks. Additionally, we develop a frame-wise association method that amalgamates predictions from `FlowI-SAM` and `FlowP-SAM`, achieving sequence-level segmentation predictions that outperform existing methods on DAVIS16, DAVIS17-m, YTVOS18-m, and MoCA.

The major limitation of this work is its extended running time, attributed to the computationally heavy image encoder in the vanilla SAM. However, our approach is generally applicable to other prompt-based segmentation models. With the emergence of more efficient versions of SAM, we anticipate a significant reduction in inference time.

Acknowledgments. This research is supported by the UK EPSRC CDT in AIMS (EP/S024050/1), a Clarendon Scholarship, a Royal Society Research Professorship RP\R1\191132, and the UK EPSRC Programme Grant Visual AI (EP/T028572/1).

Appendices

Appendices for this chapter can be found in the online version of the paper.[1]

Statement of authorship

A statement of authorship for this paper is provided in Appendix A.

[1] https://www.robots.ox.ac.uk/~vgg/research/flowsam/

Chapter 5

Grounding Moving Object Segmentation in 3D Space and Time

This paper was submitted to the Advances in Neural Information Processing Systems (NeurIPS), 2026. Under review.

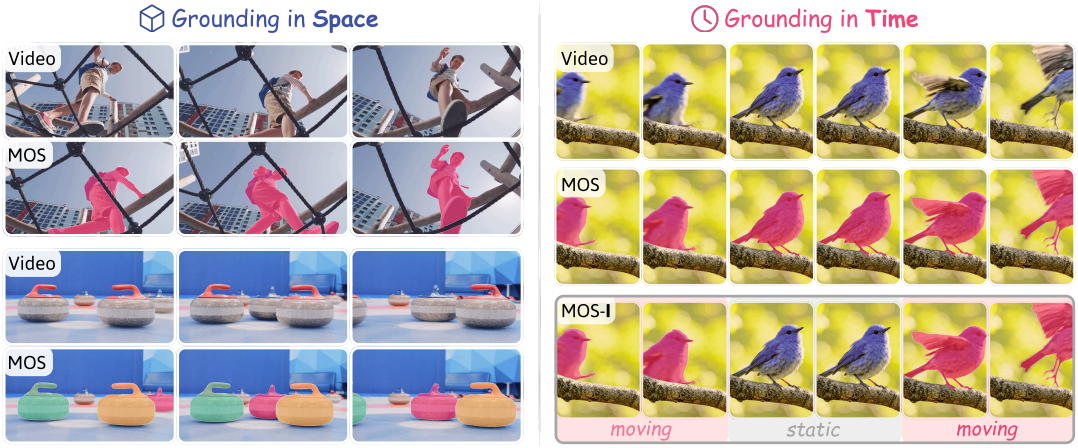


Figure 5.1: **Grounding Moving Object Segmentation (MOS) in 3D space and time. Left:** GMOS grounds MOS in 3D *space*, enabling reliable segmentation under challenging viewpoints and heavy depth parallax from moving cameras. **Right:** GMOS also grounds MOS in *time*: conventional MOS masks every object that moves at some point in the sequence, regardless of whether it is currently in motion. In contrast, our proposed *MOS-I* ("I" for Instantaneous) protocol requires predicting, for every frame, which objects are *actively moving* at that instant. This is illustrated in the bottom row, where a small bird lands, stops, then takes off, while the tree branch remains static throughout.

Abstract

Moving Object Segmentation (MOS) aims to discover, segment, and track objects that move independently of the camera. Current MOS methods, however, exhibit two fundamental limitations: they rely on pre-computed 2D auxiliary modalities such as optical flow or point trajectories that lack 3D geometric information, and they treat motion as a sequence-level attribute, overlooking the instantaneous motion state of each object. We address both by grounding MOS in 3D space and time, and propose GMOS, a framework that operates directly on RGB video to produce 3D-aware, temporally fine-grained segmentation of multiple moving objects, alongside a foreground–background variant GMOS-S for faster deployment. To support training and evaluation in this regime, we curate GMOS-2K, a dataset of 2,210 real-world videos with per-object temporal motion annotations drawn from five established Video Object Segmentation (VOS) benchmarks, and formalise MOS-I ("I" for instantaneous), a temporally fine-grained evaluation protocol with three complementary metrics. GMOS achieves state-of-the-art results across MOS, MOS-I, and Unsupervised VOS benchmarks, while running significantly faster than prior multi-object MOS methods and supporting online inference for streaming deployment.

5.1 Introduction

Moving object segmentation (MOS) aims to discover, segment, and track objects that move independently of the camera. This capability is foundational to applications such as autonomous driving [Chen et al., 2021; Li et al., 2023b], camouflaged object detection [Lamdouar et al., 2020; Yang et al., 2021a], and video surveillance [Nasaruddin et al., 2020; Weng et al., 2021], and has become increasingly central to recent 3D/4D scene reconstruction methods [Li et al., 2025b; Goli et al., 2025] that require motion masks to separate dynamic objects from the static scene.

Current MOS methods exhibit two fundamental limitations, as outlined in Table 5.1. *First*, they are not 3D-aware and remain fragile under large camera motion or depth parallax. This stems from their heavy reliance on pre-computed 2D auxiliary modalities (e.g. optical flow [Yang et al., 2021a; Xie et al., 2022; Lian et al., 2023] or point trajectories [Huang et al., 2025a; Karazija et al., 2024]), which lack 3D geometric information and incur substantial pre-processing overhead. *Second*, existing benchmarks assume that objects move throughout the video, leading methods to treat motion as a *sequence-level* attribute and to overlook the instantaneous motion state of each object. This coarse temporal granularity precludes online and real-time deployment.

We address both limitations by grounding MOS in *space* and *time*. Our approach rests on two observations: (*i*) the geometric representations learnt by recent feed-forward 3D reconstruction models already encode sufficient information to disentangle camera and object motion, and (*ii*) independent object motion is reliably identifiable from a short temporal window ($\sim$0.5 s).

Building on these insights, we propose GMOS, a proposer–propagator framework that operates directly on RGB video without any auxiliary pre-computed modalities. A per-frame *proposer* fuses geometric features from a π^3 encoder [Wang et al., 2026] with segmentation features from a SAM2 encoder [Ravi et al., 2024], jointly producing object mask proposals and per-frame motion-state predictions. A SAM2-based *propagator* then links these proposals into consistent tracks across frames. We further introduce GMOS-S ("S" for *single*), a streaming foreground–background variant that predicts all moving objects as a *single binary* mask and dispenses with the propagator for even faster inference.

Table 5.1: Comparison with Moving Object Segmentation (MOS) baselines. *Auxiliary inputs typically involve optical flow, 2D tracks and depths. The runtime measurement setting is detailed in the Appendix.

Method	Multi-object	3D-aware	Temporally fine-grained	Online	No aux. inputs*	Per-frame runtime (s)
FlowSAM [Xie et al., 2024c]	✓	✗	✗	✗	✗ (flow)	1.79
SegAnyMo [Huang et al., 2025a]	✓	✓	✗	✗	✗ (depth, track)	4.06
GeoMotion [He et al., 2026]	✗	✓	✗	✗	✗ (flow)	0.87
GMOS-S	✗	✓	✓	✓	✓	0.26
GMOS	✓	✓	✓	✓	✓	0.38

To support MOS in a more geometrically grounded and temporally fine-grained manner, we curate GMOS-2K, a collection of 2,210 real-world videos drawn from five established VOS benchmarks and annotated with per-frame motion states for every object. The subsets span complementary domains, including heavily occluded scenes, hand–object interactions, and wildlife and human-centric clips. Alongside this dataset, we formalise MOS-I ("I" for *instantaneous*), a temporally fine-grained MOS evaluation protocol. As illustrated in Figure 5.1 (right), MOS-I scores segmentation masks only on frames where objects are actively moving, and penalises false-positive predictions on static and background objects.

In summary, our contributions are as follows:

- We propose GMOS, a proposer–propagator framework that delivers 3D-aware, temporally fine-grained MOS from RGB video alone and supports online inference. We also introduce GMOS-S, a foreground–background variant for even faster deployment.
- We introduce a confidence-based training objective that automatically down-weights noisy frame-level motion annotations near static–moving transitions.
- We curate GMOS-2K, a dataset of 2,210 real-world videos drawn from five established VOS benchmarks and annotated with per-object temporal motion labels, providing a unified resource for both training and evaluating temporally fine-grained MOS methods.
- We formalise MOS-I, a temporally fine-grained evaluation protocol for MOS, together with three complementary metrics that jointly measure segmentation accuracy on actively moving frames and robustness to false-positive predictions on static and background objects.

GMOS achieves state-of-the-art results across MOS, MOS-I, and Unsupervised

VOS (UVOS) benchmarks, while running roughly three times faster than prior MOS methods (Table 5.1).

5.2 Method: GMOS

Given an RGB video $\mathcal{V} = \{I_t\}_{t=1}^{T}$, our goal is to predict, for every frame t and every independently moving object i, a segmentation mask together with its motion state,

$$\left\{ (\hat{M}_t^{(i)}, \hat{m}_t^{(i)}) \right\}_{t=1,\dots,T;\, i=1,\dots,N} = \Phi_{\text{GMOS}}(\mathcal{V}), \tag{5.1}$$

where $\hat{M}_t^{(i)} \in \{0,1\}^{H \times W}$ is a binary segmentation mask, $\hat{m}_t^{(i)} \in \{0,1\}$ is the motion state with $\hat{m}_t^{(i)} = 1$ indicating that object i is moving at frame t. GMOS produces these outputs jointly, and therefore covers both MOS-I (Section 5.4.2) and the conventional MOS settings.

Figure 5.2 (left) provides an overview of GMOS, which follows a *proposer-propagator* design. The proposer operates on a short local window and outputs, for the middle frame, a set of N per-object proposals, each comprising a moving object mask and a predicted motion state. The propagator then links these per-frame proposals into consistent object tracks across the full video.

5.2.1 GMOS proposer

As illustrated in Figure 5.2 (right), the proposer takes raw video frames as its sole input and predicts, for each of N object queries, a segmentation mask together with auxiliary outputs: a confidence score, a motion state, and an estimated mask IoU.

Geometric encoder. To achieve *3D-awareness*, we extract geometric features using a frozen π^3 encoder [Wang et al., 2026]. Rather than feeding the entire sequence, we process a short temporal window $[t-n,\, t+n]$ centred at the query frame t, sufficient for distinguishing independent object motion from camera-induced parallax. After π^3 encoding, we retain only the *middle-frame* feature $f_t^{\text{geo}} \in \mathbb{R}^{h_g \times w_g \times d_g}$, as inter-frame correspondence is already encoded in it through the model's cross-frame attention.

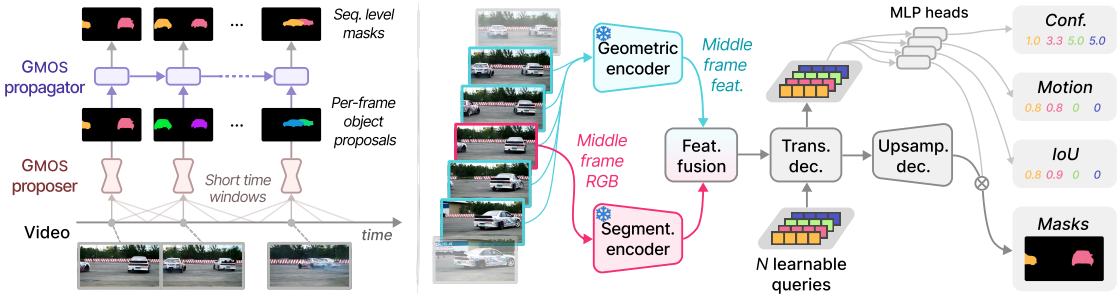


Figure 5.2: **Overview of GMOS. Left:** The overall *proposer–propagator* design. The proposer operates on a short temporal window (around 0.5 s) and outputs per-frame object proposals, which the propagator links into coherent tracks across the full video. **Right:** The GMOS proposer. A frozen π^3 geometric encoder ingests frames for a short temporal window and a frozen SAM2 segmentation encoder processes the middle frame of the window, to make predictions for the middle frame. Their features are fused and passed through a transformer decoder with N learnable object queries, decoded into segmentation masks for moving objects alongside motion states, estimated IoUs, and confidence scores.

Segmentation encoder. Since geometric features alone encode little about object segmentation cues, we leverage the strong objectness prior of a frozen SAM2 encoder [Ravi et al., 2024]. Only the query frame I_t is passed through SAM2, yielding a bottleneck feature $f_t^{\text{seg}} \in \mathbb{R}^{h \times w \times d}$ together with high-resolution features that are later used by the mask head.

Feature fusion. The geometric feature f_t^{geo} is typically at a lower spatial resolution than the segmentation one f_t^{seg}, i.e. $h_{\text{g}} < h$. We therefore bilinearly interpolate f_t^{geo} to (h, w), concatenate the two features channel-wise, and apply a linear projection to obtain the fused feature $f_t \in \mathbb{R}^{h \times w \times d}$. A light-weight transformer self-attention encoder is then applied to f_t to enforce global spatial consistency.

Transformer decoder. Following the convention in query-based segmentation [Cheng et al., 2021; Li et al., 2023a], we introduce N learnable object queries $\{q^{(i)}\}_{i=1}^{N}$ with $q^{(i)} \in \mathbb{R}^d$ and feed them into a two-way transformer decoder [Kirillov et al., 2023] together with f_t. Each layer alternates query self-attention with query-to-feature and feature-to-query cross-attention, producing refined query tokens $\{q_t^{*(i)}\}$ and refined dense features $f_t^* \in \mathbb{R}^{h \times w \times d}$.

Mask head. The refined dense features f_t^* are upsampled to full resolution through transposed convolutions, with the SAM2 high-resolution features added as skip connections. An MLP maps each refined query $q_t^{*(i)}$ to a mask embedding,

and the per-query mask $\hat{M}_t^{(i)}$ is obtained by dot product with the upsampled features, following the hypernetwork design in [Cheng et al., 2021]. During training, queries are matched to ground truth via bipartite Hungarian assignment. Note, an object's ground-truth mask is defined as *empty* whenever the object is not moving, so supervision jointly shapes the mask and the motion state.

Motion and IoU heads. Two independent MLP heads take in the refined query $q_t^{*(i)}$. The *motion head* leads to the probability $\hat{m}_t^{(i)}$ that object i is moving at frame t, and the *IoU head* produces a scalar $\hat{u}_t^{(i)}$ estimating the quality of $\hat{M}_t^{(i)}$ in terms of its expected IoU against the ground truth.

Confidence prediction. The instantaneous motion state is inherently ambiguous near transitions between motion and rest, where the ground truth switches abruptly between a full mask with $m_t^{(i)} = 1$ and an empty mask with $m_t^{(i)} = 0$. To mitigate this label noise, we follow [Wang et al., 2024c] and introduce a learnable per-query confidence $\hat{c}_t^{(i)} \in [0, 1]$, predicted by a final MLP head with sigmoid output. The resulting confidence modulates the mask and motion losses (see Equation 5.2).

Training objective. The total proposer loss reads

$$\mathcal{L} = \lambda_{\text{iou}}\mathcal{L}_{\text{iou}} + \tilde{c}\Big(\lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{motion}}\mathcal{L}_{\text{motion}}\Big) - \Big[\lambda_{\text{conf}}^{\text{mov}}\, m + \lambda_{\text{conf}}^{\text{sta}}\,(1-m)\Big]\,\log\tilde{c} \quad (5.2)$$

The loss is averaged over object queries i and frames t, with the corresponding indices dropped for clarity. Here, $\mathcal{L}_{\text{mask}}$ is a mixture of focal and dice losses, $\mathcal{L}_{\text{motion}}$ is a binary cross-entropy on the motion state, and $\mathcal{L}_{\text{iou}}$ is an MSE loss between the predicted IoU and the actual IoU against the ground truth.

The mask and motion losses are modulated by a rescaled confidence $\tilde{c} = (c_{\max} - 1)\,\hat{c} + 1 \in [1, c_{\max}]$, and the last term is an adversarial confidence regulariser [Wang et al., 2024c] that raises $\tilde{c}$ on clean, easy samples the model can solve precisely. For difficult examples, such as a walking pedestrian coming to a stop, the mask and motion losses are large due to label ambiguity around the transition. The model therefore lowers $\tilde{c}$ to attenuate the multiplicative term, paying a small $\log\tilde{c}$ penalty in exchange.

In practice, the confidence would be driven up only for static samples, since predicting an empty mask for a static object is trivially easier than predicting a precise moving-object mask, failing to address the transition noise. We therefore *split* the regularisation by the ground-truth motion state m, using distinct coefficients $\lambda_{\text{conf}}^{\text{mov}}$ and $\lambda_{\text{conf}}^{\text{sta}}$ to calibrate moving and static samples independently.

GMOS-S: *a foreground–background variant.* For efficiency-critical deployment, we additionally instantiate GMOS-S ("S" for *single* binary mask prediction), which predicts all moving objects into a single foreground mask. GMOS-S shares the encoders, feature fusion, and self-attention module with the GMOS proposer, but (i) removes the query-based transformer decoder, (ii) replaces the per-query mask head with a single convolutional decoder that outputs one binary foreground mask per frame, and (iii) dispenses with the follow-up propagator entirely. This makes GMOS-S an end-to-end model that produces instantaneous (per-frame) motion segmentation and achieves lower inference latency than GMOS. Full architectural details are provided in the Appendix.

5.2.2 GMOS propagator

For every frame t, the proposer yields a set of N candidate tuples $\{(\hat{M}_t^{(i)}, \hat{m}_t^{(i)}, \hat{u}_t^{(i)}, \hat{c}_t^{(i)})\}_{i=1}^{N}$. The role of the propagator is to lift these per-frame proposals $\{\hat{M}_t^{(i)}\}$ into coherent object tracks across the full video, guided by the predicted motion probabilities $\{\hat{m}_t^{(i)}\}$ and estimated IoUs $\{\hat{u}_t^{(i)}\}$. The confidence scores $\{\hat{c}_t^{(i)}\}$ are not used here, as they primarily serve as regularisers during proposer training. We build this process on a SAM2 video predictor [Ravi et al., 2024] used as a mask-conditioned tracker, and define an elementary propagation step from which we build online and offline procedures.

Basic propagation step. Given a start frame and a direction, SAM2 propagates the currently tracked masks frame by frame. At each visited frame t, the proposer predictions are first filtered by thresholding on the motion probabilities $\{\hat{m}_t^{(i)}\}$ and estimated IoUs $\{\hat{u}_t^{(i)}\}$. The remaining high-quality moving-object proposals are then matched to the propagated masks via Hungarian assignment based on mask IoU, and the set of active tracks is updated according to one of two cases:

- *Addition.* If a proposer prediction matches no existing SAM2 track, it is instantiated as a new mask track starting from frame t.
- *Reinforcement.* If a proposer prediction matches an existing SAM2 track at high IoU, it is injected as an additional prompt for that track, suppressing drift.

For both cases above, the track's motion state at frame t is set to moving. Conversely, if an existing track has a propagated mask at frame t but no proposer prediction falls on the same region (e.g. the object was moving earlier but has become static), the track's motion state at frame t is set to static.

Online procedure. The online procedure initialises the tracker from the *first frame* and runs the basic propagation steps in a single causal forward pass. This streams a mask and motion state for each frame as soon as it is processed, supporting MOS-I. Since the sequence is viewed only once and causally, masks cannot be recovered for frames preceding an object's first observed motion, making the online procedure incompatible with the conventional MOS setting that requires a mask at every frame.

Offline procedure. The offline procedure builds on top of the online results with a second propagation stage. It first selects an *anchor frame* $t^\star$ from the online output, defined as the frame whose moving object proposals carry the highest aggregate predicted IoU. The basic propagation step is then run *bi-directionally*, forward from $t^\star$ to T and backward from $t^\star$ to 1, yielding a dense mask for every object at every frame. This mode can be used for both MOS and MOS-I. Further algorithmic details and pseudo-code are provided in the Appendix.

5.3 Dataset: GMOS-2K

We introduce GMOS-2K, a video moving object segmentation dataset with temporally fine-grained motion labels. Beyond ground-truth masks for moving objects, it specifies the exact temporal windows during which each object is in motion, enabling precise training and evaluation of time-sensitive motion segmentation methods.

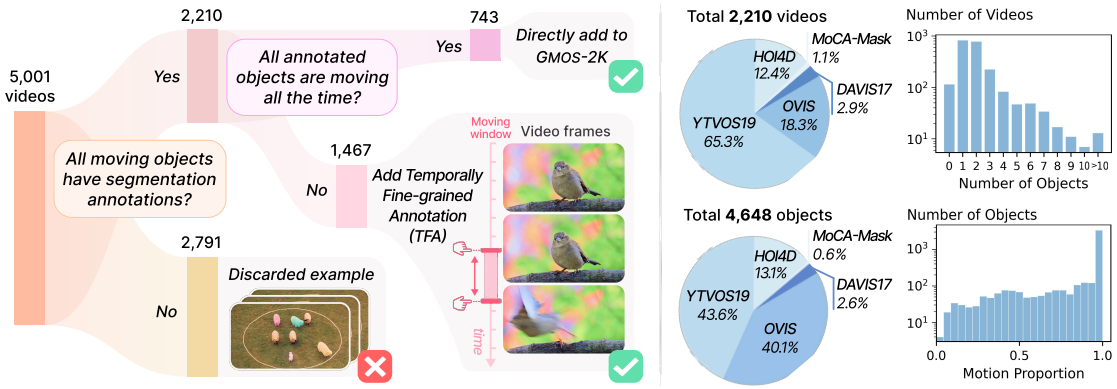


Figure 5.3: **GMOS-2K overview. Left:** Curation pipeline. 5,001 videos from five VOS datasets are filtered by two criteria, with 743 added directly to GMOS-2K and 1,467 requiring Temporal Fine-grained Annotation (TFA), which labels per-object motion intervals along the time axis. **Right:** Dataset statistics. Pie charts show the per-subset distribution of videos and objects. The upper histogram reports the number of moving objects per video, and the lower one reports the per-object motion proportion, i.e. the fraction of the sequence in which each object is actively moving.

Curation pipeline. Temporal motion labels are substantially cheaper to obtain than dense segmentation masks. We therefore build upon five existing video object segmentation datasets (DAVIS17 [Pont-Tuset et al., 2017], YTVOS19 [Xu et al., 2018], OVIS [Qi et al., 2022], MoCA-Mask [Cheng et al., 2022b], and HOI4D [Liu et al., 2022]) and annotate them with per-object motion timestamps. Figure 5.3 (left) illustrates the data filtering and annotation pipeline. Starting from 5,001 candidate videos, we apply two sequential filters:

(*i*) *Do all independently moving objects in the video have ground-truth segmentation masks?*

If yes, the sequence proceeds to the second question, otherwise it is filtered out. Note, an object is considered "moving" if it undergoes motion independent of the camera for at least one frame. This question concerns only moving-object masks. Static objects without masks are permissible.

(*ii*) *Are all annotated objects moving throughout the entire sequence?*

If yes, no additional annotation is needed. Otherwise, the video is forwarded for Temporally Fine-grained Annotation (TFA).

Temporally Fine-grained Annotation (TFA). Given a video sequence, TFA aims to annotate time intervals on the temporal axis that specify when objects are moving, with each object annotated independently. The annotation process follows

that for video temporal grounding. Further details on the annotation interface, guidelines, and quality controls are provided in the Appendix.

Dataset statistics. GMOS-2K comprises 2,210 videos with 4,648 annotated moving objects, drawn from five constituent subsets that span diverse domains: heavily occluded scenes (OVIS), hand-object interactions (HOI4D), and wildlife and human-centric clips (DAVIS17, YTVOS19, MoCA-Mask). Example sequences are shown in the Appendix, with instantaneously moving objects labelled.

The pie charts in Figure 5.3 (right) show the per-subset distribution of videos and objects. OVIS contributes a disproportionately large share of objects, reflecting its dense multi-object scenes. The accompanying histograms characterise two complementary aspects of the dataset. The per-video object count (top) concentrates on sequences with 0–4 moving objects, with a long tail extending beyond 4. The motion proportion (bottom), defined as the fraction of frames in which an object is actively moving, peaks at fully moving objects, with the remaining cases distributed roughly uniformly across lower values.

The training and test splits consist of 1,930 and 280 videos, respectively. All five subsets contribute to the training split, while DAVIS17 and YTVOS19 additionally contribute to the test split designed for MOS-I evaluation, renamed DAVIS17-IM and YTVOS19-IM ("IM" for Instantaneous Motion) to distinguish them from the conventional MOS datasets. Full per-subset statistics are reported in the Appendix.

5.4 Experiments

5.4.1 Training datasets

We train GMOS on a composite dataset of 14,171 videos (782K annotated frames), combining three data sources. (*i*) *Synthetic data.* Kubric [Greff et al., 2022], PointOdyssey [Zheng et al., 2023], and DynamicReplica [Karaev et al., 2023] provide 10,526 videos with pixel-perfect ground truth and motion labels. (*ii*) *Real moving-object data (GMOS-2K).* The five GMOS-2K training subsets (Section 5.3) contribute 1,930 real-world videos with temporally fine-grained motion labels. (*iii*) *Real static-scene data.* 1,715 clips from the Mannequin Challenge dataset [Li

et al., 2019b], in which people remain frozen while the camera moves freely, provide static-scene supervision that teaches the model to distinguish depth parallax from independent object motion. The detailed per-dataset statistics are provided in the Appendix.

5.4.2 Benchmarks for evaluation

We evaluate GMOS on the following video object segmentation tasks.

Moving Object Segmentation (MOS) predicts, at every frame, a segmentation mask for every object that moves at any point in the sequence, regardless of its per-frame motion state. We evaluate on established MOS benchmarks: DAVIS16-M [Huang et al., 2025a] ("M" for "Moving"), DAVIS17-M [Dave et al., 2019], SegTrackv2 [Li et al., 2013], FBMS-59 [Ochs et al., 2014], and MoCA [Lamdouar et al., 2020], along with YTVOS19-M, a new benchmark derived from the GMOS-2K test set. Following standard practice, all benchmarks report region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$, except MoCA, which uses detection success rate (SR) [Yang et al., 2021a].

MOS-I ("I" for *instantaneous*) is a temporally fine-grained evaluation protocol for MOS that (i) retains the sequence-level object identity association of MOS, (ii) scores segmentation masks only on frames where an object is actively moving, and (iii) explicitly penalises false-positive predictions on static or background objects. We instantiate it on two new benchmarks derived from the GMOS-2K test set, namely DAVIS17-IM and YTVOS19-IM ("IM" for "Instantaneous Motion"), and report performance on three complementary metrics.

- *Moving-object Jaccard* ($\mathcal{J}_{\text{mov}}$) – segmentation accuracy on actively moving frames. For each annotated object, $\mathcal{J}$ is averaged over the frames in which it is labelled as moving in GMOS-2K.
- *False-positive count* (FP count) – penalisation of static and background predictions. We measure the average number of static or background objects per frame that are incorrectly predicted as moving. While $\mathcal{J}_{\text{mov}}$ rewards detecting true motion, FP count penalises spurious motion claims.
- *Mean temporal IoU* (mtIoU) – joint measure of moving-object detection and

false-positive suppression. For each sequence, we compute a temporal IoU between predicted and ground-truth moving objects, counted *per object per frame*. The *union* covers all (object, frame) entries marked as moving by either the prediction or the ground truth. The *intersection* covers entries where a predicted and a ground-truth moving object are matched by region similarity $\mathcal{J}$ at threshold k. Sweeping $k \in \{0.5, 0.55, \ldots, 0.95\}$ and averaging the resulting tIoUs yields the final mtIoU.

Unsupervised Video Object Segmentation (UVOS) requires automatically segmenting salient objects in a video, where targets are defined by perceptual salience and need not be moving. We report results on DAVIS16 [Perazzi et al., 2016] and DAVIS17 [Pont-Tuset et al., 2017] using the standard region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$.

5.4.3 Implementation details

We detail the design choices of GMOS below. Further implementation details are provided in the Appendix.

Architecture. GMOS uses pretrained π^3 [Wang et al., 2026] (DINOv2 [Oquab et al., 2024] ViT-L/14) and SAM2 [Ravi et al., 2024] (Hiera-L [Ryali et al., 2023]) as the geometric and segmentation encoders, both kept frozen. Bottleneck features have spatial resolution 64×64 and channel dimension 256. The trainable transformer encoder and decoder each have 3 layers and 8 attention heads. We use 100 object queries and predict masks at 256×256.

Input setting. The GMOS proposer operates on 5 frames sampled uniformly from a temporal window centred on the target frame. The window size is randomly sampled between 0.16 s and 1.00 s during training to expose the model to a range of motion speeds, and fixed at 0.5 s at inference.

Training setup. We optimise with AdamW [Loshchilov and Hutter, 2019] using learning rate 1×10^{-4}, weight decay 0.01, gradient clipping at norm 1.0, and a cosine schedule with 2,000 warmup steps. Training runs on 2 NVIDIA RTX A6000 GPUs for approximately 2 days, with an effective batch size of 8. For the proposer loss

Table 5.2: **Comparison with Moving Object Segmentation (MOS) baselines.** The Δ vs. prev. SoTA row reports the difference between our best result (GMOS or GMOS-S) and the previous state of the art.

Methods	Multi-Object				Foreground–Background										
	YTVOS19-M		DAVIS17-M		YTVOS19-M		DAVIS17-M		DAVIS16-M		STv2		FBMS		MoCA
	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	SR↑
OCLR-TTA [Xie et al., 2022]	-	-	48.4	49.9	-	-	76.0	75.3	80.1	76.9	72.1	75.3	69.9	68.3	0.559
ABR [Xie et al., 2024b]	-	-	50.9	51.2	-	-	74.6	75.2	70.2	73.7	76.3	81.1	81.9	79.6	-
FlowSAM [Xie et al., 2024c]	66.6	68.2	74.6	81.0	52.4	54.3	87.9	87.0	85.7	83.8	77.9	83.9	82.8	80.9	0.643
SegAnyMo [Huang et al., 2025a]	64.6	66.2	77.4	83.6	70.9	73.0	90.0	89.0	89.2	89.7	76.3	85.4	78.3	82.8	0.675
GeoMotion [He et al., 2026]	-	-	-	-	65.4	65.7	82.2	82.3	83.5	84.3	77.3	84.3	72.5	78.5	0.615
GMOS	**79.9**	**82.2**	**79.7**	**86.0**	**85.1**	**84.6**	90.0	90.6	88.7	90.3	**82.0**	**88.3**	**91.9**	**93.0**	**0.837**
GMOS-S	-	-	-	-	81.4	81.8	**90.7**	**93.1**	**90.0**	**93.2**	78.9	85.9	83.0	84.9	0.785
Δ vs. prev. SoTA	+13.3	+14.0	+2.3	+2.4	+14.2	+11.6	+0.7	+4.1	+0.8	+3.5	+4.1	+2.9	+9.1	+10.2	+0.162

Table 5.3: **Comparison with MOS-I baselines.** "I" is short for "instantaneous".

Methods	YTVOS19-IM			DAVIS17-IM		
	mtIoU↑	$\mathcal{J}_{\text{mov}}\uparrow$	FP count↓	mtIoU↑	$\mathcal{J}_{\text{mov}}\uparrow$	FP count↓
OCLR-TTA [Xie et al., 2022]	-	-	-	28.3	54.1	0.550
ABR [Xie et al., 2024b]	-	-	-	36.0	57.0	0.271
FlowSAM [Xie et al., 2024c]	21.1	62.8	3.085	22.7	75.3	4.101
SegAnyMo [Huang et al., 2025a]	55.9	63.2	0.320	67.0	81.7	0.253
GMOS (*online*)	**64.2**	76.4	**0.197**	**72.9**	83.5	**0.075**
GMOS	63.4	**76.6**	0.238	**72.9**	**84.2**	0.079
Δ vs. prev. SoTA	+8.3	+13.4	+0.123	+5.9	+2.5	+0.178

Table 5.4: **Comparison with Unsupervised VOS (UVOS) baselines.**

Methods	DAVIS17		DAVIS16	
	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
OCLR-TTA [Xie et al., 2022]	43.7	44.5	80.8	76.8
ABR [Xie et al., 2024b]	45.0	45.3	71.8	73.2
DEVA [Cheng et al., 2023a]	70.4	76.4	87.6	90.2
FlowSAM [Xie et al., 2024c]	71.2	76.3	87.1	84.9
SegAnyMo [Huang et al., 2025a]	66.5	71.4	**90.6**	91.0
GeoMotion [He et al., 2026]	-	-	84.5	85.0
GMOS	**76.2**	**80.6**	90.1	**91.5**
GMOS-S	-	-	87.9	90.8
Δ vs. prev. SoTA	+5.0	+4.2	−0.5	+0.5

in Equation 5.2, we set $(\lambda_{\text{mask}}, \lambda_{\text{motion}}, \lambda_{\text{iou}}, \lambda^{\text{mov}}_{\text{conf}}, \lambda^{\text{sta}}_{\text{conf}}) = (20, 1, 1, 0.1, 0.001)$, with a rescaled confidence upper bound $c_{\max} = 5$. The mask loss combines focal (α=0.25, γ=2) and dice losses in a 20:1 ratio.

5.4.4 Quantitative results

Table 5.2 reports results on the conventional MOS setting, where moving-object masks are predicted at the sequence level. GMOS, and its foreground–background variant GMOS-S, achieves new state-of-the-art performance across all benchmarks. Gains are most pronounced on YTVOS19-M, whose denser and more varied object interactions expose the limitations of prior flow- and trajectory-based methods. GMOS also remains robust on camouflaged objects (MoCA), where segmentation must rely exclusively on motion cues.

As shown in Table 5.3, performance gaps widen further on MOS-I, which requires predictions at finer temporal granularity. Since prior methods treat motion as a sequence-level attribute and cannot capture instantaneous motion states, GMOS surpasses them by a substantial margin on all three metrics. We additionally evaluate an online variant of GMOS (Section 5.2.2), which performs on par with

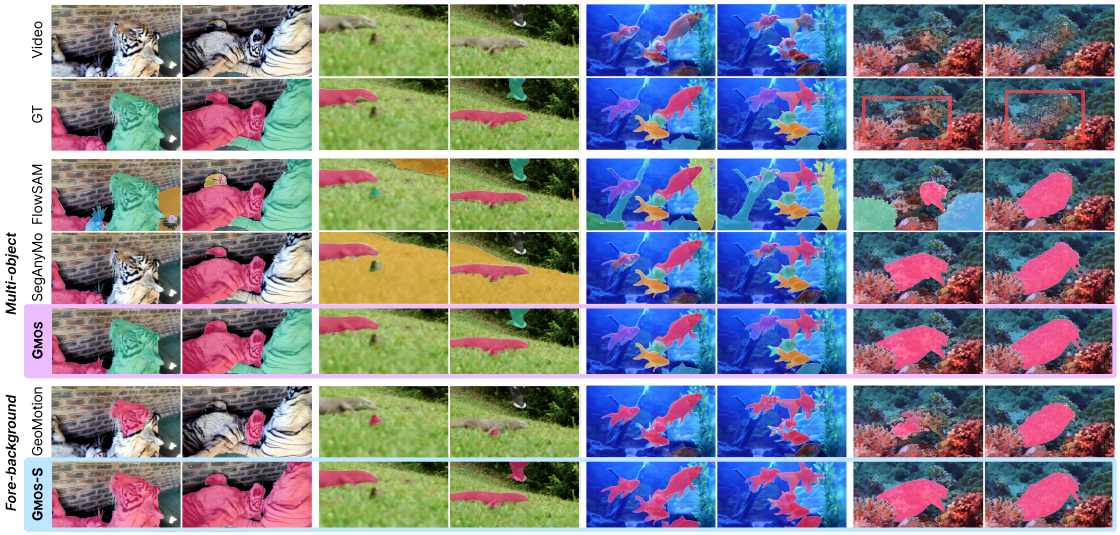


Figure 5.4: **Qualitative comparison on the MOS task.** Example videos are sampled from YTVOS19 (first two columns), DAVIS17 (third column), and MoCA (last column). The middle block shows multi-object results, and the bottom block shows foreground–background results.

the default offline mode. This can be attributed to the instantaneous nature of MOS-I, where a short causal window already provides sufficient motion evidence. The offline mode's second global propagation even slightly hurts FP count and mtIoU, as the bidirectional pass amplifies any borderline false positives from the first pass. The online mode, however, is restricted to the MOS-I setup, since a single forward pass cannot recover masks for objects that are initially static and only later begin to move. The offline procedure therefore remains our default across all tasks, including MOS and UVOS.

We also report performance on Unsupervised Video Object Segmentation (UVOS), which targets the prediction of salient objects regardless of their motion state. Although UVOS lies outside our primary focus of segmenting only moving objects, Table 5.4 shows that GMOS generalises well to this adjacent task, achieving state-of-the-art results on the multi-object DAVIS17 benchmark and remaining on par with the prior best on the single-object DAVIS16.

5.4.5 Qualitative results

Figure 5.4 compares GMOS against representative baselines on sequences from YTVOS19, DAVIS17, and MoCA. Regarding multi-object predictions, FlowSAM [Xie et al., 2024c] tends to over-segment moving objects (e.g. treating the coral as moving), while SegAnyMo [Huang et al., 2025a] misses moving objects or bleeds masks

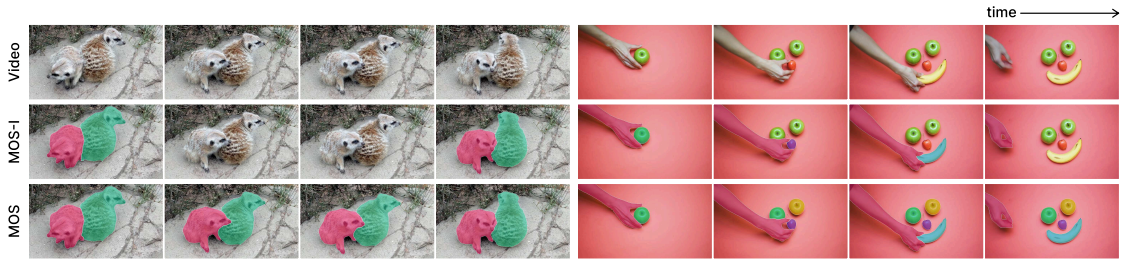


Figure 5.5: **GMOS on in-the-wild videos.** Two in-the-wild sequences (sourced outside our training or test datasets) illustrate our MOS and MOS-I predictions. Under MOS-I, instantaneously moving objects are segmented, with object identities consistently associated across frames. Under MOS, GMOS produces full segmentation masks for every object that moves at any point in the sequence, regardless of its motion state at a given frame.

into the background. In contrast, GMOS assigns each moving object a consistent identity and boundary that closely follow the ground truth, even under heavy occlusion. Under the foreground–background protocol, GeoMotion [He et al., 2026] captures prominent motion but overlooks instances near image boundaries (e.g. the right tiger), whereas GMOS-S recovers them reliably.

Figure 5.5 illustrates GMOS on in-the-wild videos, contrasting MOS-I and MOS predictions. In the left sequence, the meerkats become static mid-sequence. Our MOS-I predictions correctly identify the static states, while MOS produces consistent segmentation across frames. In the right sequence, a hand arranges fruits one piece at a time. Our MOS-I predictions correctly release each piece once it comes to rest, whereas MOS keeps them masked throughout.

5.5 Related works

Unsupervised Video Object Segmentation (UVOS) aims to identify salient objects in videos without any annotation at inference time, in contrast to semi-supervised VOS [Perazzi et al., 2016; Pont-Tuset et al., 2017; Xu et al., 2018; Oh et al., 2019; Cheng and Schwing, 2022; Yang and Yang, 2022; Cheng et al., 2024a; Ravi et al., 2024], which propagates masks from first-frame ground truths. Early methods adopt two-stream architectures that fuse motion and appearance inputs [Jain et al., 2017; Lu et al., 2019; Zhou et al., 2020; Yang et al., 2021b; Ren et al., 2021; Liu et al., 2021; Ji et al., 2021; Zhang et al., 2021; Pei et al., 2022], while others rely primarily on saliency or attention mechanisms [Wang et al., 2019a; Yang et al., 2019d; Fan et al., 2019; Lee et al., 2023; Cho et al., 2023;

Yuan et al., 2023; Zhou et al., 2021; Liu et al., 2024; Cho et al., 2024; Lee et al., 2024; Song et al., 2024b]. More recent works scale UVOS with new data strategies and modular pipeline designs: VideoCutLER [Wang et al., 2023b] scales training via copy-paste augmentation transferred from unsupervised image instance segmentation, DEVA [Cheng et al., 2023a] composes an image-level segmenter with a separate temporal propagator, and SAM-based approaches [Kirillov et al., 2023; Ravi et al., 2024; Carion et al., 2026; Zhang et al., 2024b] build UVOS pipelines around pre-trained foundation segmenters. Despite significant progress, UVOS methods typically target the most salient objects, which do not necessarily correspond to objects that are actually moving.

Motion Segmentation (MS) seeks to separate independently moving regions from the static background, producing a binary foreground–background mask. Mainstream approaches take optical flow as the motion signal and cluster pixels according to similar motion patterns [Papazoglou and Ferrari, 2013; Bideau and Learned-Miller, 2016b; Tokmakov et al., 2017; Tokmakov et al., 2019; Yang et al., 2021a; Lamdouar et al., 2020; Dave et al., 2019; Yang et al., 2021c; Choudhury et al., 2022; Yang et al., 2019c; Meunier et al., 2023; Meunier and Bouthemy, 2023a]. However, these flow-based approaches are inherently bound by the quality of the pre-computed optical flow, and also vulnerable to large camera motion and induced depth parallax. A complementary line of work aggregates motion cues over longer temporal windows via point trajectories [Brox and Malik, 2010; Ochs and Brox, 2011; Keuper et al., 2015; Fragkiadaki et al., 2012]. Recent advances in point tracking [Karaev et al., 2025; Doersch et al., 2022; Doersch et al., 2023] have substantially improved long-range correspondence, enabling segmentation to be learnt directly from dense trajectories [Karazija et al., 2024; Huang et al., 2025a]. However, such reliance on long-term tracking makes these methods insensitive to static–dynamic transitions, limiting their online applicability.

Another line of work exploits 3D or 4D geometric priors for motion segmentation. MegaSAM [Li et al., 2025b] learns motion probability maps from estimated camera parameters and depth to separate moving and static content, while RoMo [Goli et al., 2025] combines optical flow with epipolar constraints and iteratively refines motion masks using segmentation priors from SAM [Kirillov et al., 2023]. Building upon feed-forward reconstruction methods [Wang et al., 2024c; Wang et al., 2025b],

MonST3R [Zhang et al., 2025] and DAS3R [Xu et al., 2024] jointly decode moving regions alongside temporally consistent video reconstructions. Easi3R [Chen et al., 2025] reveals that the attention layers of such reconstruction models already encode relative camera-object motion. GeoMotion [He et al., 2026] further exploits the latent 4D features of π^3 [Wang et al., 2026] to directly disentangle object and camera motion in a feed-forward manner.

Moving Object Segmentation (MOS) aims to discover, segment, and track individual moving objects. For videos containing a single moving object, MOS degenerates to foreground–background motion segmentation. Recent works instead focus on the more challenging multi-object scenario, similarly building upon the aforementioned motion cues (i.e. flow, trajectories, and geometric priors).

Without relying on human annotations, one group of methods [Safadoust and Güney, 2023; Lian et al., 2023; Lao et al., 2023] pursues purely unsupervised training via flow reconstruction as a proxy objective. Alternatively, OCLR [Xie et al., 2022] takes optical flow as input and trains object-centric layered representations on synthetic data to discover and track multiple objects under occlusion, while ABR [Xie et al., 2024b] extends it with appearance-based refinement.

With increasingly available annotated masks and strong pre-trained models, Flow-SAM [Xie et al., 2024c] injects optical flow into SAM [Kirillov et al., 2023], either as an input channel or as prompt signals, for moving object segmentation. SegAnyMo [Huang et al., 2025a] fuses long-range trajectory cues with DINO [Caron et al., 2021] features and employs SAM2 [Ravi et al., 2024] for mask densification.

5.6 Conclusion

We presented GMOS, a feed-forward framework that grounds Moving Object Segmentation in 3D space and time, making it robust to camera-induced parallax in space and temporally sensitive to per-frame motion-state changes. GMOS operates on RGB video alone and supports both online and offline inference. To facilitate training and evaluation in this regime, we curated GMOS-2K, a real-world video dataset with per-object temporal motion annotations, and formalised MOS-I, a temporally fine-grained evaluation protocol with three complementary

metrics. GMOS achieves state-of-the-art results on MOS, MOS-I, and UVOS benchmarks, while running roughly three times faster than prior MOS methods.

Acknowledgements. This research is supported by the UK EPSRC Programme Grant VisualAI (EP/T028572/1), a Royal Society Research Professorship RSRP\R\241003, and a Clarendon Scholarship.

Appendices

Appendices for this chapter can be found in the online version of the paper.[1]

Statement of authorship

A statement of authorship for this paper is provided in Appendix A.

[1] https://www.robots.ox.ac.uk/~vgg/research/gmos/

Part II

Video Understanding and Audio Description

Chapter 6

AutoAD-Zero: A Training-Free Framework for Zero-Shot Audio Description

This paper was published at the Asian Conference on Computer Vision (ACCV), 2024.

Abstract

Our objective is to generate Audio Descriptions (ADs) for both movies and TV series in a training-free manner. We use the power of off-the-shelf Visual-Language Models (VLMs) and Large Language Models (LLMs), and develop visual and text prompting strategies for this task. Our contributions are three-fold: (i) We demonstrate that a VLM can successfully name and refer to characters if directly prompted with character information through visual indications without requiring any fine-tuning; (ii) A two-stage process is developed to generate ADs, with the first stage asking the VLM to comprehensively describe the video, followed by a second stage utilising a LLM to summarise dense textual information into one succinct AD sentence; (iii) A new dataset for TV audio description is formulated. Our approach, named AutoAD-Zero, demonstrates outstanding performance (even competitive with some models fine-tuned on ground truth ADs) in AD generation for both movies and TV series, achieving state-of-the-art CRITIC scores.

6.1 Introduction

An Audio Description (AD) soundtrack provides a description of the visual content of a video for the visually impaired. It covers aspects of the story that cannot be inferred from the audio soundtrack, particularly "who" is in the scene and "what" they are doing. Furthermore, in order not to overlap with the dialogue, the AD is usually placed in the gaps between characters speaking. Given the increasing power of Visual-Language Models (VLMs), there has naturally been a growing research interest in automating the production of ADs for movies. Most recent approaches have either fine-tuned components of an open-source pre-trained large-scale VLM, such as the visual-text bridge, for the AD task [Han et al., 2023b; Han et al., 2023a; Han et al., 2024; Wang et al., 2025a; Srivastava et al., 2023; Raajesh et al., 2024], or have used very powerful propriety models such as GPT-4/GPT-4v in a zero-shot way, without fine-tuning [Zhang et al., 2024a; Lin et al., 2023; Chu et al., 2024]. Both approaches have limitations: there is insufficient data to fine-tune these large models for the AD task – to enable the model to distill out the essential information for AD and ignore the rest, and current zero-shot techniques fail due to a lack of knowledge of who the characters are, a core requirement for

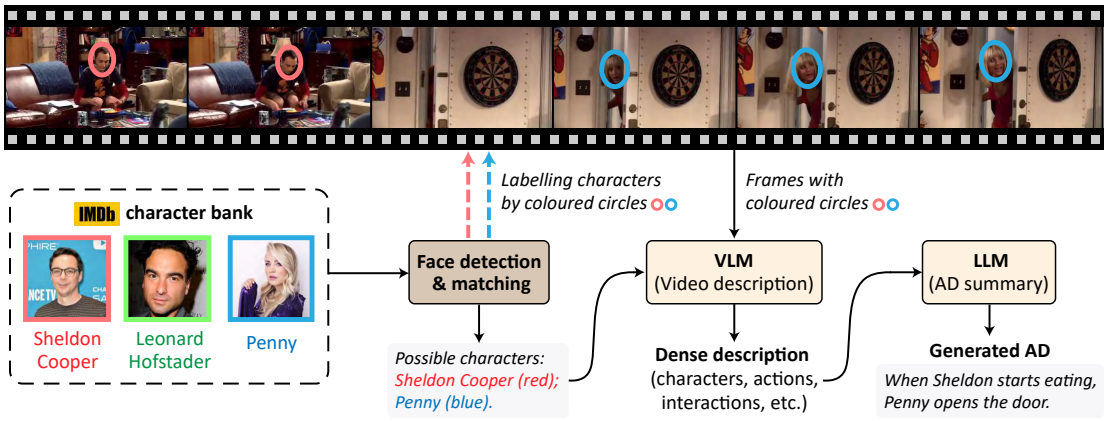


Figure 6.1: **A training-free framework for zero-shot AD generation.** AutoAD-Zero features a two-stage process, where a VLM initially generates a comprehensive video description from multiple aspects, followed by an LLM-based AD summary in the second stage. To incorporate character information into this framework, character faces in the input video are matched with those in an external character bank and labelled with coloured circles. The corresponding character names and colour codes are then provided as text prompts to the VLM.

AD.

In this paper, we show that it is possible to adopt pre-trained models with novel visual and text prompting. Somewhat surprisingly, this zero-shot, training-free method reaches competitive performance with all previous AD approaches, and even gets state-of-the-art in some metrics. To achieve this we explore two ideas, as demonstrated in Figure 6.1: first, use a *visual prompt* on the video frames so that we can refer to characters by name when text prompting the VLM. We achieve this by "circling" the characters in the frame and color coding for each identity. This "circle" prompting [Shtedritski et al., 2023] enables the VLM both to identify the characters in the clip, and also to refer to them by name when describing their actions, interactions, and locations. The second idea is *not* to predict the AD directly, but instead to take advantage of the strength of Large Language Models (LLMs) at providing summaries. We introduce a two-stage approach to generating the AD: in the first stage, a detailed description of "who is doing what" is generated by prompting the VLM with questions about the visually tagged characters; in the second stage, an LLM is used to summarise the detailed description into the style and desired length for the AD. This two-stage approach has multiple advantages – it forces the AD to be more visually grounded; it enables the information in the AD to be prioritised for the temporal interval available; it enables the style of AD to be tailored for particular groups, such as being age appropriate [Salewski et al.,

2024] or a simpler form of the language (for non-native speakers).

The great advantage of our training-free approach, is that the method can be directly plugged into new VLMs as they are released (which is very often at the moment) to benefit from improvements in their descriptive power. It can also be plugged into propriety models because no training is required.

We also investigate generating AD automatically for TV series, such as *"Friends"* and *"The Big Bang Theory"*. This domain differs from movies in a number of aspects: a TV series can often have a cast of characters that is persistent over multiple episodes; and due to shorter episodes, the series often have denser dialogues than films. In older series, the cinematography can also differ, with more close-ups and mid-shots in TV than in movies, whereas there may be more long shots in movies. To this end, we introduce a new dataset of TV series (TV-AD) with ground truth aligned AD annotations.

In summary, we make the following contributions: (i) We devise a visual-textual prompting mechanism to incorporate character information into the VLM in a training-free manner; (ii) We propose a two-stage AD generation process, where a comprehensive character-aware video description is first generated by a VLM, followed by an LLM-based summary stage to obtain the final AD; (iii) We formulate a new TV-AD dataset and investigate the nature of TV ADs in comparison to their movie counterparts; (iv) The proposed AutoAD-Zero model is evaluated on multiple movie/TV AD datasets, including MAD-Eval [Han et al., 2023b], CMD-AD [Han et al., 2024], and TV-AD, significantly outperforming existing training-free AD generation methods. When compared to models explicitly fine-tuned on ground truth AD annotations, AutoAD-Zero demonstrates competitive performance and achieves state-of-the-art results on the CRITIC scores.

6.2 Related work

Previous AD and video captioning models. With the power of large vision and language models, rapid progress has been made in video captioning. The dense video captioning task focuses on short video clips [Lin et al., 2022; Luo et al., 2020; Seo et al., 2022], and other variants summarise longer video segments that span a few minutes [Lu and Grauman, 2013; Zhang et al., 2016]. The task

of Audio Description (AD) generation is closer to dense video captioning with some differences, where the model aims to describe visual elements in the movie or TV densely over time, in a story-telling manner. Unlike video captioning, AD generation requires a highly summarised output with strict formatting, coherent context [Han et al., 2023b; Wang et al., 2025a; Ye et al., 2024], and character information [Han et al., 2023a; Raajesh et al., 2024]. Notably in [Han et al., 2023a], a separate character recognition module is trained to feed character identities to the VLM, which is shown to be critical compared with an alternate method that applies GPT-4 model for AD generation [Zhang et al., 2024a]. Character recognition in movies or TV typically can be achieved by face matching with portrait exemplars [Huang et al., 2018] or joint audio-visual matching that includes voice [Korbar et al., 2024]. The emotion of the characters is also explored for movie descriptions [Srivastava et al., 2023].

On the data acquisition side, due to the copyright constraints of movies and TV series, early works share short video clips [Rohrbach et al., 2015] or pre-computed visual features [Soldan et al., 2022], limiting the scale and power of AD generation models. [Han et al., 2024] proposes to align YouTube videos from [Bain et al., 2020] with AD annotations to acquire longer movie clips with pixels. A recent work [Ghermi et al., 2024] collects datasets with amateur-made short movies (13 mins on average) that are publicly available and easier to share.

Prompting for LLMs and VLMs. Prompting refers to designing instructions that guide the pre-trained models to generate desired outputs. One stream of works learns vector prompts with targeted training data, on both LLMs [Li and Liang, 2021; Lester et al., 2021] and VLMs [Zhou et al., 2022b; Zhou et al., 2022a; Ju et al., 2022]. Another stream designs text-only prompts which is more compatible with black-box pre-trained models prohibiting any model modifications. For example, it is found that adding a text prompt "let's think step by step" improves the performance of GPT3 on various tasks [Kojima et al., 2022]. Using text prompts with chain-of-thought examples can activate the reasoning in LLM's answers [Wei et al., 2022].

Our work is closer to "visual prompting", that provides supplementary *visual* information to the VLMs without any fine-tuning [Jia et al., 2022; Bar et al., 2022].

In [Shtedritski et al., 2023], the authors explore visual prompting for large-scale VLMs like CLIP, discovering that drawing a red circle around an object can effectively direct the model's attention, leading to strong performance in zero-shot referring expressions comprehension, and keypoint localization tasks. In [Cai et al., 2024a], the authors introduce a multimodal model that decodes arbitrary visual prompts, *i.e.* visual markers overlayed on the RGB image, allowing intuitive interaction with images through natural cues such as "red bounding box" or "pointed arrow". In [Yang et al., 2023b], the authors introduce Set-of-Mark (SoM), a visual prompting method that enhances the visual grounding capabilities of large multimodal models like GPT-4v. By using interactive segmentation models to partition images and overlaying them with marks, SoM enables models to answer questions requiring visual grounding.

Training-free VLM adaptation. With the strong performance and huge training cost of large-scale pretrained VLMs, many works attempt to adapt VLMs to specific tasks in a training-free manner. Yan *et al.* [Yang et al., 2023c] showcases GPT-4v on a wide range of multi-modal applications, such as recognizing celebrities and famous landmarks, and understanding code snippets from screenshots. State-of-the-art VLMs are also adapted to various medical applications [Wu et al., 2023a; Yan et al., 2023], but these works find the VLMs still lack expertise in medical knowledge.

Our work is closely related to MM-Narrator [Zhang et al., 2024a] and LLM-AD [Chu et al., 2024], where MM-Narrator leverages GPT-4/GPT-4v with multimodal in-context learning to generate the audio descriptions for long movies, in an auto-regressive manner with memories. A concurrent work, LLM-AD, applies visual prompting to GPT-4v by overlaying character names onto movie frames. In contrast, we introduce a visual-textual prompting approach that provides character names as a part of the text prompts with colour-coded circles as fine-grained visual indicators, leading to superior performance.

6.3 Prompting VLM with character information

To incorporate character information into ADs, prior works [Han et al., 2023a; Han et al., 2024; Wang et al., 2025a] explicitly tune the VLM to adapt to character

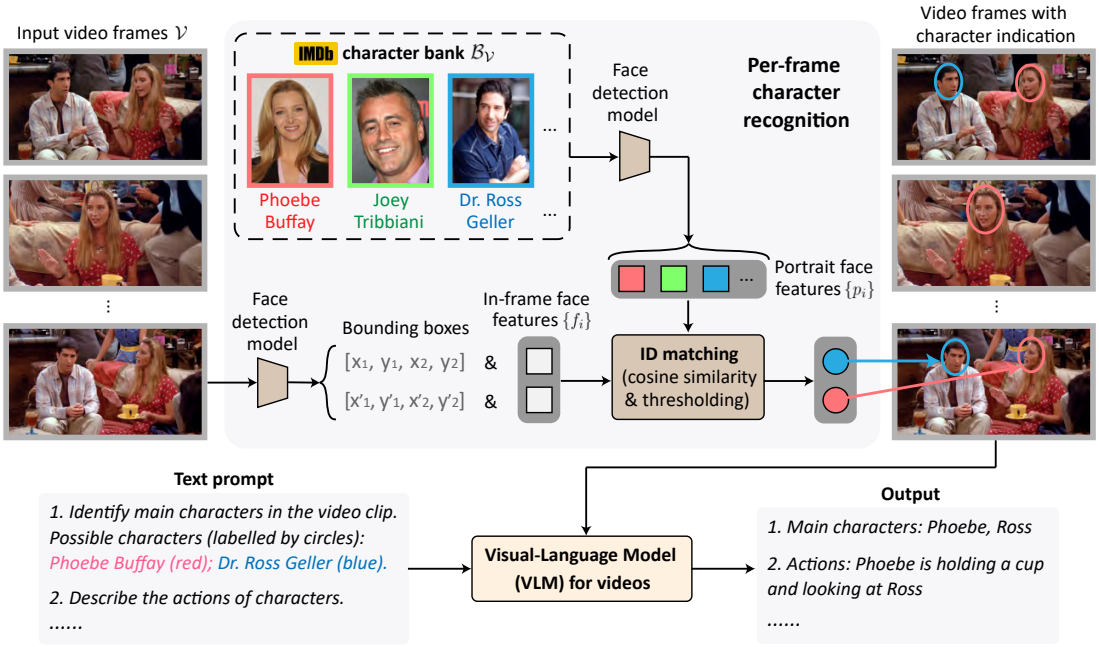


Figure 6.2: **Character recognition and VLM prompting**. An off-the-shelf face detection model is employed to obtain bounding boxes and face features in video frames. These "in-frame face features" are then matched with "portrait face features" extracted from character profile images, which determines the identities in the video. To prompt the VLM with character information, character faces are labelled by coloured circles, with corresponding names and colour codes provided in the text prompt.

information. We instead explore a training-free route that involves two steps, as detailed in Figure 6.2: (i) character recognition based on an off-the-shelf face detection model, then (ii) prompt VLM with visual character indications.

6.3.1 Character recognition by face detection

Given a video clip $\mathcal{V}$ from a movie or a TV series episode, we query the corresponding cast list from the database. The video-specific character bank can be then obtained as $\mathcal{B}_\mathcal{V} = \{[\mathsf{char}_i, \mathcal{P}_i]\}_{i=1}^{C}$ where there are C characters, char_i indicates the character name, and $\mathcal{P}_i$ denotes the profile portrait image of the corresponding actor. An exemplary character bank shown in Figure 6.2 is:

$$\{[\textit{"Phoebe Buffay"}, \mathcal{P}_1], [\textit{"Joey Tribbiani"}, \mathcal{P}_2], [\textit{"Dr. Ross Geller"}, \mathcal{P}_3], ...\}$$

We then formulate a pipeline to detect faces in the video frame and match them with those in the character bank. This process is *independently* conducted for each frame.

Face feature extraction. Given a video frame $\mathcal{I}$ in the clip $\mathcal{V}$, an off-the-shelf face detection model Φ is employed to identify faces:

$$\{\texttt{Bbox}_j, f_j\}_{j=1}^{N} = \Phi(\mathcal{I}) \tag{6.1}$$

where N faces are detected in a frame, with the feature embedding of each face f_j extracted, together with a predicted bounding box coordinate $\texttt{Bbox}_j$. We refer to the set of face embeddings as *in-frame face features* $\{f_i\}_{i=1}^{N}$.

Similarly, a face embedding is extracted from each portrait image $\mathcal{P}_i$ in the character bank, together forming a set of *portrait face features* $\{p_i\}_{i=1}^{C}$.

Face ID matching. To match the faces within video frames with those in portrait images, we evaluate the cosine similarities $\mathcal{A}_{ij}$ between two sets of features and identify the best-matched portrait face for each in-frame face feature. Formally,

$$\texttt{ID}_j = \max_i(\mathcal{A}_{ij}), \quad \text{where } \mathcal{A}_{ij} = \frac{p_i \cdot f_j}{|p_i||f_j|} \tag{6.2}$$

where $\texttt{ID}_j$ denotes the predicted face ID in the character bank. As a result, the predicted bounding boxes within the video frame could be matched with a character name, forming a triplet set $\{[\texttt{char}_{\texttt{ID}_j}, \texttt{Bbox}_j, \mathcal{A}_{\texttt{ID}_j j}]\}_{j=1}^{N}$, where $\mathcal{A}_{\texttt{ID}_j j}$ is the matching score.

We filter out triplets based on the matching scores and the following rules:

- If $\mathcal{A}_{\texttt{ID}_j j} < \epsilon$, where ϵ denotes a threshold, this indicates that the face matching is not confident, possibly owing to poor video quality, motion blur, non-important characters' faces, *etc.* We therefore remove those faces (triplets) from the set.
- If the predicted IDs for multiple faces are the same within a frame, we keep the one with the highest matching score $\mathcal{A}_{\texttt{ID}_j j}$.

6.3.2 Visual prompting VLM by circling character faces

In the training-free design, one key challenge lies on the strategy to prompt VLM for understanding the association between visible characters and their names. Recent work [Shtedritski et al., 2023] has revealed that drawing a red circle (ellipse)

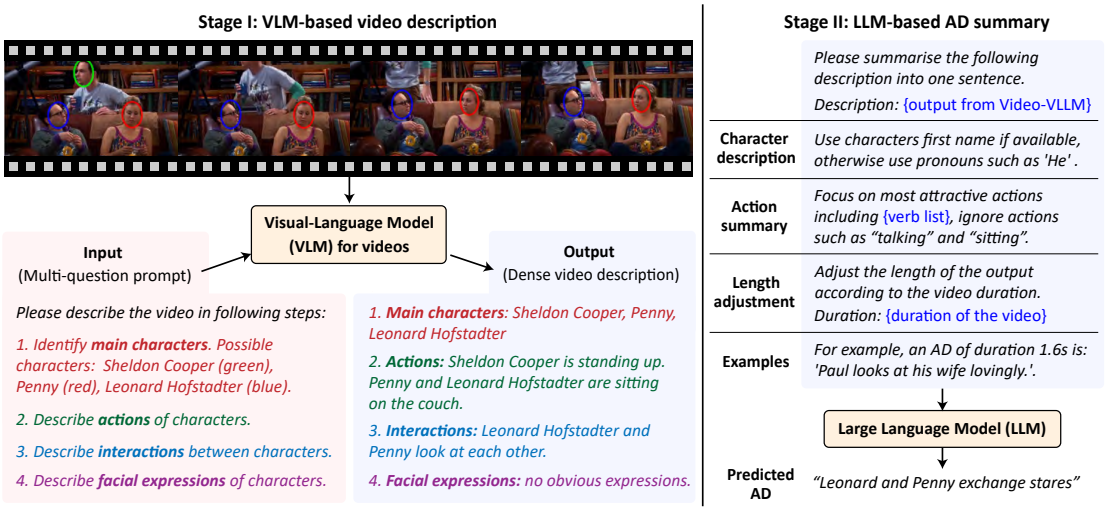


Figure 6.3: **Two-stage training-free AD generation**. The first stage adopts a VLM to produce a comprehensive video description, covering aspects including main characters, actions, interactions, and facial expressions. The second stage uses an LLM to summarise the video into a single AD sentence, extracting the most relevant character and action information, and adjusting the content and style according to specific rules.

could effectively direct the CLIP's attention to a local region while maintaining global information understanding. Inspired by this, and given the fact that CLIP is popularly adopted as the visual encoder for recent VLMs, we overlay coloured circles around character faces in the frame as direct visual indications.

As shown in Figure 6.2, we assign a unique colour code for each character, and transform the corresponding bounding box coordinates into circles that are overlaid on the raw video frames. These frames with coloured circles are then directly taken as input to a VLM, while the character names and corresponding colours are listed in the input text prompts, *e.g. "Phoebe Buffay (red)"*. By leveraging coloured circles as a bridge, it can be observed that the VLM is capable of associating character information between text and video and using it to accomplish other tasks, such as action description.

6.4 Two-stage training-free AD generation

In this section, we introduce a training-free method to generate AD sentences from video frame inputs. This approach consists of two stages: (i) comprehensive character-aware video description by a VLM, and (ii) AD summary by an LLM based on the outputs of the first stage.

6.4.1 Stage I: VLM-based video description

The VLM processes video frames along with a list of text-form questions (*e.g.* who are the characters? and what are they doing?) to generate a comprehensive video description. Compared to directly outputting a one-sentence AD, this process adopts the Chain-of-Thought (CoT) approach, guiding the model to systemically understand the video.

Figure 6.3 (left) provides a simplified version of the input text prompt, where we begin by asking the model to **identify the main characters** ("who") in the video (point 1). Following the process described in Section 6.3.2, we label individual characters with colored circles and refer to them in the text by specifying the colors, such as *"Sheldon Cooper (red)"*. We found that by first asking the model for character names, it effectively remembers and naturally uses these names for describing actions.

We then proceed with questions about "what", that consists of three major directions, asking the model to:

- Describe the **actions** of characters (point 2). This includes characters' states (*e.g.* sitting or standing), gestures, movements (*e.g.* walking), and interactions with objects (*e.g.* pouring a drink).

- Describe the **interactions** between characters (point 3). Despite potentially overlapping with point 2, this point emphasises the importance of focusing globally on multiple characters and understanding their interactions, including physical contacts (*e.g.* hugging) and distant communications (*e.g.* looking at others).

- Describe the **facial expressions** of characters (point 4). This instruction encourages the model to focus on facial details that may convey the characters' emotions. Furthermore, directing the model to zoom in on facial regions enhances the accuracy of character recognition.

Additional directions on describing the environment and character appearances are also investigated, but no significant improvements are observed (see ablations in Section 6.6.5).

These points together form a multi-question prompt that is fed into VLM along

with the raw video frames. We ensure that the model answers are in the same order as the questions (the template is shown in Figure 6.3), resulting in a dense video description. One can also ask the VLM model to summarise the description into a one-sentence AD at the end. However, we found it challenging to enforce its adherence to the AD format while simultaneously generating high-quality descriptions.

6.4.2 Stage II: LLM-based AD summary

As demonstrated in Figure 6.3 (right), the second stage aims to summarise the video description obtained from the first stage into a single AD sentence, while adjusting the outputs according to specific rules. Note that, we use an LLM for this task that takes text-form inputs only (no video inputs).

Specifically, the text prompt starts with *"Please summarise the following description into one sentence."*, followed by the provision of first-stage outputs. The remaining instructions contain four main components:

- **Character description**, which formalises the use of character names (*i.e.* only the first name) and pronouns, rather than descriptions such as "a man".
- **Action summary**, which asks the LLM to extract the most important characters and their actions. To guide this process, we extract top-k ($k = 15$) most frequent verbs in *training splits* of AD datasets to form a verb list. Additionally, actions related to static states (*e.g.* sitting) and talking-related actions are considered non-informative for ADs, as they can be inferred from previous contexts or dialogues.
- **Length adjustment**, which provides the duration of the time windows for ADs as hints that indicate how much detail should be included in the output.
- **Examples**, which offer references for the LLM regarding AD styles and lengths.

After combining all these factors into the text prompt, the LLM outputs a single sentence as the summarised AD.

6.5 TV-AD dataset

In this section, we introduce a new dataset, namely TV-AD, that contains ground truth audio descriptions for TV series. Section 6.5.1 specifies the dataset details and provides a comparison with a movie AD dataset. Section 6.5.2 elaborates on how the dataset is formulated by aligning AD annotations with TV episodes.

6.5.1 Dataset details

Table 6.1 provides statistics of the TV-AD dataset, which features episodes across multiple TV series including *"The Big Bang Theory"*, *"Friends"*, *"Frasier"*, *"Seinfeld"*, *etc.* (The full list is available in the Appendix. We further divide the dataset into training (TV-AD-Train, $\sim$ 31k ADs) and evaluation splits (TV-AD-Eval, $\sim$ 3k ADs), ensuring that the TV series do not overlap between the two splits. The evaluation split contains AD annotations for TV videos that are publicly available (from TVQA [Lei et al., 2018]) and will be publicly released.

Split	TV series	Seasons	Episodes	ADs	Total duration
Train	11	18	326	31030	214h
Eval	2	5	100	2983	37h
Total	13	23	426	34013	251h

Table 6.1: **Statistics of the TV-AD dataset.** Train and Eval splits consist of different TV series.

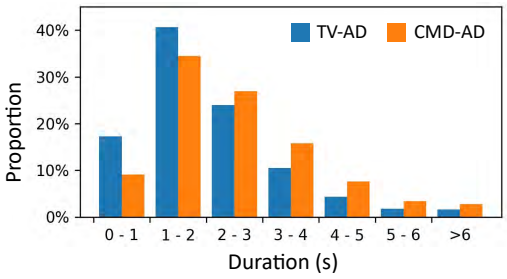


Figure 6.4: **Comparison of AD duration** between TV-AD and CMD-AD.

Comparison with CMD-AD. CMD-AD [Han et al., 2024] (where "CMD" stands for Condensed Movie Dataset [Bain et al., 2020]) is a movie AD dataset that contains $\sim$ 100k ADs from $\sim$ 1.4k movies. The duration of ADs between TV-AD and CMD-AD are compared in Figure 6.4. Movie ADs (mean duration = 2.51s) are shown to be generally longer, while TV ADs (mean duration = 2.07s) consist of a large proportion within $1-2$s. This could be due to the shorter time of each TV episode compared to movies, as well as fast-paced plots and dialogues in TV series, resulting in more compressed audio descriptions.

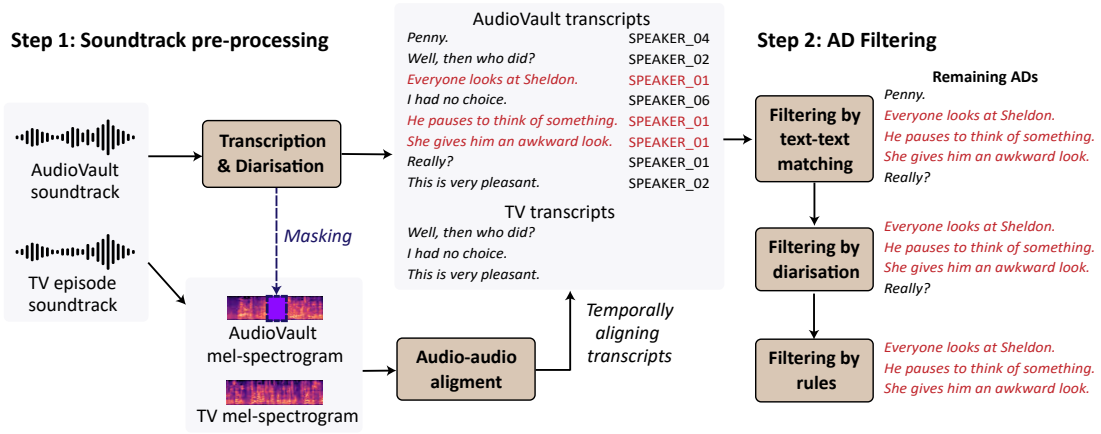


Figure 6.5: **Dataset formulation**. The aim is to convert AD annotations from AudioVault into text form and align them with TV episodes. The main pipeline consists of two steps: (i) The soundtrack pre-processing step, which aligns the AudioVault and TV timestamps via audio-audio matching and transcribes both sound sources into transcripts; (ii) The AD filtering step, which retrieves text-form ADs from the transcripts and performs further cleaning.

6.5.2 Dataset formulation: Aligning ADs with TV series

The AudioVault website[1] (as also used by [Soldan et al., 2022; Han et al., 2023b; Han et al., 2024]) provides human-annotated audio descriptions for thousands of movies and TV series in the form of audio files, with the spoken ADs merged with the original TV soundtracks. To formulate the TV-AD dataset, we (i) temporally align the AudioVault soundtracks with TV episodes; (ii) extract human-annotated ADs from AudioVault soundtracks and transcribe them into text forms.

As noted by [Han et al., 2024], the alignments between movies and AudioVault soundtracks are non-trivial owing to (i) non-identical soundtracks; (ii) different recording speeds (*e.g.* NTSC 29.97 fps *vs.* RAL 25 fps). Furthermore, in TV series, ADs are typically shorter than those in movies, and there is more frequent mixing between spoken ADs and sound effects (*e.g.* laughing). Both factors pose greater challenges for accurate AD extraction from the soundtracks.

To solve these challenges, we propose an improved pipeline that starts with AudioVault and TV soundtracks and consists of two major steps, namely soundtrack pre-processing and AD filtering, as demonstrated in Figure 6.5.

Step 1: Soundtrack preprocessing. In this step, we convert both AudioVault and TV soundtracks into text-form transcripts and temporally align them. In

[1] https://audiovault.net/

detail, we first use WhisperX [Bain et al., 2023] to transcribe both soundtracks into text forms along with the corresponding timestamps. Additionally, we apply a diarisation module to identify speakers for the AudioVault transcripts. Formally, the resultant transcripts are denoted as $\mathcal{S}_{\text{AV}} = \{(s_1, t_1, c_1), \dots (s_n, t_n, c_n)\}$ and $\mathcal{S}_{\text{TV}} = \{(s'_1, t'_1), \dots (s'_m, t'_m)\}$, where s_i represents transcribed sentences, along with the time stamps t_i, and c_i indicates the speaker IDs estimated by the diarisation module.

These transcribed sentences are then temporally matched in terms of the timestamps by an audio-audio alignment. This process follows the same design as proposed in [Han et al., 2024]. (Note, we found audio-audio alignment alone is already precise enough, therefore skipping the first text-text alignment step in their original proposal). Specifically, the soundtracks are transformed into mel-spectrograms ($\mathcal{M}_{\text{AV}}$ and $\mathcal{M}_{\text{TV}}$) that capture the low-level audio representations. Based on the diarisation results, we mask out the regions in the AudioVault mel-spectrogram that overlap with AD timestamps. We then evaluate the correlation map $\mathcal{M}_{\text{AV}}$ and $\mathcal{M}_{\text{TV}}$, and fit it with a linear function f using RANSAC. This function can be used to convert timestamps from AudioVault to TV soundtracks, *i.e.* f:$\{t_{\text{AV}} \rightarrow t_{\text{TV}}\}$, therefore aligning two modalities.

Step 2: AD filtering. As previously noted, simple diarisation on AudioVault soundtracks for TV series can not accurately determine AD speakers with non-AD speakers, and therefore cannot be directly applied to extract ADs. We instead formulate a multi-step filtering mechanism that identifies ADs from ASR transcripts that contain both AD and non-AD speakers.

First, we compare the pairs of AudioVault sentences s_t (either AD or non-AD speakers) and TV sentences (just non-AD speakers) $s'_{t\pm\Delta t}$, with Δt denoting the maximum difference between their timestamp centers. If the word error rate (WER) between a pair of sentences is below a threshold ϵ', *i.e.* $\text{WER}(s_t, s'_{t\pm\Delta t}) < \epsilon'$, we consider the two sentences are matched (therefore correspond to a non-AD speaker), and remove the sentence s_t from the AudioVault transcripts. As exemplified in Figure 6.5, this filtering by text-text-matching effectively removes a majority of non-AD contents, though still affected by transcription and alignment errors.

Second, we perform diarisation-based filtering that first determines the AD speaker by majority voting, followed by the removal of all sentences from non-AD speakers. After filtering, a few non-AD sentences may remain, albeit marginally, due to potential diarisation errors.

Finally, we conduct rule-based heuristic filtering, which discards sentences containing certain symbols (*e.g. "?"* and *"!"*), pronouns (*e.g. "I"* and *"We"*), and phrases (*e.g. "Let's go."*), that never appear in ADs. Further details are discussed in the Appendix.

6.6 Experiments

6.6.1 Datasets

We evaluate the model performance using several datasets, focusing on character recognition (as an intermediate task), movie AD generation, and TV AD generation.

Character recognition dataset. For character recognition evaluation, we follow [Han et al., 2023a] and adopt a 4-movie subset of **MovieNet** [Huang et al., 2020], where ground-truth character names are provided in each keyframe (shot) within the movie.

Movie AD datasets. We report the performance on two movie AD datasets, namely **CMD-AD** [Han et al., 2024] and **MAD-Eval** [Han et al., 2023b]. The former is generated by aligning AD data with video clips from CMD (Condensed Movie Dataset) [Bain et al., 2020], resulting in 101k ground truth ADs within $\sim$ 1.4k movies. In this work, we adopt its evaluation set (CMD-AD-Eval), which consists of $\sim$ 7k ADs for 100 movies. On the other hand, MAD-Eval is a 10-movie subset from LSMDC [Rohrbach et al., 2017], featuring $\sim$ 6.5k ground truth AD annotations.

TV AD dataset. We adopt the new **TV-AD** dataset (introduced in Section 6.5) and measure the performance on its evaluation split, which includes around 3k ADs in 100 TV episodes.

6.6.2 Evaluation metrics

To measure the **character recognition performance** in the intermediate stage, we compare the predicted and ground truth character name lists and compute the IoU, precision, and recall of the prediction.

For AD evaluation, we monitor the performance on three metrics, namely CIDEr, CRITIC, and LLM-AD-Eval.

CIDEr [Vedantam et al., 2015] is a common metric that evaluates the quality of text descriptions. It measures the relevance and uniqueness of words in candidate descriptions based on a Term Frequency-Inverse Document Frequency (TF-IDF) weighting scheme. However, this word-matching-based metric has its limitations, as ADs with the same semantics can be presented in various ways. Since many recent works have adopted CIDEr scores, we report the performance for comparison purposes.

CRITIC [Han et al., 2024] measures the accuracy of character identification in the predicted ADs. Specifically, a co-referencing model is employed to replace ambiguous pronouns such as *"He"* or *"She"* in ADs with official names from the character banks. Then, two name sets from each pair of predicted and GT ADs are compared, and the IoU is calculated to give a CRITIC score.

LLM-AD-Eval [Han et al., 2024] utilises LLMs to judge the quality of generated ADs by scoring them between 1 (lowest) and 5 (highest). The evaluation focuses on the matching between generated and ground truth ADs in terms of human actions, objects, and interactions. We directly adopt the LLM prompt provided in the original paper and use open-source models LLaMA2-7B [Touvron et al., 2023] and LLaMA3-8B [AI@Meta, 2024] for the evaluation.

6.6.3 Implementation details

Character recognition details. To formulate the character bank for each movie or TV episode, we extract top-10 characters from the IMDb website[2] along with their corresponding actor profile images. For face detection and recognition,

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

Table 6.2: **Character recognition results** reported on 4 MovieNet movies. Note, AutoAD-III [Han et al., 2024] shares the same character recognition module as AutoAD-II [Han et al., 2023a].

Methods	IoU	Precision	Recall
AutoAD-II [Han et al., 2023a]	70.8	75.8	85.6
AutoAD-Zero **(Ours)**	**75.8**	**79.3**	**85.8**

we adopt the InsightFace package[3], which is developed based on ArcFace [Deng et al., 2019] and RetinaFace [Deng et al., 2020]. The threshold for face matching (detailed in 6.3.1) is set to $\epsilon = 0.2$.

AD prediction details. For stage I, we use VideoLLaMA2-7B [Cheng et al., 2024b] as the Video-LLM model, which takes in 8 video frames. At stage II, we adopt LLaMA3-8B [AI@Meta, 2024] for AD summary. The exact text prompts are available in the Appendix.

6.6.4 Character recognition results

In Table 6.2, we compare the character recognition module with the one adopted in AutoAD-II [Han et al., 2023a], which trains a transformer-based model with CLIP feature inputs to predict the probability of each character. In contrast, our method first employs an off-the-shelf face detection model with a face-matching mechanism to obtain raw character predictions. These predictions are then encoded into coloured circles to visually prompt the Video-LLM, as detailed in Figure 6.2.

Following [Han et al., 2023a], we report the character recognition performance on 4 MAD-Eval movies with keyframe character annotations from MovieNet. Note that our reported performance is measured on the final Video-LLM outputs. As demonstrated in Table 6.2, AutoAD-Zero achieves higher IoU, precision, and recall in character recognition, indicating that our predicted character names are more accurate and reliable compared to those predicted by AutoAD-II.

6.6.5 Audio Description predictions

In this section, we compare our training-free method with prior models and provide qualitative examples. We also present a detailed ablation analysis in the Appendix.

[3] https://github.com/deepinsight/insightface

Table 6.3: **Comparison with other methods on CMD-AD, MAD-Eval, and TV-AD.** "*" denotes the results of the AutoAD-III model that we reproduced by training it on the TV-AD training split. The LLM-AD-eval scores are evaluated using LLaMA2-7B (left) and LLaMA3-8B (right).

Models	Training -free	CMD-AD			MAD-Eval	TV-AD		
		CIDEr	CRITIC	LLM-AD-eval	CIDEr	CIDEr	CRITIC	LLM-AD-eval
AutoAD-I [Han et al., 2023b]	✗	–	–	–	14.3	–	–	–
AutoAD-II [Han et al., 2023a]	✗	13.5	8.2	–	19.2	–	–	–
AutoAD-III [Han et al., 2024]	✗	**25.0**	32.7	**2.89** | **2.01**	24.0	**26.1***	15.9*	2.78* | 1.99*
Uni-AD [Wang et al., 2025a]	✗	–	–	–	**28.2**	–	–	–
MM-Narrator (GPT-4) [Zhang et al., 2024a]	✓	–	–	–	13.9	–	–	–
MM-Narrator (GPT-4v) [Zhang et al., 2024a]	✓	–	–	–	9.8	–	–	–
LLM-AD [Chu et al., 2024]	✓	–	–	–	20.5	–	–	–
AutoAD-Zero **(Ours)**	✓	17.7	**43.7**	2.83 | 1.96	22.4	22.6	**27.6**	**2.94** | **2.00**

Quantitative comparison. Overall, AD generation methods are majorly divided into two classes: (i) models explicitly trained on ground truth movie or TV AD annotations, including AutoAD-I to III [Han et al., 2023b; Han et al., 2023a; Han et al., 2024] and Uni-AD [Wang et al., 2025a]; and (ii) *training-free* approaches that perform *zero-shot* AD generation *without* finetuning on AD datasets, including MM-Narrator [Zhang et al., 2024a], LLM-AD [Chu et al., 2024], and AutoAD-Zero (our method). The latter methods are more flexible and extendable in terms of performance, as they could be directly integrated with more advanced models. However, it is challenging for training-free models to outperform those fine-tuned on human-annotated ADs that directly benefit from domain-specific knowledge.

We focus on evaluating model performance on CMD-AD and TV-AD. Since several prior works report their CIDEr scores on MAD-Eval, we also include it in our comparison. As demonstrated in Table 6.3, among the training-free methods, AutoAD-Zero surpasses MM-Narrator (both GPT-4 and GPT-4V versions) on the MAD-Eval evaluation. When compared to models fine-tuned on AD datasets, AutoAD-Zero delivers competitive results on CIDEr and LLM-AD-eval metrics and achieves state-of-the-art performance on CRITIC score, indicating our superior character identification accuracy.

Qualitative visualisation. Figure 6.6 provides several examples on CMD-AD and TV-AD. By prompting VLM with character information (*i.e.* coloured circles in images and names in text prompts), AutoAD-Zero is capable of associating characters with their actions. In the bottom left example, even though faces are sparsely labelled (as characters may turn away from the camera), the VLM suc-

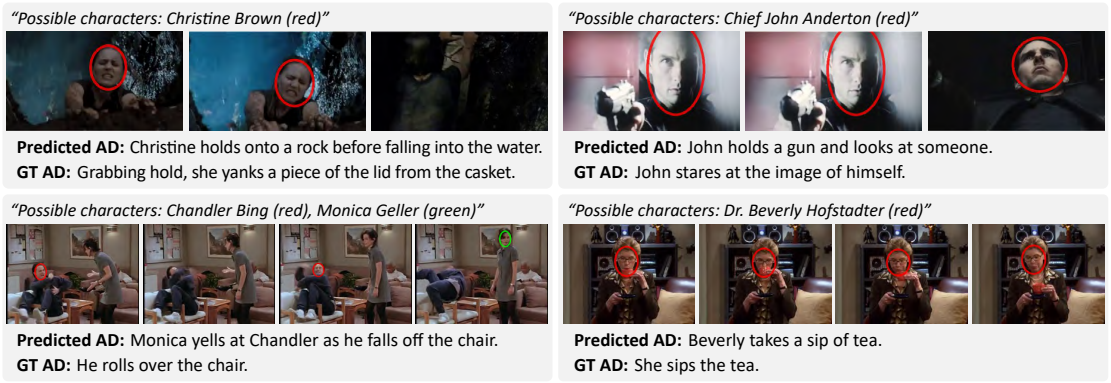


Figure 6.6: **Qualitative visualisations** for CMD-AD (top) and TV-AD (bottom). The faces are labelled by coloured circles, with corresponding names and colour codes listed above. The examples are from *"Drag Me to Hell"* (top left), *"Minority Report"* (top right), *"Friends"* (bottom left), and *"The Big Bang Theory"* (bottom right).

cessfully links the character across the frames (tracking) and identifies individual actions (*e.g. Monica - yelling, Chandler - falling off the chair*).

6.7 Discussion and extensions

In this paper, we propose AutoAD-Zero, a zero-shot AD generation framework featuring visual prompting of character information and a two-stage description-summarisation process. The two-stage design offers notable advantages: it brings the extensibility to use more advanced VLMs and LLMs in the future, and the flexibility to tailor the second-stage LLM to fit specific styles.

However, the visual and text prompt design is still based on a heuristic trial-and-error approach which could be optimised. Also, errors accumulate between stages – if the first-stage VLM fails to describe an action, the second-stage LLM cannot summarise it. Moreover, recent VLMs and LLMs may have already encountered popular characters and scenes during training, potentially undermining the zero-shot nature of the task. In the future, data leakage could be alleviated by adopting more recent or amateur film data [Ghermi et al., 2024] for evaluation.

Potential extensions of this work include: (i) Enhancing the VLM with additional knowledge from "experts" or "specialists" (*e.g.* location from a scene recognition model); (ii) Expanding the second stage by providing the LLM with more context, including previous ADs, dialogues, and plot summaries.

Acknowledgments.

This research is supported by the UK EPSRC Programme Grant Visual AI (EP/T028572/1), a Clarendon Scholarship, a Royal Society Research Professorship RP\R1\191132, and ANR-21-CE23-0003-01 CorVis.

Appendices

Appendices for this chapter can be found in the online version of the paper.[4]

Statement of authorship

A statement of authorship for this paper is provided in Appendix A.

[4] https://www.robots.ox.ac.uk/~vgg/research/autoad-zero/

Chapter 7

Shot-By-Shot: Film-Grammar-Aware Training-Free Audio Description Generation

This paper was published at the International Conference on Computer Vision (ICCV), 2025.

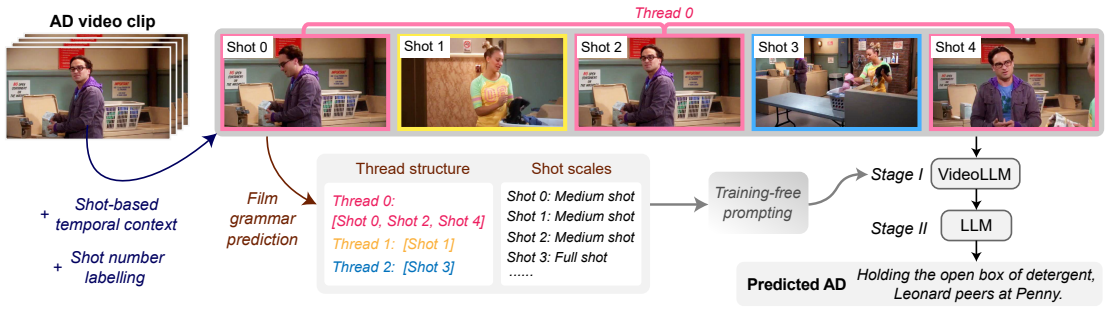


Figure 7.1: **Overview of our training-free Audio Description (AD) generation framework.** The input video clip (corresponding to the AD interval) is extended to include adjacent shots, providing richer temporal visual context. Each video frame is labelled with its corresponding shot number. The extended clip then undergoes a film grammar prediction process, where the thread structure and shot scales are estimated. The AD is generated in two stages: Stage I utilises the predicted cinematic information as training-free prompt guidance to produce dense video descriptions. Stage II then employs an LLM to generate a summarised AD output.

Abstract

Our objective is the automatic generation of Audio Descriptions (ADs) for edited video material, such as movies and TV series. To achieve this, we propose a two-stage framework that leverages "shots" as the fundamental units of video understanding. This includes extending temporal context to neighbouring shots and incorporating film grammar devices, such as shot scales and thread structures, to guide AD generation. Our method is compatible with both open-source and proprietary Visual-Language Models (VLMs), integrating expert knowledge from add-on modules without requiring additional training of the VLMs. We achieve state-of-the-art performance among all prior training-free approaches and even surpass fine-tuned methods on several benchmarks. To evaluate the quality of predicted ADs, we introduce a new evaluation measure – an action score – specifically targeted to assessing this important aspect of AD. Additionally, we propose a novel evaluation protocol that treats automatic frameworks as AD generation assistants and asks them to generate multiple candidate ADs for selection.

7.1 Introduction

In movies and TV series, Audio Descriptions (ADs) are narrations provided for the visually impaired, conveying visual information to complement the original soundtrack. Their purpose is to ensure a continuous and coherent narrative flow,

enabling audiences to follow the plot effectively. Unlike video captions, ADs are constrained by length, prioritising the most visually salient and story-centric information, such as character dynamics and significant objects, whilst omitting redundant details like background figures or unchanging locations. Additionally, ADs are typically produced by professional narrators in a specific style and format, ensuring coherence while not interfering with the original audio.

With the advent of Visual-Language Models (VLMs), there has been a growth of interest in automatically generating ADs for both movies [Ye et al., 2024; Zhang et al., 2024a; Han et al., 2023b; Han et al., 2023a; Han et al., 2024; Wang et al., 2025a; Lin et al., 2024; Chu et al., 2024] and TV material [Xie et al., 2024a; Fang et al., 2024]. However, as anyone who has ever watched a movie or read a book about film editing knows – the fundamental unit of edited video material is the *shot*, not the frame [Monaco, 2009; Katz, 2019; Murch, 2021]. Shots are used to *structure* the video material, defining the granularity and temporal context through choices in its scale (close-up, long shot, *etc.*) and its movement (panning, tracking, *etc.*). The "film grammar" is then used to convey meaning through specific editing choices on shot transitions (cuts, fades, *etc.*), durations, and composition (threads, montage, *etc.*).

Current approaches to automating AD generation, and video large language models (VideoLLMs) in general, are not shot-aware, hindering interpretation of edited video material that has frequent shot transitions [Tang et al., 2024]. In this paper, our principal objective is to incorporate shot information and editing structure into the AD generation process. To this end, we consider temporal context in terms of shots, and take account of the two key properties of *thread structure* and *shot scale*.

Thread structure identifies the sequence of shots captured with the same camera. An example of a scene with multiple simultaneous threads in space-time is shown in Figure 7.3. We develop a robust thread clustering method and use its predictions to guide VLMs in understanding shot-wise relationships.

Shot scale typically implies the type of content of the frame. For instance, close-up shots often highlight characters' facial expressions or gaze interactions, while long shots tend to depict the overall environment and ambience of the scene, as shown

in Figure 7.4 (top). We leverage this property by developing an off-the-shelf shot scale classifier and employing a scale-dependent prompting strategy to improve the contextual relevance of VLM-generated descriptions.

In this paper, we develop a *training-free* AD generation approach. Inspired by AutoAD-Zero [Xie et al., 2024a], as illustrated in Figure 7.1, we adopt a two-stage pipeline in which a VideoLLM generates dense text descriptions in the first stage, followed by an LLM that produces the final AD outputs from this text. We improve the effectiveness of training-free approaches by incorporating two key improvements based on: (i) shot-based temporal context; and (ii) the shot scale and thread structure, that are both crucial to cinematic composition and understanding. Previous methods could describe human-object interactions or human-human interactions, such as looks [Marin-Jimenez et al., 2020], within a frame or shot. With the improvements we introduce, the generated AD also includes such interactions when they are implied by the shot structure.

As well as the challenge of generating ADs, another challenge is *how to evaluate* the predicted ADs. Apart from conventional metrics [Vedantam et al., 2015; Lin, 2004; Banerjee and Lavie, 2005; Papineni et al., 2002], several new AD metrics [Zhang et al., 2024a; Han et al., 2024; Han et al., 2023b] have been proposed, including CRITIC [Han et al., 2024] measuring *who* is mentioned in the AD. However, these metrics fail to emphasise character *actions* – one of the most critical aspects of ADs. To address this, we introduce an *action score*, assessing whether the predictions accurately capture the actions described in GT AD, independent of character information.

Moreover, as ADs are time-limited, there is always a choice on which visual aspect (characters, actions, objects, *etc.*) should be included. Consequently, a single video clip can have multiple equally valid ADs, each highlighting a different aspect. This observation is supported by the inter-rater agreement experiments in AutoAD-III [Han et al., 2024] and previous user studies [Wang et al., 2021b]. Therefore, beyond evaluating single AD performance, we assess our framework's capability as an *assistant* to generate multiple AD candidates, and report the performance of the selected *one* AD.

In summary, we propose an enhanced training-free AD generation framework, with

the following contributions:

- We incorporate shot-based temporal context into AD generation via training-free prompting techniques including shot number referral and dynamic frame sampling.
- We develop state-of-the-art methods for thread structure and shot scale predictions, and demonstrate that incorporating predicted film grammar knowledge enhances AD generation.
- We improve the current AD evaluation by introducing the character-free action score, and a new assistance-oriented evaluation protocol.
- Our approach achieves state-of-the-art performance in training-free AD generation, and furthermore, surpasses fine-tuned models on multiple benchmarks. This is the first time a training-free approach has achieved superior performance to fine-tuned methods.

7.2 Related work

Audio Description generation. Efforts have been made to curate Audio Description (AD) datasets for both movies [Soldan et al., 2022; Han et al., 2024] and TV series [Xie et al., 2024a], with human annotations sourced from platforms such as AudioVault [AudioVault, 2025].

For automatic AD generation, prior works [Han et al., 2023b; Han et al., 2023a; Han et al., 2024; Lin et al., 2024; Wang et al., 2025a; Fang et al., 2024] fine-tune pre-trained VLMs on AD annotations to produce descriptions in an end-to-end manner. However, these methods face challenges due to limited high-quality AD annotations and the high computational cost of fine-tuning each new backbone. Alternatively, training-free approaches [Zhang et al., 2024a; Ye et al., 2024; Chu et al., 2024; Xie et al., 2024a] have gained traction for their scalability and flexibility, allowing customised AD output based on official guidelines [*Guidelines for Audio Describers* n.d.] or specific needs. Yet, these methods still lag in performance, while our approach is the first to achieve results on par with fine-tuned methods.

Instead of limiting the video input to each AD interval, UniAD [Wang et al., 2025a]

and DistinctAD [Fang et al., 2024] fine-tune VLMs with multiple AD clip inputs to incorporate broader temporal context. In contrast, our method systematically extends AD clip to adjacent shots and introduces a training-free approach that enables pre-trained VideoLLMs to better capture localised temporal context.

Film grammar analysis. Prior research has sought to understand film grammar from two major perspectives: (i) intra-shot properties, (ii) shot-wise relationships.

For individual shots, several datasets [Rao et al., 2020; Argaw et al., 2022; Huang et al., 2020; Jiang et al., 2022; Savardi et al., 2023] categorise their characteristics based on camera setups, including shot scales (examples shown in Figure 7.4) and camera movements. Correspondingly, various models [Rao et al., 2020; Chen et al., 2022; Lu et al., 2024; Li et al., 2023c; Vacchetti and Cerquitelli, 2022] have been developed for shot type classification.

Regarding shot-wise relationships, a few datasets [Argaw et al., 2022; Chen et al., 2023a; Pardo et al., 2022] have been proposed to explore transitions (*i.e.* cuts) between shots, which have also been leveraged in video content generation [Rao et al., 2022; Zhang et al., 2023e; Fedorishin et al., 2024; Pardo et al., 2024]. Beyond pairwise shot transitions, research has also investigated longer temporal contexts with thread-based editing structures. Notably, Hoai et al. introduced the Thread-Safe [Hoai and Zisserman, 2014] dataset, demonstrating that thread information can enhance action recognition. These structures have also been utilised in video-based face and people clustering [Tapaswi et al., 2014; Brown et al., 2021].

While prior work on film grammar has mainly focused on classification and generation tasks, we specifically utilise shot scale and thread structure information to enhance AD generation in movies and TV series.

Dense video captioning. Dense video captioning is closely related to AD generation. Early works [Krishna et al., 2017; Iashin and Rahtu, 2020a; Iashin and Rahtu, 2020b; Wang et al., 2018; Wang et al., 2020] in video captioning typically treat event localisation and captioning as independent stages, whereas more recent approaches [Yang et al., 2023a; Perrett et al., 2024; Wu et al., 2024a; Islam et al., 2024; Kim et al., 2024; Zhou et al., 2024; Zhou et al., 2018b; Wang et al., 2021a;

Chen and Jiang, 2021; Deng et al., 2021; Mun et al., 2019; Rahman et al., 2019] integrate these tasks in an end-to-end manner. Video captioning benchmarks cover a range of domains, including cooking [Zhou et al., 2018a], actions [Krishna et al., 2017], movies [Rohrbach et al., 2015], TV series [Lei et al., 2020], and open-domain settings [Shvetsova et al., 2023; Miech et al., 2019; Zhao et al., 2023b; Zellers et al., 2022; Xue et al., 2022; Hendricks et al., 2018; Chen et al., 2024b; Chen and Dolan, 2011; Xu et al., 2016].

Temporal grounding in VLM. To equip conventional VLMs with temporal grounding capability, some studies [Huang et al., 2024b; Wang et al., 2024d; Li et al., 2024] generate additional data with enhanced temporal information for fine-tuning. A more common approach explicitly incorporates temporal information into inputs, either by inserting temporal tokens [Huang et al., 2024a; Hong et al., 2024; Guo et al., 2024a; Guo et al., 2024b; Qian et al., 2024; Wang et al., 2024a; Chen et al., 2024a; Deng et al., 2024] or embedding time information within visual tokens [Huang et al., 2024b; Wang et al., 2024d; Li et al., 2024].

Training-free approaches [Qu et al., 2024; Zheng et al., 2024a; Wu et al., 2025] have also been explored for temporally grounded understanding. Notably, a recent work [Wu et al., 2025] achieves temporal grounding by overlaying frame numbers as visual prompts. Instead of using uniformly sampled timestamps, we leverage the natural shot structures in movies and TV series, adopting them as fundamental units for training-free temporal referral.

Evaluation for Audio Description. Early AD evaluation often adopts captioning metrics, including CIDEr [Vedantam et al., 2015], ROUGE [Lin, 2004], BLEU [Papineni et al., 2002], and METEOR [Banerjee and Lavie, 2005], which measure word/n-gram overlap with different weighting strategies. Other metrics focus on semantics, leveraging rule-based scene graph matching (SPICE [Anderson et al., 2016]), word embedding similarity (BERTScore [Zhang* et al., 2020], Recall@k/N [Han et al., 2023b]), or LLM-based evaluation [Han et al., 2024; Zhang et al., 2024a].

However, the fixed and distinct set of character names in each video can bias conventional captioning metrics. For instance, TF-IDF [Robertson, 2004] weighting

in CIDEr assigns high importance to character names. To provide a more independent measure of character names and other AD content, CRITIC [Han et al., 2024] assesses character recognition in predicted ADs, while we develop an action-centric metric with minimised dependence on character names.

Most prior works evaluate single AD outputs; however, training-free methods are compatible with generating multiple candidates for selection. While early studies [Wang et al., 2021b] explored this through user studies, we propose a quantitative evaluation protocol to further assess AD generation methods as assistive tools.

7.3 Training-free AD generation framework

Given a video clip $\mathcal{V} = \{\mathcal{I}_0, ..., \mathcal{I}_T\}$, the task of audio description is to generate a concise narration $\mathcal{N}$ describing what happens around a given AD interval $[t_A, t_B]$. In this work, we propose a two-stage framework that leverages VideoLLMs and LLMs to predict ADs in a training-free manner.

In Stage I, we employ a VideoLLM that takes a sequence of frames (from multiple shots) as input and generates a dense description $\mathcal{D}$, guided by instructions $\mathcal{P}_{\text{VideoLLM}}$:

$$\mathcal{D} = \text{VideoLLM}(\mathcal{V}, [t_A, t_B], \mathcal{P}_{\text{VideoLLM}}) \tag{7.1}$$

In Stage II, we then prompt an LLM (with instructions $\mathcal{P}_{\text{LLM}}$) to extract key information from the dense Stage I description and format it into an AD-style narration $\mathcal{N}$:

$$\mathcal{N} = \text{LLM}(\mathcal{D}, [t_A, t_B], \mathcal{P}_{\text{LLM}}) \tag{7.2}$$

In this section, we focus on enhancing the visual understanding of the Stage I VideoLLM for edited video material, and make three innovations: In Section 7.3.1, we incorporate shot-based temporal context into Stage I visual inputs. In Section 7.3.2, we leverage the thread structure to enrich cross-shot understanding. Finally, in Section 7.3.3, we incorporate shot-scale awareness into Stage I prompt formuation.

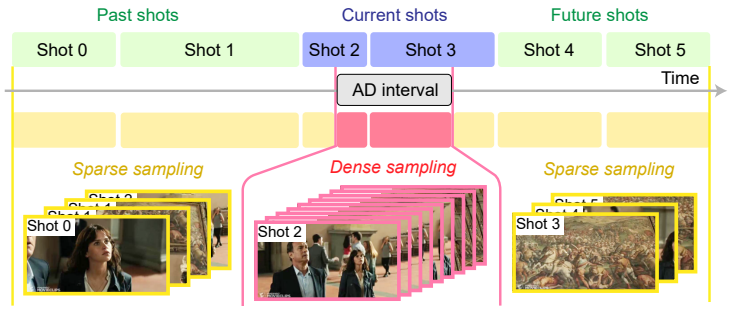


Figure 7.2: **Shot-based temporal context,** where *current shots* are defined as those temporally overlapping with the AD interval. *Past shots* and *future shots* provide extended contextual information. Shot numbers are visually overlaid on the top-left of each frame, and frames within the AD interval are sampled more densely than context frames.

7.3.1 Leveraging shot-based temporal context

Regarding the visual inputs to VideoLLM, prior works [Lin et al., 2024; Han et al., 2024] often sample frames $\{\mathcal{I}_{t_A}, ..., \mathcal{I}_{t_B}\}$ that directly correspond to the AD interval $[t_A, t_B]$. However, this approach can be problematic due to (i) misalignment between the AD interval and the actual timestamps when the action occurs, and (ii) the lack of contextual information from adjacent shots. Therefore, we investigate how incorporating temporal context can enhance the understanding of the video clip.

Structuring temporal context with shots. To obtain more visual context information, instead of simply extending the AD interval by fixed timestamps, we explore a more structured approach that treats shots as the fundamental units.

Specifically, we apply an off-the-shelf shot segmentation model to partition the entire video clip into individual shots. For each AD interval, we first identify the shots that (partially) overlap with it, referred to as "current shots", as illustrated in Figure 7.2. We then consider *at most* two "past shots" and two "future shots" adjacent to current shots as temporal context. For all shots included, we label them sequentially from past to future with a number starting from *"Shot 0"*.

Emphasising the targeted AD interval. To ensure that the VideoLLM focuses on describing visual content within the AD interval, we propose two strategies: dynamic frame sampling and shot number referral.

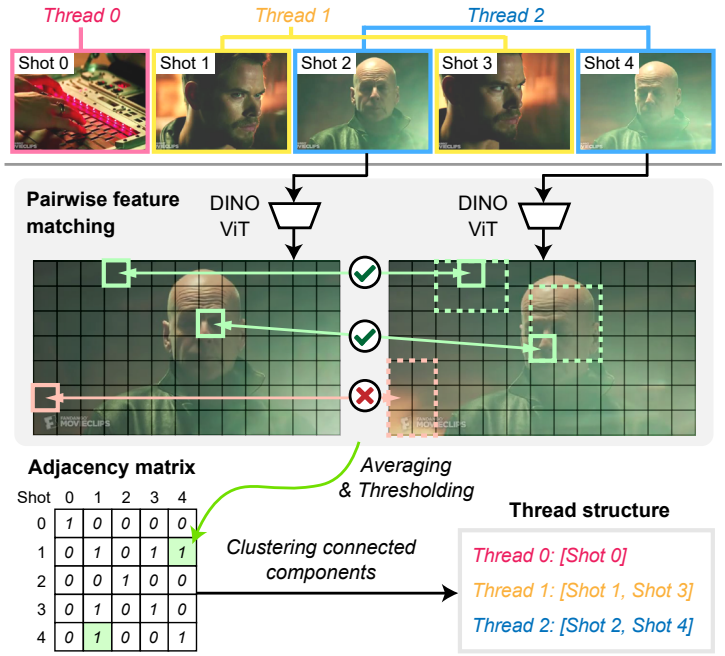


Figure 7.3: **Thread structure. Top:** Example of a thread structure with interleaving shots. **Bottom:** A training-free approach for thread clustering, where shots are pairwise compared using dense feature matching to construct an adjacency matrix, which is then used to predict the thread structure.

Dynamic frame sampling. As shown in Figure 7.2, to emphasise the frames of interest, we adopt denser sampling within the AD interval (red region) and sparser sampling for the surrounding context frames (yellow region). In practice, we specify fixed numbers of frames to be sampled within and outside the AD interval and apply uniform sampling according to these constraints.

Shot number referral. To further enhance the attention towards the current shot content, we label each sampled frame with its shot number (*e.g. "Shot 0"*) at the top-left. During the formulation of the text prompt, instead of prompting the VideoLLM to *"describe what happened in the video clip"*, we ask it to *"describe what happened in [Shot 2, Shot 3]"* (*i.e.* current shots). Through this visual-textual prompting strategy, we found that the VideoLLM could successfully interpret the meaning of shot numbers and refer to the correct shots.

7.3.2 Leveraging thread structure

Movies are generally edited such that viewpoints from two or more cameras are intertwined in shot *threads*, as illustrated in Figure 7.3 (top). These interleaved arrangements of shot threads often imply relationships between objects and charac-

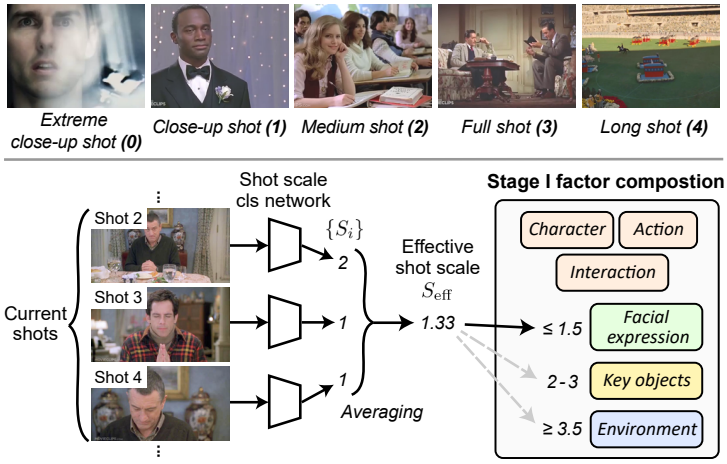


Figure 7.4: **Shot scales. Top:** Examples of five different shot scales. **Bottom:** Stage I factor composition based on shot scale classification. The scales of the current shots in the clip are predicted and averaged. The resulting effective shot scale then guides the formulation of the Stage I prompt, incorporating additional factors such as facial expressions, *etc.*

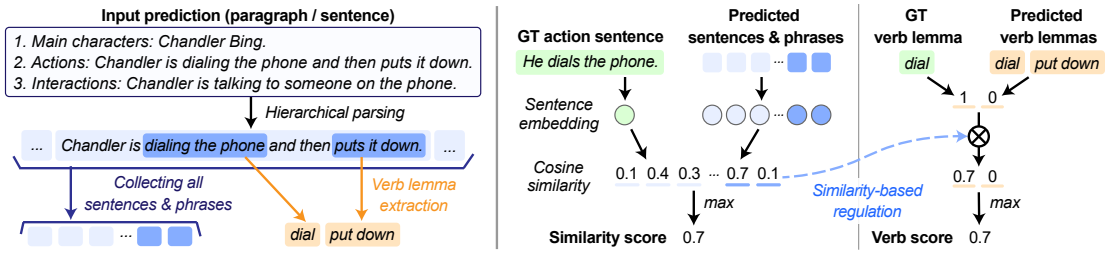


Figure 7.5: **Action score. Left:** Hierarchical prediction parsing. Given a predicted paragraph or sentence, it is first divided into a set of sentences (light blue) and action phrases (dark blue). Each action phrase is further processed to obtain its verb lemma (light orange). **Middle:** Scoring based on semantic similarity. Sentence embeddings for the GT action sentence and the set of predicted sentences and phrases are extracted, with the maximum cosine similarity defined as the similarity score. **Right:** Scoring based on verb matching. The predicted verb lemma is compared with the GT lemma and multiplied by the corresponding semantic-based similarity. The highest resultant score is defined as the verb score.

ters (*e.g.* gaze interactions) and their 3D arrangement. To leverage this information for Stage I description, we first determine the thread structure using a separate module, then incorporate it into the VideoLLM through prompt guidance.

Thread structure prediction. We develop a training-free method to predict thread clustering. The problem here is to determine if two shots correspond to the same viewpoint or not. Specifically, given two shots (Shot i and a later Shot j), we compare the last frame of Shot i ($\mathcal{I}^i_{T_i}$) with the first frame of Shot j ($\mathcal{I}^j_0$), extracting their DINOv2 [Oquab et al., 2024] features as $f^i_{T_i}$ and $f^j_0 \in \mathbb{R}^{h\times w\times c}$.

Inspired by [Jabri et al., 2020], we assess frame-wise dense feature correlations by

computing a cost volume between their feature maps:

$$\mathcal{C}_{p,q}^{i,j} = m(p,q) \circ (\hat{f}_{T_i;p}^{i} \cdot \hat{f}_{0;q}^{j}) \tag{7.3}$$

where $\hat{f}_{T_i;p}^{i}$ and $\hat{f}_{0;q}^{j} \in \mathbb{R}^c$ are the normalised p-th and q-th spatial elements in the respective feature maps. The binary attention mask $m(p,q)$ is set to 1 only if the spatial position of the q-th element is within an $n \times n$ neighbourhood of the p-th element.

We then apply a softmax operation along the last dimension (q) and find the maximum similarity for each p, followed by averaging over all p-th elements to obtain a matching score between Shot i and Shot j:

$$s^{i,j} = \frac{1}{N} \sum_{p}^{N} \max_{q} \left(\frac{\exp(\mathcal{C}_{p,q}^{i,j}/\tau)}{\sum_{l}^{N} \exp(\mathcal{C}_{p,l}^{i,j}/\tau)} \right) \tag{7.4}$$

where N denotes the number of feature patches, and τ is the softmax temperature.

Intuitively, as shown in Figure 7.3, this process effectively checks whether each patch (p) in one shot frame matches with a patch (q) in the other shot frame at a roughly similar spatial position (*i.e.* within an $n \times n$ neighbourhood).

Finally, we construct an adjacency matrix based on the predicted scores $s^{i,j}$ for all possible pairs of Shot i and Shot j. By thresholding the adjacency matrix (threshold ϵ) and identifying the largest connected components, we cluster the set of shots into multiple threads.

Thread structure injection. Once the thread information is obtained, we inject it into the text prompt $\mathcal{P}_{\text{VideoLLM}}$ for the Stage I VideoLLM. In practice, we conduct this information injection only to clips that exhibit thread structures (*i.e.* $N_{\text{thread}} < N_{\text{shot}}$). For each given thread [Shot i, ..., Shot j], we formulate the prompt as: *"[Shot i, ..., Shot j] share the same camera setup"*. This statement implies that the cameras in these shots maintain consistent angles and scales.

Rather than simply providing this information, we further engage the VideoLLM by asking it to explain why the given thread structure is correct. This effectively corresponds to a Chain-of-Thought (CoT) process, enhancing its understanding of these repetitive thread structures.

7.3.3 Leveraging shot scale information

In movies and TV series, shot scales are often carefully designed during filming or post-editing to implicitly convey information to the audience. Our objective is to use the shot scale to choose what should be included in the Stage I prompts.

Shot scale classification. We first build a shot scale classification network by fine-tuning a pre-trained DINOv2 [Oquab et al., 2024] model. For all *current* shots, we classify their shot scales $\{S_i\}$ into one of *five* classes, represented by values 0–4, as shown in Figure 7.4 (top). We then compute their average to obtain the effective shot scale S_{eff}.

Stage I factor composition. We then leverage the predicted shot scale to determine the factors to include in Stage I instructions. We first consider three fixed factors that form the basis of ADs, namely characters, actions, and interactions. The additional factor can be determined through applying a set of thresholds to S_{eff}, as detailed in Figure 7.4 (bottom). For example, for close-up shots ($S_{\text{eff}} \leq 1.5$), we would ask the VideoLLM to additionally describe the "facial expression", whereas for long shots ($S_{\text{eff}} \geq 3.5$), a description on the "environment" will be included.

7.4 Action score

In this section, we introduce a new metric, termed "action score", which focuses on whether a specific ground truth (GT) action is captured within a predicted Stage I description paragraph or a Stage II AD output. For instance, for the GT action *"He dials the phone"*, we want the metric to measure the action of *"dial the phone"*, but not be sensitive to character names and other predicted content. Therefore, this metric is designed to possess two key properties: (i) it is character-free, meaning that the presence of character names has minimal impact on the evaluation, and (ii) it is recall-oriented, without penalising additional action information in the prediction.

Preprocessing of GT actions. For each GT AD, we extract the character-free GT actions by (i) replacing character names with pronouns, and (ii) splitting the

AD into subsentences, each containing one action (verb). For example, given a GT AD *"Chandler dials the phone, then hurriedly hangs up."*, the extracted GT actions are *"He dials the phone."* and *"He hurriedly hangs up."*

Hierarchical parsing of predictions. To process the predicted paragraph during evaluation, we first decompose it into individual sentences (light blue), as shown in the left column of Figure 7.5. For each predicted sentence, we then perform rule-based dependency parsing to obtain action phrases (deep blue). Then, all extracted sentences and action phrases are collected to form a prediction set. Additionally, for each action phrase, we extract the corresponding verb lemma (*i.e.* the root form of verbs).

Action score computation. The action score features a combination of semantic-based and verb-matching-based components, as illustrated in the middle and right columns of Figure 7.5, respectively.

Semantic-based evaluation. For each GT action sentence, we assess its semantic similarity with the set of predicted sentences and phrases. Specifically, we employ a general text embedding (GTE) model to compute *sentence-level* embeddings for the GT action sentence e_{GT} and each element in the predicted set $\{e_{\text{pred};i}\}$. By computing cosine similarities and taking the maximum value, the *similarity score* is defined as $s_{\text{sim}} = \max_i(s_{\text{sim};i}) = \max_i[(e_{\text{GT}} \cdot e_{\text{pred};i})/(\|e_{\text{GT}}\|\|e_{\text{pred};i}\|)]$.

Verb-matching-based evaluation. In addition to semantic matching, we credit predictions that contain the same verbs as those in GT actions. Practically, we compare the predicted verb lemmas with GT verb lemma and compute binary matching scores $\{m_j\}$. These matching scores are further weighted by the corresponding similarity scores $\{s_{\text{sim},j}\}$ through element-wise multiplication. Finally, we take the maximum of the regulated scores to compute the *verb score*, *i.e.* $s_{\text{verb}} = \max_j(s_{\text{sim};j} \circ m_j)$.

To obtain the final action score, we combine the scores from both sources using a weighted average $s_{\text{action}} = \alpha_{\text{sim}} \circ s_{\text{sim}} + \alpha_{\text{verb}} \circ s_{\text{verb}}$, with α_{sim} and α_{verb} denoting the weighted factors.

7.5 Experiments

This section begins with datasets and metrics for AD generation in Section 7.5.1, followed by implementation details in Section 7.5.2. Results on film grammar predictions are presented in Section 7.5.3 and 7.5.4, while Section 7.5.5 analyses human alignment with action scores. AD generation results are detailed in Section 7.5.6, Section 7.5.7, and 7.5.8.

7.5.1 AD generation datasets and metrics

We evaluate our framework on AD generation datasets for both movies (CMD-AD [Han et al., 2024], MAD-Eval [Han et al., 2023b]) and TV series (TV-AD [Xie et al., 2024a]). In more detail, CMD-AD is constructed by aligning ground truth ADs with the Condensed Movie Dataset (CMD) [Bain et al., 2020], comprising 101k ADs (94k for training and 7k for testing) from 1.4k movies. MAD-Eval consists of 6.5k ADs sampled from 10 movies within LSMDC [Rohrbach et al., 2015]. On the other hand, TV-AD features 34k AD annotations from 13 TV series, with its test set sourced from TVQA [Lei et al., 2018], consisting of 3k ADs.

For AD evaluation, we follow the prior works [Han et al., 2024; Fang et al., 2024] to assess general prediction quality using CIDEr [Vedantam et al., 2015], Recall@k/N [Han et al., 2023a], and LLM-AD-Eval [Han et al., 2024]. We also consider character recognition accuracy using CRITIC [Han et al., 2024] and character-free action evaluation using action scores (described in Section 7.4).

7.5.2 Implementation details

In this section, we provide key implementation details for AD generation and action score computation. For additional information on film grammar predictions and other details, please refer to the Appendix.

Shot detection. To segment the video clip into shots, we use PySceneDetect [SceneDetect, 2025] with the "Adaptive Detection" method, which compares the ratio of pixel changes with the neighbouring frames. On average, each shot spans 3.5s, 3.3s, 3.8s in CMD-AD, TV-AD, and MAD-Eval, respectively.

Table 7.1: **Thread structure prediction on thread-safe.** "Side" refers to comparisons between the temporally nearest frames in two shots; "Mid" refers to comparisons between the middle frames of each shot; "All" refers to comparisons between all frame pairs from the two shots.

Feature	Frame setup	Precision	Recall	AP	WCP
CLIP-L14 CLS	Side	0.691	0.635	0.705	0.922
DINOv2-L14 CLS	Side	0.759	0.683	0.788	0.933
DINOv2-g14 CLS	Side	0.761	0.675	0.786	0.936
DINOv2-g14 spatial	Mid	0.808	0.717	0.822	0.953
DINOv2-g14 spatial	All	0.870	0.795	0.896	0.964
DINOv2-g14 spatial	Side	**0.878**	**0.799**	**0.902**	**0.965**

AD generation setting. During dynamic frame sampling, we select a total of 32 frames, with 16 frames uniformly sampled within and outside the AD interval, respectively. For character recognition, we adopt the same visual-textual prompting method proposed by [Xie et al., 2024a], which applies coloured circles around faces for visual character indication. For simplicity, the visualisations in this paper do not display these circle labels.

Regarding the base models, we use Qwen2-VL-7B [Wang et al., 2024b] as the VideoLLM in Stage I and LLaMA3-8B [Meta, 2024] as the LLM in Stage II. This setup is used as the *default* unless stated otherwise. Additionally, we explore the framework with the proprietary GPT-4o [OpenAI, 2024] model for both stages.

Action score evaluation setting. For action score computation, we set the weight factors as $\alpha_{\text{sim}} = 0.8$ and $\alpha_{\text{verb}} = 0.2$. When aggregating the action score results, we first average over multiple GT actions within each GT AD, and then perform global averaging across all AD samples. Moreover, in practice, we find that most action scores are clustered within the range of $0.25 - 0.75$. To improve clarity, we apply further rescaling $f(x) = (x - 0.25) \times 2$ as post-processing. For evaluations in this paper, unless otherwise specified, we use the action score to assess the Stage II AD outputs.

7.5.3 Thread structure prediction

We evaluate thread structure prediction on Thread-Safe [Hoai and Zisserman, 2014], which consists of approximately 4.7k video clips collected from 15 TV series. Each video clip contains a multi-shot scene with corresponding thread clusters manually annotated.

For evaluation, we first construct an adjacency matrix from the GT clusters and extract binary GT labels $\hat{s}_{i,j}$ from the off-diagonal entries, where each label indicates whether a given pair of shots belong to the same cluster. We then compute the Average Precision (AP) between the predicted relationships $\{s_{i,j}\}$ and the ground truth $\{\hat{s}_{i,j}\}$, as well as report the precision and recall values. Additionally, we directly compare the predicted clusters with ground truth clusters by reporting the weighted clustering purity (WCP) [Tapaswi et al., 2014].

Table 7.1 verifies the choice of DINOv2-g14 [Oquab et al., 2024] as the feature extractor for frame pair comparison. Compared to abstract CLS tokens, dense spatial matching achieves higher AP in frame pair relationship prediction and higher WCP in thread clustering. Additionally, we observe that using the temporally closest frames from two shots ("Side") leads to a noticeable performance improvement. This can be attributed to the continuous story flow across shots within the same thread.

7.5.4 Shot scale classification

Following prior work on shot scale classification [Lu et al., 2024; Li et al., 2023c], we use the MovieShots [Rao et al., 2020] dataset, which consists of 46k shots (train:val:test = 7:1:2) collected from over 7k movie trailers. We follow its definition of shot scales, categorising shots into five classes ranging from extreme close-up to long shots, as illustrated in Figure 7.4 (top). To evaluate the model performance, we report classification accuracy and Macro-F1 [Li et al., 2023c] scores on the MovieShots test set.

Since previous state-of-the-art methods on shot scale classification are not open-sourced, we develop a new network by fine-tuning DINOv2 [Oquab et al., 2024], achieving superior performance over prior approaches that rely on additional optical flow or SAM-based mask inputs, as demonstrated in Table 7.2.

7.5.5 Human alignment with action scores

Action scores aim to evaluate whether a GT action is captured within a predicted description, making them recall-oriented. Such descriptions can be in the form of paragraphs (Stage I descriptions) or single sentences (Stage II ADs). To assess

Table 7.2: **Shot scale classification on MovieShots.**

Metric	Input	Accuracy	Macro-F1
ViViT [Arnab et al., 2021]	RGB	0.747	0.751
SGNet [Rao et al., 2020]	RGB + Flow	0.875	–
Lu et al. [Lu et al., 2024]	RGB + mask	0.892	–
Li et al. [Li et al., 2023c]	RGB	0.895	0.897
Ours	RGB	**0.897**	**0.899**

Table 7.3: **Comparison of action score with other metrics.** The listed metrics measure the similarity between predicted paragraphs/sentences and ground truth actions. The reported values indicate the correlation (*i.e.* alignment) between these metrics and human-annotated scores.

Metric	Paragraph		Sentence	
	Pearson	Spearman	Pearson	Spearman
CIDEr [Vedantam et al., 2015]	0.205	0.264	0.412	0.528
ROUGE-L [Lin, 2004]	0.305	0.280	0.526	0.512
METEOR [Banerjee and Lavie, 2005]	0.462	0.406	0.602	0.641
BLEU-1 [Papineni et al., 2002]	0.265	0.264	0.477	0.481
SPICE [Anderson et al., 2016]	0.022	0.048	0.031	0.012
BERTScore [Zhang* et al., 2020]	0.377	0.393	0.508	0.507
LLM-based (LLaMA3.1-70B [Meta, 2024])	0.569	0.491	0.779	0.798
LLM-based (GPT-4o [OpenAI, 2024])	0.742	0.678	0.797	0.807
Action Score (w/o verb matching)	0.735	0.728	0.765	0.790
Action Score (w verb matching)	**0.749**	**0.729**	**0.806**	**0.820**

whether action scores align with human judgments, we create a dataset containing pairs of predicted descriptions and GT actions. For each GT action, human annotators *manually annotate* the quality of predictions into $\{0, 1, 2, 3\}$ based on the relevance towards GT action ranging from "unrelated" (0) to "exact matching" (3).

Comparison with other metrics. Next, we use the human-annotated scores as a reference to compare different metrics in terms of human agreement (measured by correlations), as reported in Table 7.3. Additionally, we consider an LLM-based metric to predict scores, following the same scoring criteria as human annotations. In general, the action score achieves the best correlations with human annotations. Note, it is also more efficient than LLM-based metrics, with 0.15s compared to 6s per prediction evaluation. For more details regarding this human agreement study, please refer to the Appendix.

7.5.6 AD generation – evaluation of components

Shot-based temporal context. We investigate different setups for leveraging temporal context in AD generation, as shown in Table 7.4, leading to the following

Table 7.4: **Leveraging shot-based temporal context.** Key changes relative to the default setting (Exp. F) are highlighted in light blue. "Temporal context" indicates the number of context (past & future) shots. "Shot num." refers to overlaying the shot number at the top-left of each current shot frame, while "frame box" represents highlighting the boundary of each current shot frame with a red box.

Exp.	Temporal context	Frame sampling	Shot label	CMD-AD			TV-AD		
				CIDEr	CRITIC	Action	CIDEr	CRITIC	Action
A	–	–	–	22.4	45.7	27.0	26.0	42.2	22.3
B	1 shot	Dyn.	Shot num.	24.7	46.8	**27.8**	**28.9**	41.6	**23.1**
C	2 shots	Dyn.	–	24.5	46.6	27.4	27.8	**42.3**	22.3
D	2 shots	Dyn.	Frame box	24.8	46.4	27.4	25.0	41.8	22.8
E	2 shots	Uni.	Shot num.	24.1	47.5	26.2	26.5	42.2	22.0
F	2 shots	Dyn.	Shot num.	**25.1**	**47.5**	**27.8**	**28.9**	42.1	23.0

Table 7.5: **Thread structure injection.** Thread structure information is injected only into subsets predicted to exhibit thread structures (∼30% in CMD-AD and ∼60% in TV-AD).

Thread structure	CMD-AD subset			TV-AD subset		
	CIDEr	CRITIC	Action	CIDEr	CRITIC	Action
✗	29.9	47.5	27.4	28.8	42.0	22.6
✓	**30.7** ↑0.8	**48.9** ↑1.4	**27.7** ↑0.3	**30.7** ↑1.9	**42.7** ↑0.7	**22.9** ↑0.3

observations: (i) Expanding the temporal context range noticeably boosts the performance, with gains saturating around "2 shots" (Exp. A, B, and F); (ii) "Shot number referral" is the most effective strategy for outlining the current shot. (Exp. C, D, and F); (iii) Dynamically sampling the current shots at a higher frame rate boosts AD generation (Exp. E and F). The latter two improvements can be attributed to a more efficient focus on the visual content around the targeted AD interval.

Thread structure injection. After extending the context information with neighbouring shots, we further enhance the VideoLLM's understanding by incorporating thread structures. Note that this guidance is applied only to video sequences exhibiting thread structures. As shown in Table 7.5, this additional information improves AD generation performance across both datasets.

Scale-dependent Stage I factors. Table 7.6 explores the impact of different Stage I factors on final AD performance. Using shot scales as guidance for Stage I factor formulation (*i.e.* scale-dependent) not only outperforms configurations with single fixed factors but also surpasses the case where all factors are included in Stage I. This could be attributed to that the scale-dependent description con-

Table 7.6: **Factors included in Stage I description.** Base: character + action + interaction; Face: facial expression; Env.: environment; Obj.: object. "Scale-dependent" refers to our approach, which leverages shot scale predictions to determine the relevant factors for each clip.

Stage I factors	CMD-AD			TV-AD		
	CIDEr	CRITIC	Action	CIDEr	CRITIC	Action
Base	25.4	47.4	27.7	29.2	42.0	23.0
Base + Face (AutoAD-Zero)	25.2	46.8	27.8	30.0	42.3	22.5
Base + Obj.	26.1	45.8	27.9	30.4	**42.9**	22.8
Base + Env.	25.2	46.7	27.4	29.8	40.7	22.2
Base + Face + Obj. + Env.	26.0	47.2	27.4	30.1	42.7	22.9
Scale-dependent (Ours)	**26.3**	**47.8**	**28.4**	**31.1**	42.2	**23.9**

Table 7.7: **Quantitative comparison on CMD-AD and TV-AD.** For training-free methods, "VLM" and "LLM" refer to the models used in separate stages, while for fine-tuned models, they denote the pre-trained components within an end-to-end model.

Method	VLM	LLM	Training -free	Propriet. model	CMD-AD					TV-AD				
					CIDEr	CRITIC	Action	R@1/5	LLM-AD-Eval	CIDEr	CRITIC	Action	R@1/5	LLM-AD-Eval
AutoAD-II [Han et al., 2023a]	CLIP-B32	GPT-2	✗	✗	13.5	8.2	–	26.1	2.08 | –	–	–	–	–	–
AutoAD-III [Han et al., 2024]	EVA-CLIP	LLaMA2-7B	✗	✗	25.0	32.7	31.5	31.2	2.89 | 2.01	26.1	28.8	26.4	30.1	2.78 | 1.99
DistinctAD [Fang et al., 2024]	CLIP_{AD}-B16	LLaMA3-8B	✗	✗	22.7	–	–	33.0	2.88 | 2.03	27.4	–	–	32.1	2.89 | 2.00
Video-LLaMA [Zhang et al., 2023c]	Video-LLaMA-7B	–	✓	✗	4.8	0.0	–	22.0	1.89 | –	–	–	–	–	–
VideoBLIP [Yu, 2023]	VideoBLIP	–	✓	✗	5.2	0.0	–	23.6	1.91 | –	–	–	–	–	–
AutoAD-Zero [Xie et al., 2024a]	VideoLLaMA2-7B	LLaMA3-8B	✓	✗	17.7	43.7	25.5	26.9	2.83 | 1.96	22.6	39.4	21.7	27.4	2.94 | 2.00
AutoAD-Zero [Xie et al., 2024a]	Qwen2-VL-7B	LLaMA3-8B	✓	✗	21.9	44.3	26.9	30.8	3.00 | 2.20	26.4	41.6	22.1	30.4	3.05 | 2.27
Ours	Qwen2-VL-7B	LLaMA3-8B	✓	✗	**26.3**	47.8	28.4	33.0	3.15 | 2.42	31.1	42.2	23.9	33.1	3.09 | 2.35
AutoAD-Zero [Xie et al., 2024a]	GPT-4o	GPT-4o	✓	✓	22.4	45.1	30.7	32.9	3.08 | 2.49	30.9	44.4	26.8	34.7	**3.12** | 2.57
Ours	GPT-4o	GPT-4o	✓	✓	26.1	**49.1**	**32.5**	**36.5**	**3.17** | **2.66**	**34.2**	**46.5**	**27.4**	**36.6**	**3.12** | **2.59**

tains more relevant and less redundant information, enabling more efficient AD extraction in Stage II.

7.5.7 AD generation – comparison with SotA

Table 7.7 provides a comprehensive summary of AD generation performance on CMD-AD and TV-AD, comparing across training-free methods with and without proprietary models, as well as models fine-tuned on human-annotated ADs. Notably, with the same base model setup (Qwen2-VL-7B + LLaMA3-8B), our training-free framework significantly outperforms AutoAD-Zero, primarily due to the usage of temporal context and film grammar information. By incorporating the more powerful GPT-4o models, our performance scales up further, surpassing even existing fine-tuned models.

Regarding the MAD-Eval results, as shown in Table 7.8, our framework achieves state-of-the-art performance among all training-free approaches, also demonstrating competitiveness with fine-tuned models.

Table 7.8: **Quantitative comparison on MAD-Eval.** For training-free methods, "VLM" and "LLM" refer to the models used in separate stages, while for fine-tuned models, they denote the pre-trained components within an end-to-end model.

Method	VLM	LLM	Training -free	Propriet. model	MAD-Eval					
					CIDEr	R@5/16	Rouge-L	SPICE	METEOR	BLEU-1
ClipCap [Mokady et al., 2021]	CLIP-B32	GPT-2	✗	✗	4.4	36.5	8.5	1.1	–	–
CapDec [Nukrai et al., 2022]	–	–	✗	✗	6.7	–	8.2	1.4	–	–
AutoAD-I [Han et al., 2023b]	CLIP-B32	GPT-2	✗	✗	13.4	42.1	11.9	4.4	–	–
AutoAD-II [Han et al., 2023a]	CLIP-B32	GPT-2	✗	✗	19.5	51.3	13.4	–	–	–
AutoAD-III [Han et al., 2024]	EVA-CLIP	LLaMA2-7B	✗	✗	24.0	52.8	13.9	6.1	5.5	13.1
MovieSeq [Lin et al., 2024]	CLIP-B16	LLaMA2-7B	✗	✗	24.4	51.6	15.5	7.0	–	–
DistinctAD [Fang et al., 2024]	CLIP_{AD}-B16	LLaMA3-8B	✗	✗	27.3	56.0	**17.6**	8.3	–	–
UniAD [Wang et al., 2025a]	CLIP-L14	LLaMA-8B	✗	✗	**28.2**	54.9	17.2	–	–	–
Video-LLaMA [Zhang et al., 2023c]	Video-LLaMA-7B	–	✓	✗	4.8	33.8	–	–	–	–
Video-BLIP [Yu, 2023]	Video-BLIP	–	✓	✗	5.0	35.2	–	–	–	–
AutoAD-Zero [Xie et al., 2024a]	VideoLLaMA2-7B	LLaMA3-8B	✓	✗	22.4	47.0	14.4	7.3	6.6	15.1
AutoAD-Zero [Xie et al., 2024a]	Qwen2-VL-7B	LLaMA3-8B	✓	✗	23.6	51.3	14.6	7.8	6.6	13.6
Ours	Qwen2-VL-7B	LLaMA3-8B	✓	✗	25.0	50.6	14.7	7.8	7.2	**16.2**
VLog [*Vlog: Video as A Long Document* 2023]	BLIP-2 + GRIT	GPT-4	✓	✓	1.3	42.3	7.5	2.1	–	–
MM-Vid [Lin et al., 2023]	GPT-4V	–	✓	✓	6.1	46.1	9.8	3.8	–	–
MM-Narrator [Zhang et al., 2024a]	Azure API + CLIP-L14	GPT-4V	✓	✓	9.8	–	12.8	–	7.1	10.9
MM-Narrator [Zhang et al., 2024a]	Azure API + CLIP-L14	GPT-4	✓	✓	13.9	49.0	13.4	5.2	6.7	12.8
LLM-AD [Chu et al., 2024]	GPT-4V	–	✓	✓	20.5	–	13.5	–	–	–
AutoAD-Zero [Xie et al., 2024a]	GPT-4o	GPT-4o	✓	✓	25.4	54.3	14.3	8.1	6.7	13.7
Ours	GPT-4o	GPT-4o	✓	✓	26.9	**56.4**	15.0	**8.5**	**7.4**	15.9

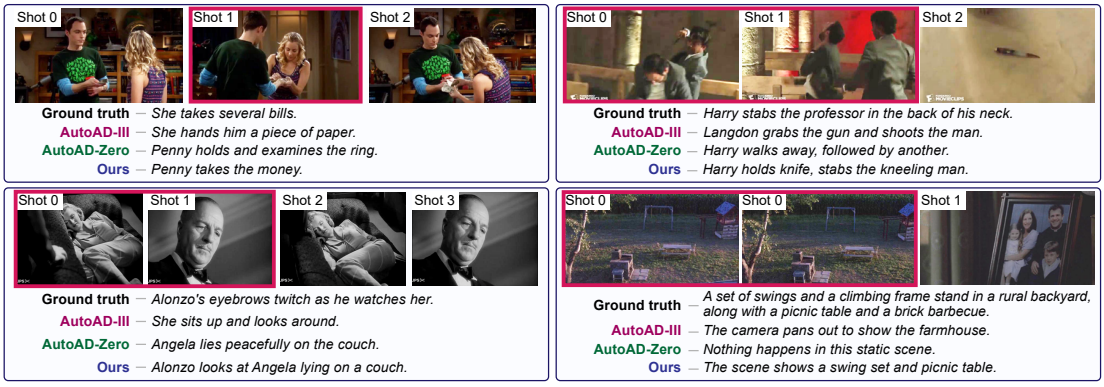


Figure 7.6: **Qualitative visualisations.** Current shots (corresponding to AD intervals) are outlined by red boxes for illustration purposes only. For simplicity, not all context shots are shown. Training-free methods adopt Qwen2-VL + LLaMA3. Examples are taken from *The Big Bang Theory* (S2E14) (top left), *Inferno* (2016) (top right), *The Asphalt Jungle* (1950) (bottom left), and *Signs* (2002) (bottom right). The top-left example demonstrates the benefits of shot-based temporal context, where the objects (*i.e.* bills) in Penny's hands are not clearly visible within the AD interval (Shot 1), leading to ambiguous or incorrect predictions by AutoAD-Zero. In contrast, our method successfully identifies the objects from the context shot (Shot 2). The top-right example describing the action of *Harry Sims* similarly verifies the effectiveness of incorporating context shots.

Qualitative visualisations. Figure 7.6 presents several qualitative examples, where the top two cases illustrate how temporal context information aids in identifying key objects.

In the bottom-left example, prior methods fail to associate characters, leading to the omission of the man (*Alonzo*). In contrast, our method recognises the thread structure (*i.e. [Shot 0, Shot 2], [Shot 1, Shot 3]*), which guides the correct prediction of the man's gaze direction towards the lying woman.

Table 7.9: **Assisted AD generation results.** All methods adopt Qwen2-VL + LLaMA3-8B as base models. The first row provides single AD generation results as references (labelled in gray), the rest rows report the performance of one selected AD out of five candidates. "Indep. output" denotes five random independent Stage II runs, with p as the hyperparameter for top-p (nucleus) sampling and τ_p as the sampling temperature. "Joint output" generates five ADs simultaneously in a single run.

Method	Candidate sampling	CMD-AD			TV-AD		
		CIDEr	CRITIC	Action	CIDEr	CRITIC	Action
Ours	Single AD (Ref.)	26.3	47.8	28.4	31.1	42.2	23.9
Ours	Indep. output ($p=0.90; \tau_p=0.6$)	33.3	49.6	32.5	41.2	44.3	28.6
Ours	Indep. output ($p=0.95; \tau_p=1.5$)	37.0	**50.3**	35.1	45.5	46.4	31.5
AutoAD-Zero [Xie et al., 2024a]	Joint output	31.6	46.4	33.8	43.2	46.8	30.1
Ours	Joint output	**38.4**	49.2	**35.7**	**51.3**	**47.3**	**31.8**

The bottom-right example highlights the effectiveness of scale-dependent Stage I factor formulation. AutoAD-Zero, designed to query characters, actions, interactions, and facial expressions, sometimes overlooks environmental details. Our method, in contrast, correctly identifies the shot as a long shot and instructs the VideoLLM to incorporate environmental context, resulting in more accurate scene descriptions. For more visualisations, please refer to the Appendix.

7.5.8 Assisted AD generation

The subjective nature of AD sets a practical limit on metric scores, lower than the theoretical maximum, because human annotators often provide different but valid descriptions. Therefore, beyond enforcing generating a *single* AD sentence, we also consider employing our framework as an *assistant* to produce *multiple* candidate AD sentences.

To standardise such a protocol, we consider five candidate ADs generated by an *assistant* and employ an *expert* to select the best one. To effectively benchmark performance against existing GT ADs, we define the *"expert"* as an automatic selection mechanism that chooses the candidate with the highest average CIDEr and action score.

To develop an AD generation assistant, we fix the Stage I dense descriptions and explore generating multiple candidates in Stage II. This can be achieved by either running Stage II independently five times (termed the "independent output" setup) or generating five AD outputs simultaneously within a single run (termed the "joint output" setup). As observed in Table 7.9, the assistant-based setup significantly improves upon the single AD performance, highlighting the potential

of training-free methods in effectively capturing the desired content for AD generation. Within the "independent output" setup, increasing the randomness of sampling (*i.e.* higher p and τ_p) enhances the quality of the selected AD, owing to greater candidate diversity. Meanwhile, the "joint output" setup achieves superior performance, which could be attributed to reduced information redundancy across the simultaneously generated ADs. For additional visualisations, discussions, and detailed text prompts, please refer to the Appendix.

7.6 Discussion – summary and limitations

We have demonstrated the benefit of shot-based context and film grammar awareness in AD generation – our training-free two-stage framework achieves state-of-the-art performance among all training-free counterparts, even surpassing fine-tuned models on multiple benchmarks.

The current framework has two main limitations: (i) the performance depends on the base VideoLLM, which may occasionally hallucinate details inconsistent with the visual content; and (ii) story-level context is not incorporated into the AD generation process. These limitations could potentially be addressed in future work by improving visual grounding and extending the visual and textual context to include the plot.

Acknowledgements. This research is supported by the UK EPSRC Programme Grant Visual AI (EP/T028572/1), a Royal Society Research Professorship RSRP\R\241003, a Clarendon Scholarship, and ANR-21-CE23-0003-01 CorVis.

Appendices

Appendices for this chapter can be found in the online version of the paper.[1]

Statement of authorship

A statement of authorship for this paper is provided in Appendix A.

[1] https://www.robots.ox.ac.uk/~vgg/research/shot-by-shot/

Chapter 8

As the Story Unfolds: Watching a Film and Identifying Characters as a Human Does

This paper was submitted to the Advances in Neural Information Processing Systems (NeurIPS), 2026. Under review.

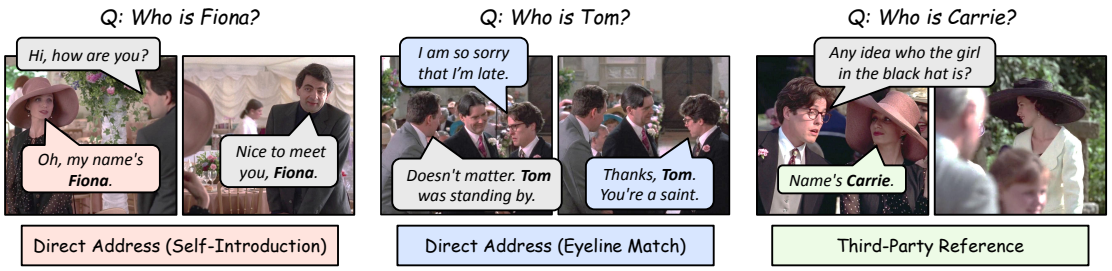


Figure 8.1: **Character naming task.** We associate character names to their facial appearances in movies, mirroring how viewers infer who is who while watching. Characters may be introduced through diverse cues, including self-introduction, direct address, and third-party reference. We show three representative examples from *Four Weddings and a Funeral* (1994). Correctly linking a character to their name is central to movie understanding, enabling coherent character tracking across scenes as the story unfolds.

Abstract

Films introduce characters incrementally as they progress. We study the problem of identifying principal characters and assigning their names directly from the movie itself, without using external cast lists, actor photographs, or pre-built character banks. We introduce the character naming task: given a name mentioned in dialogue and the associated video context, the model must determine which visible character bears that name. This task is challenging because name resolution often depends on cinematic cues such as shot–reverse-shot structure, self-introduction, direct address, third-party reference, gaze, and temporal continuity. We provide a dataset and annotations for evaluating character naming and first character appearances, develop a video-and-dialogue model for the task, and compare it with proprietary multimodal large language models. We further introduce a progressive framework for building a character bank as the film unfolds. Together, these contributions support online, human-like movie understanding in which character identities are inferred from the film itself.

8.1 Introduction

The objective of this paper is to watch a film as a human would – ingesting the film frame-by-frame in an online mode, and identifying the principal characters and their names *as the story unfolds*. Traditional films, such as a romantic comedy, consist of characters and their story arcs. In order for the viewer to quickly understand who these principal characters are, the film must introduce and name

them towards the start of the film. This is achieved by a combination of cinematic devices (such as the camera following the person, and close-ups) and using the dialogue or other information to name them, as illustrated in Figure 8.1.

Obtaining character names explicitly only from the film is in stark contrast to the methods employed recently in the literature for tasks like generating audio descriptions and transcripts [Pini et al., 2019; Han et al., 2023b; Han et al., 2023a; Han et al., 2024; Xie et al., 2024a; Xie et al., 2025; Gui et al., 2025]. In these works, a *character bank* is first constructed using a cast list and actor photos from IMDB, Fandom and other online sources. The character bank is then used to label the principal characters throughout the film using face recognition (or voice recognition for animated films [Gui et al., 2025]).

To operationalize our goal, we first investigate how the viewer is presented with the name of a character as they watch the film – almost always it is from the dialogue, and very occasionally from other sources, such as name badges. This enables the *character naming* task to be automated and evaluated – given a name in the dialogue, the associated shots can be used to ask the question: which of the people appearing in this clip has this name? To this end, the first contribution of this paper is to build a dataset from MovieNet [Huang et al., 2020] and Movie101 [Yue et al., 2023] films that can be used to train and evaluate this task. We develop a method to solve this task, using the video and dialogue, and compare its performance to proprietary Multimodal LLMs (MLLMs), such as Gemini [Google DeepMind, 2025]. This is a challenging task, as correct inference often requires an understanding of film editing style (as the person addressed by name may be shown in a different shot) and also character gaze and eyeline (to know who they are looking at when saying a name).

This establishes that characters can be named in films, using the information presented by the screenwriter and director. Often, the name of a principal character is presented multiple times across different dialogues throughout the film, providing opportunities for error correction (by both human and machine) if the first presentation had been missed or incorrectly assigned. Once the principal characters are named (possibly multiple times), they can be recognized throughout the rest of the movie, and retrospectively named from their first appearance (using human or machine memory). To this end, we introduce two further tasks: (i) annotating

the movies with the first appearance of the principal characters (which may well be before they are named in the dialogue); and (ii) progressively building up a *character bank* as the film progresses. The character bank records the appearance, voice and names of the principal characters (as in previous work). We provide annotations to evaluate the character first appearance over a number of movies. For example, in the movie *Juno*, the protagonist *Juno* is introduced in the first shot of the film, and appears in almost every scene, but is not named until 7.2 minutes into the film.

In summary, we make the following four contributions: (i) we introduce the new task of *character naming* from dialogue alone, a key enabling step for watching and understanding a film as a human does; (ii) we provide the Movie Character Naming (MCN) dataset with 23,792 instances of multiple-choice character naming over 407 movies, and also 20 movies with the first appearance of the principal characters annotated; (iii) we develop a character naming model using open-source MLLMs, and compare its performance to proprietary MLLMs, analysing what is difficult to solve; and (iv) we introduce the first automatic method to progressively build up a character bank for a feature-length movie. These capabilities put in place the foundation for understanding character arcs and, ultimately, the story of a film. All annotations and code will be released.

8.2 Movie character naming (MCN) dataset

In this section, we introduce the Movie Character Naming (MCN) Dataset, which provides annotations for two complementary tasks: *Character Naming* and *First Occurrence*. The *Character Naming* annotations consist of video segments in which character names are uttered in the dialogue, each paired with ground-truth name-to-track mappings. The *First Occurrence* annotations, by contrast, are designed to evaluate the streaming pipeline for character bank construction, and provide two (possibly coinciding) timestamps per principal character: the moment they first appear on screen, and the moment their name is first revealed.

Figure 8.2 shows a representative MCN sample for the character *Carrie* from *Four Weddings and a Funeral*. The movie timeline is annotated with *Character Naming* and *First Occurrence* labels. Additional examples are provided in the Appendix.

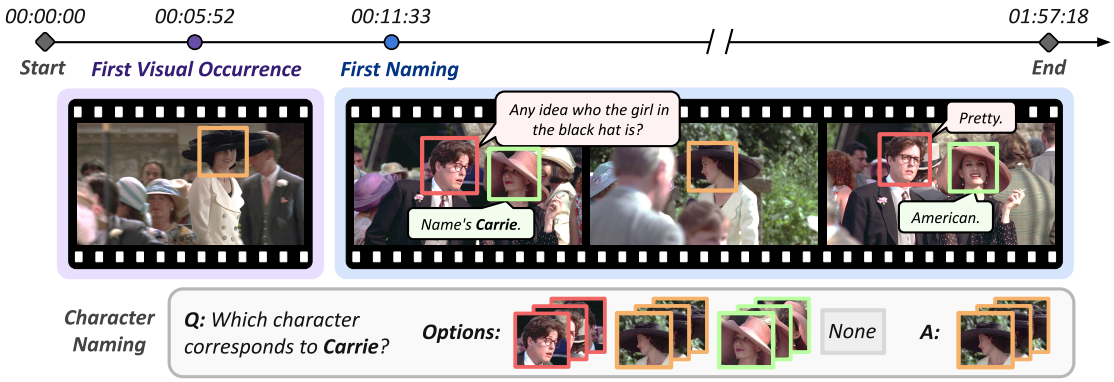


Figure 8.2: **Example annotations from the MCN dataset.** The MCN dataset provides manually annotated timestamps for each character's first visual occurrence and first naming, along with ground-truth name-to-face correspondences for the *Character Naming* task.

8.2.1 Data collection and curation

We build MCN on top of MovieNet [Huang et al., 2020] and Movie101 [Yue et al., 2023], which already provide ground-truth mappings between character names and body bounding-box coordinates.

Annotations for Character Naming (CN). We construct the associations between character names and face tracks in three steps. *First,* since MovieNet and Movie101 already provide ground-truth name–body associations, we align character body boxes with their face tracks in space and time, yielding name–face associations for *all* characters appearing on screen. *Second,* we keep only those associations whose corresponding names are also mentioned in the dialogue, since these are the only ones inferable from the audio–visual signal we use. *Third,* we handle the case where a name is mentioned in the dialogue but the corresponding face track is unavailable (*e.g.* when the character is referred to in absentia). In this case, we label the association as none, since no in-clip cues support the assignment. As a result, we provide the final name–face associations as ground-truth labels, along with all character-level face tracks in the video clip as candidate options, to construct a multiple-choice instance.

Annotations for First Occurrence (FO). For every character in the movie, we manually annotate two timestamps: the *first visual occurrence*, *i.e.* the moment the character first appears on screen, and the *first naming*, *i.e.* the moment at which a human viewer can first associate the character's visual appearance with their name based solely on the preceding content.

Table 8.1: **Statistics of the character naming (CN) and first occurrence (FO) annotations.** Each question corresponds to a name-to-track mapping query or a first-occurrence timestamp annotation.

Task	*Split*	*# Movies*	*Duration (h)*	*# Characters*	*# Questions*
CN	Train	377	108.13	1853	22210/21150
CN	Test	30	6.51	226	1582/1399
FO	Test	20	39.98	287	574

Quality Control. To ensure a high-quality test set, we manually inspect every test sample and correct errors in face-track clustering and name labelling. We then run state-of-the-art proprietary models on the curated set and examine cases where their outputs disagree with the annotation, fixing any remaining errors (or confirming the annotation is correct but the model fails).

8.2.2 Dataset statistics

The statistics of both annotation types in the MCN dataset are summarised in Table 8.1. For the CN task, the training splits comprise clips from 377 training movies (108.13 hours), covering 1,853 characters and 22,210 name-to-track questions, of which 21,150 have non-none answers. The training movies are disjoint from those in both the CN and FO test sets, ensuring no movie-level leakage across splits. The CN test split contains clips from 30 movies (6.51 hours), covering 226 characters and 1,582 manually verified questions, of which 1,399 have non-none answers. In particular, the CN questions in the test set can be broken down by evidence type: approximately 12% involve self-introduction, 70% involve direct address by other characters, 16% involve third-party reference, and 2% require reasoning from textual cues in the movie, such as name badges.

The FO annotations form a test set of 20 movies (39.98 hours) with 287 annotated characters, yielding 574 timestamp annotations in total. Each character has two annotations, corresponding to the first visual occurrence and the first naming. Together, the MCN dataset spans a diverse range of genres, casts, and narrative styles, ensuring that our evaluation captures performance across varied cinematic settings.

8.2.3 Preprocessing – character-level face-tracks and associated speech

To obtain multiple-choice instances for *Character Naming*, we need to extract character-level tracks and speech from video clips around name occurrences in the dialogue, therefore enabling us to ask models to select one of them for each query name. This involves extracting a face-track cluster f_i for each character and linking it with the corresponding voice-track cluster v_j. Note, this pre-processing is done at the clip level when annotating the dataset, but results are also made available over the entire movie for downstream tasks (*e.g.* character bank construction).

Face Track Extraction. For each recurring character i, we detect and track faces across shots and group the tracks into a character-level face-track cluster f_i. The detailed procedure is given in the Appendix.

Linking Voices to Face Tracks. For the clean voice segments in the video, we first apply speaker diarization to group them into voice clusters, with each cluster denoted v_j. To match each voice cluster to an existing face track (*i.e.* to identify who is speaking on screen), we apply TalkNet [Tao et al., 2021] to obtain synchronization scores $s_{i,j}$ between faces f_i and voices v_j. Each voice cluster is then assigned to the face-track cluster with the highest synchronization score, yielding complete character identities with paired faces and voices.

As a further preprocessing step, we transcribe the voice tracks into textual dialogue with Parakeet [Rekesh et al., 2023]. Since each voice track is linked to a face cluster, this naturally yields visually grounded subtitles. Finally, the character names n emerge as a natural by-product of subtitle extraction. We retain clips whose subtitles contain at least one character name, passing them to the subsequent character naming step.

8.3 Assigning names to characters

The task of assigning names to characters, *i.e.* Character Naming (CN), is performed naturally by humans while watching a film, but is not trivial for deep learning systems owing to multiple challenges: first, a character may be addressed directly by name, or referred to in the third person while off-screen, or revealed

only through what others say about them; second, who is being addressed may only be indicated by gaze or by the editing structure, for example, as one character says a name, and the named character then reacts in the subsequent shot. We therefore treat CN as a new multimodal task that requires joint reasoning over textual cues (*e.g.* conversational logic, vocative address) and visual cues (*e.g.* eye contact, on-screen presence).

Formally, given a video clip V containing a set of face-track clusters $\{f_i\}$ (each optionally linked to a voice cluster v_j, as described in Section 8.2.3) and a candidate name n extracted from the dialogue, the Character Naming (CN) task is to predict an assignment $\psi(n) \in \{f_i\} \cup \{\varnothing\}$. Here, $\psi(n) = f_i$ indicates that the name n refers to the character represented by face-track cluster f_i, while $\psi(n) = \varnothing$ indicates that none of the visible characters corresponds to n (*e.g.* the named character is off-screen).

In this section, we first describe how we operationalize Character Naming for MLLMs through visual prompting alone (Section 8.3.1). We then introduce a fine-tuning recipe that improves the MLLMs' reasoning on this task, enabling more reliable assignments of character names (Section 8.3.2).

8.3.1 MLLM-based character naming

To adapt MLLMs for character naming in a zero-shot manner, we use the following prompting strategies.

Prompting the MLLM with visual identities. Rather than providing the MLLM with bounding-box coordinates of the face tracks, we adopt the color-based visual prompting scheme of Xie et al. [2024a]. Each face track is assigned a unique color, and its bounding boxes are drawn in that color on the raw video frames. The boxes are then referred to by color in the text prompt, allowing the MLLM to associate identities with their spatial locations.

Prompting the MLLM with visually grounded subtitles. Subtitles are embedded directly into the text prompt, with each speaker referred to by the color tag of their associated face track. We additionally include timestamps to enable more temporally accurate inference. For example, a line spoken by the character

with blue face bounding boxes would be formatted as `<693.2s> [BLUE] “Any idea who the girl in the black hat is?” <695.0s>`.

Prompting the MLLM for character naming. Given the visual identities and visually grounded subtitles, we directly ask the MLLM to match each character name from the subtitles with a color, *e.g.* “`Which color corresponds to John Constantine?`” The color code thus bridges each visual identity’s spatial position to its corresponding name.

8.3.2 Fine-tuning the MLLM for character naming

Following the prompting strategies described above, we adopt a two-stage R1-style fine-tuning recipe [Guo et al., 2025] to further improve the naming accuracy.

Stage I: Supervised fine-tuning (SFT). The first stage is a cold start that supervises the MLLM with synthetic reasoning traces. These traces are generated by a teacher MLLM, which is given the video frames, visually grounded subtitles, and ground-truth name-to-color mapping, and asked to produce a reasoning trace that leads to the correct answer. This teaches the student MLLM to follow the desired output format (*i.e.* a reasoning trace followed by the answer) before the reinforcement learning stage.

Stage II: Reinforcement learning. The second stage applies GRPO [Guo et al., 2025] with a composite reward $R = r_f + r_a$, where r_f rewards adherence to the expected response structure and r_a measures the correctness of the predicted name–face pairs. Letting $\hat{i}(n)$ and $i^*(n)$ denote the face track assigned to name n under the prediction $\hat{\psi}$ and the ground truth ψ^*, we set $r_a = \sum_{n \in N} \mathbf{1}[\hat{i}(n) = i^*(n)]$. The GRPO advantage and policy update follow the standard formulation [Guo et al., 2025], summarized in the Appendix.

A naive application of GRPO to this task suffers from a subtle shortcut: the model can latch onto color-specific cues. For instance, it may always associate the protagonist with “red” simply because “red” is the most frequently used color in the training distribution. We address this with a simple but effective *color-permutation augmentation*: during RL training, we randomly permute the color assignment for each sample. The ground-truth name–face mapping is invariant under such

permutations, so consistent rewards across permuted variants are achievable only by reasoning about identity itself. This regularizes the policy toward color-agnostic reasoning and, empirically, accounts for a significant share of our gains over the cold-start baseline.

8.4 Streaming pipeline for character bank construction

Given a full movie, our goal is to construct a character bank in an online manner, taking the shot as the elementary unit. The *character bank* $\mathcal{B}_t$ serves as a memory of all characters observed so far. At each shot t, it aggregates the corresponding character information as triplets (f, n, v), where f is a face-track cluster, n a character name, and v a voice cluster.

Formally, the bank is updated auto-regressively (as illustrated in Figure 8.3):

$$\mathcal{B}_t = \phi_{\mathcal{B}}(\mathcal{B}_{t-1}, \psi(V_t)), \tag{8.1}$$

where V_t denotes the video clip at shot t, and ψ is the MLLM fine-tuned for CN, which outputs character names and corresponding face tracks and voices $\{(n, f, v)\}$. We also retain identities with no assigned name, *i.e.* $\{(\varnothing, f, v)\}$, mirroring how a human viewer can visually remember a character even before they are introduced by name. These triplets are then merged into the previous bank by the update operator $\phi_{\mathcal{B}}$.

Overall, this streaming pipeline maintains and auto-regressively updates the character bank $\mathcal{B}$. We detail the bank update process in Section 8.4.1 and introduce an error-correction mechanism in Section 8.4.2 that keeps the update robust to error accumulation over multiple steps.

8.4.1 Character bank update

Given the names and corresponding face and voice tracks $\{(n, f, v)\}$ extracted from a new shot (with n possibly empty), we merge them into the existing entries of the previous bank in two steps.

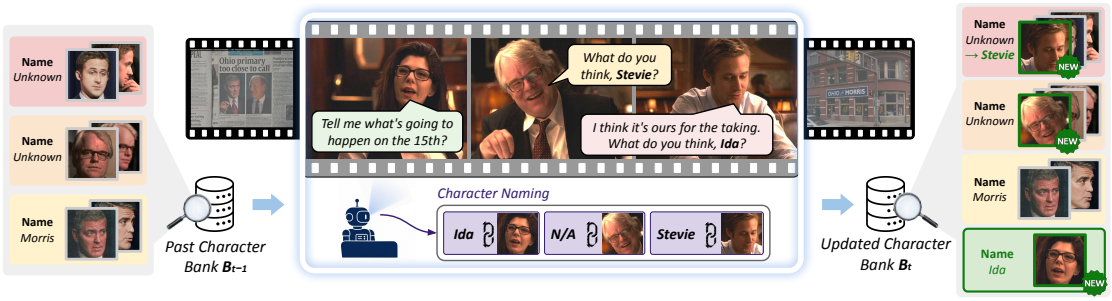


Figure 8.3: **Streaming pipeline for character bank construction.** As new shots arrive, we extract observations (character tracks, subtitles, and mentioned names) and reason jointly over them to link face tracks to names. Each step incrementally updates the bank through three operations: *naming* an existing cluster (*Stevie*, first cluster), *enriching* a cluster with additional face exemplars (second cluster), and *instantiating* a new entry for a previously unseen character (fourth cluster). The sample is drawn from *The Ides of March* (2011).

Track Assignment. We treat face tracks as the primary signal and assign each f_i to an existing face cluster in the bank. We extract the corresponding face embedding e_i and find the closest cluster by distance to its embedding centroid $\bar{e}_b$ as in Equation 8.2. A smaller distance indicates greater similarity between the face embedding e_i and the cluster centroid $\bar{e}_b$.

$$b^* = \arg\min_{b \in \mathcal{B}_{t-1}} d_{\cos}(e_i, \bar{e}_b). \tag{8.2}$$

If $d_{\cos}(e_i, \bar{e}_{b^*}) < \tau_f$, the track is assigned to entry b^*. Otherwise, we fall back to voice as the secondary signal: if the shot's speaker embedding u_i matches an entry's voice centroid $\bar{u}_b$ under a tighter distance threshold $\tau_v < \tau_f$, the track is assigned there. If neither threshold is satisfied, a new entry is instantiated in the bank. Upon assignment, the matched centroids are updated.

Name Binding. We then bind a name to each entry in the character bank. For each entry, we maintain a histogram over the names associated with its face tracks across all observed shots, and use the majority-voted name as its canonical name. This handles aliases and nicknames; for example, in *Juno*, the character *Juno* may be addressed as *Mama Bear*, *Home Skillet*, or *General MacGuff*. Majority voting helps recover the most representative name for each character. It also improves robustness to noisy predictions from the character naming module, since occasional erroneous name assignments are unlikely to dominate the histogram.

Retrospective assignment is a special case of name binding: when the CN module assigns a name to a previously unnamed face-track cluster, the new name is added to that cluster's histogram with a count of one, and the canonical name is updated accordingly. Thus, clusters that are initially unnamed can still acquire names once sufficient naming evidence becomes available in later shots.

8.4.2 Error correction during character bank construction

The Markov structure of the update process $\phi_{\mathcal{B}}$ makes it susceptible to error accumulation from upstream components (*e.g.* transcription, active-speaker detection, and identity grounding). To mitigate this, we introduce three error correction mechanisms deployed along the streaming chain. First, we use an MLLM to filter out noisy or ambiguous face tracks before clustering, ensuring high-quality embeddings. Second, we use an LLM to filter out query names from the subtitles that are unlikely to refer to on-screen characters. Third, we resolve cross-entry name conflicts after each shot using a greedy voting scheme that enforces one-to-one name assignment. Together, these mechanisms address distinct error modes and stabilize the evolution of $\mathcal{B}_t$ without revisiting past observations. We refer the reader to the Appendix.

8.5 Experiments

This section presents our evaluation. We first introduce the metrics (Section 8.5.1), then evaluate each task. Section 8.5.2 evaluates our character naming (CN) module on the MCN test set against state-of-the-art open- and closed-source models. Section 8.5.3 evaluates the full streaming character bank pipeline, measuring how well our method tracks each character's first visual appearance and the first moment it can be linked to a name, and compares the final character bank against the IMDB ground truth. We ablate key design choices, including the two-stage fine-tuning strategy and data augmentation for CN, and the error correction mechanisms for the character bank. Implementation details and ablations are provided in the Appendix.

8.5.1 Evaluation metrics

Character Naming Accuracy. We evaluate for two settings. *Overall accuracy* is computed over all name queries, including those labelled `none` when the mentioned character cannot be linked to any on-screen face track. *Known accuracy* is computed only over queries whose ground-truth label is a valid face track, thereby examining the model's ability to correctly ground on-screen characters.

Character Bank Mean Temporal IoU. To evaluate the character bank streaming pipeline, we propose a *Mean Temporal IoU* (mTIoU) metric that measures how closely the bank tracks ground-truth human knowledge of each character over time.

For a movie of duration T and character c, let $H_c \subseteq [0, T]$ denote the interval during which a human viewer is aware of c, and $P_c \subseteq [0, T]$ the interval during which the model correctly identifies c in the bank. We define $\mathrm{TIoU}(c) = |H_c \cap P_c|/|H_c \cup P_c|$, which penalizes both *late* identification and *incorrect* identification (*e.g.* an entry later overwritten with a wrong name). To reflect narrative importance, we weight each character by their subtitle mention frequency w_c and report $\mathrm{mTIoU} = \sum_c w_c \,\mathrm{TIoU}(c) / \sum_c w_c$.

We instantiate two variants from the MCN timestamps: the *visual* variant mTIoU_v uses $H_c = [t_c^v, T]$ with t_c^v the first visual occurrence of c, and the *name* variant mTIoU_n uses $H_c = [t_c^n, T]$ with t_c^n the first naming timestamp (when a human can first link a face to a name).

Principal Character Recall (PCR@K). To evaluate the quality of the final character bank, we propose *Principal Character Recall* at K (PCR@K), measuring how reliably the pipeline recovers a movie's most important characters with both correct names and correct visual exemplars.

For each movie, we take the top-K principals from IMDB, ranked by their narrative importance. A ground-truth principal c is *recovered* iff some single entry $b \in \mathcal{B}$ satisfies both of the following conditions: (i) the canonical name of b phonetically matches the IMDB name of c, and (ii) the distance between its face centroid $\bar{e}_b$ and the IMDB reference embedding $\bar{e}_c^*$ is below the threshold τ_{eval}. We report per-movie recall as the fraction of recovered principals and average across movies.

Table 8.2: **Character naming accuracy on the MCN test set.** We report overall accuracy (including none labels) and known accuracy (restricted to queries with a valid face track). Our fine-tuned Qwen3VL-4B-Thinking closes most of the gap to proprietary frontier models despite being substantially smaller in terms of model size.

Model	*Propriet.*	*Fine-Tuned*	*Overall Acc.*	*Known Acc.*
Random Baseline	×	×	29.90	33.74
Gemini-3-Flash-Preview	✓	×	79.57	86.62
Gemini-3.1-Pro-Preview	✓	×	**82.47**	**89.69**
Qwen3VL-4B-Thinking	×	×	52.15	56.68
Qwen3VL-8B-Instruct	×	×	53.60	58.83
Qwen3.5-9B	×	×	61.57	67.69
Qwen3VL-8B-Instruct (FT)	×	✓	64.92	71.77
Qwen3VL-4B-Thinking (SFT + RL)	×	✓	**72.76**	**81.06**

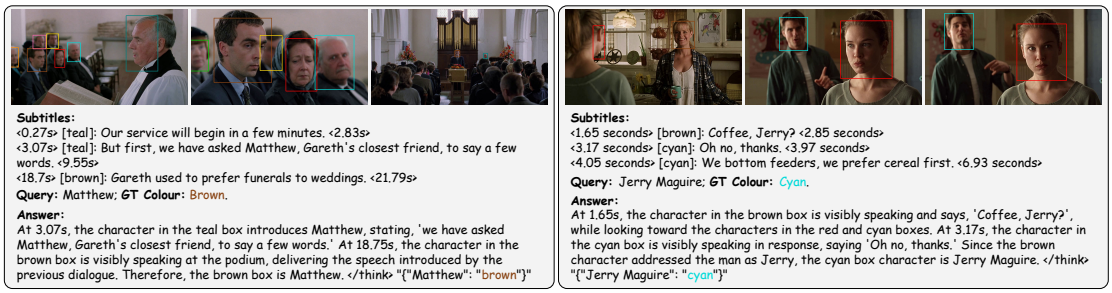


Figure 8.4: **Qualitative results for character naming.** Examples are from *Four Weddings and a Funeral* and *Jerry Maguire*. In each case, our default "Qwen3VL-4B-Thinking (SFT + RL)" model reasons jointly over subtitles and video frames to link a queried name to the correct face — in the first, by following a verbal introduction to a later podium shot; in the second, by resolving direct address across a shot–reverse-shot exchange.

8.5.2 Results on character naming

We evaluate Character Naming (CN) with three sets of methods. First, we compare against several baselines, including a random baseline that uniformly selects from the candidate colors, non-proprietary open-source MLLMs (Qwen3VL-4B-Thinking, Qwen3VL-8B-Instruct, and Qwen3.5-9B [Bai et al., 2025]), and a naive fine-tuning baseline obtained by directly fine-tuning Qwen3VL-8B-Instruct on the MCN training set. Second, we report the result of our CN model, which is obtained by two-stage R1-style fine-tuning of Qwen3VL-4B-Thinking on the MCN training set. Third, we compare against state-of-the-art proprietary MLLMs: Gemini-3-Flash-Preview and Gemini-3.1-Pro-Preview [Google DeepMind, 2025]. All models receive the same inputs with a visually prompted video clip, visually grounded subtitles, and queried character names, and are asked to output the corresponding box colors.

Table 8.3: **Results on the MCN first occurrence dataset.** We report mean Temporal IoU (mTIoU) against two reference intervals, first visual occurrence and first naming, and Principal Character Recall (PCR@K) on the final character bank for $K \in \{5, 10, 15\}$. Our method operates in a streaming setting, whereas proprietary baselines are given full-movie access.

Method	*Streaming*	$mTIoU_v$	$mTIoU_n$	*PCR@5*	*PCR@10*	*PCR@15*
Gemini-3-Flash-Preview	×	51.81	30.92	35.00	32.04	29.84
Gemini-3.1-Pro-Preview	×	49.05	30.10	44.00	37.15	34.93
Ours	✓	**84.90**	**54.40**	**63.00**	**54.54**	**48.19**

Results. We report Character Naming results in Table 8.2. The pretrained Qwen3VL-8B-Instruct achieves 53.60% overall accuracy, improving to 64.92% via naive fine-tuning. In contrast, our two-stage SFT+GRPO approach with *color-permutation augmentation*, applied to a smaller 4B-Thinking backbone, yields a +7.84% gain over the naive 8B baseline. Despite the smaller model size, our approach closes most of the gap to Gemini-3.1-Pro-Preview and clearly outperforms all open-source alternatives. Qualitative examples are shown in Figure 8.4.

Discussion. Compared with open-source models such as Qwen3.5-9B, our two-stage fine-tuned model more accurately names characters in direct address scenarios (*e.g.* shot–reverse-shot, telephone conversations), better exploits visual cues (*e.g.* pointing, handshakes), and improves temporal grounding between subtitles and frames. A qualitative comparison is provided in the Appendix.

Nevertheless, a gap remains to the strongest proprietary model, likely driven by errors in transcription and active-speaker detection. We hypothesise that models such as Gemini [Google DeepMind, 2025] can partially recover from these errors by leveraging lip motion, whereas our model is more sensitive to such upstream noise.

There are also cases where both Gemini [Google DeepMind, 2025] and our model fail. In particular, both struggle with (i) subtle cues (*e.g.* brief pointing among multiple characters), (ii) complex scenes with motion, and (iii) over-predicting on-screen characters when the ground-truth character is absent. We refer readers to the Appendix.

8.5.3 Results on streaming character bank construction

Baseline. For streaming character bank construction, we use Gemini-3-Flash-Preview and Gemini-3.1-Pro-Preview [Google DeepMind, 2025] as baselines, prompting them to output each character's name, first visual occurrence timestamp, and first naming timestamp. To obtain face clusters, we ask the models to identify timestamps with close-up or clear solo shots, from which we perform face detection, extract features, and cluster faces, selecting the largest connected component as the predicted cluster. As these models are not designed to directly output face crops or clusters, we consider a prediction correct if any face in the cluster matches the ground-truth identity.

Results. As shown in Table 8.3, our pipeline substantially outperforms both proprietary baselines across all metrics, despite operating in the more challenging streaming setting. These improvements reflect more accurate identification of when characters first appear and are named (mTIoU), as well as stronger alignment of the final character bank with canonical identities from external sources (PCR). These gains are especially notable given that the baselines have full-movie access while ours processes the video in a single streaming pass.

Discussion. Our pipeline performs well on movies such as *Juno* and *Meet the Parents*, which feature relatively simple scenes with few participants and a limited set of recurring characters. In these cases, identity clustering is generally reliable, aided by our face-track filtering. Even when Character Naming errors occur, our method can often recover as additional evidence emerges over time.

Performance degrades, however, in movies with many scenes containing numerous minor or background characters. In *Harry Potter and the Goblet of Fire*, for example, background characters introduce noise into clustering, which distracts the Character Naming process. Movies with very large named casts, such as *Four Weddings and a Funeral*, with over 50 named characters among hundreds of recurring faces, are also particularly challenging, as the model must maintain and disambiguate many identities over long temporal contexts.

8.6 Related work

Character-Centric Movie Understanding. Movie understanding has emerged as a milestone task for video understanding, owing to its long temporal span and rich multimodal narrative structure, motivating a variety of datasets [Huang et al., 2020; Tapaswi et al., 2016; Bain et al., 2020; Soldan et al., 2022; Vicol et al., 2018]. A representative line of research is *Audio Description (AD) generation* [Rohrbach et al., 2015; Han et al., 2023b; Han et al., 2023a; Han et al., 2024; Xie et al., 2024a; Xie et al., 2025; Khandelwal et al., 2025; Gui et al., 2025], where Han et al. [2023a] identifies character recognition as central to the task. Existing work typically takes one of two shortcuts: relying on curated sources like IMDB cast lists [Pini et al., 2019; Han et al., 2023b; Han et al., 2023a; Han et al., 2024; Xie et al., 2024a; Xie et al., 2025] or fan wikis [Gui et al., 2025], which are often incomplete or unavailable for new films, or sidestepping naming entirely with placeholders like "Person A" [Fu et al., 2025; Zhou et al., 2025d; Song et al., 2024a]. A separate line of work links names to faces directly from video. Early approaches align shooting scripts with subtitles for weak labels and use active-speaker detection to associate them with face tracks [Everingham et al., 2006; Sivic et al., 2009]. Extensions add probabilistic models over face, speech, and clothing cues [Tapaswi et al., 2012], coreference resolution [Ramanathan et al., 2014], and script-free dialogue-based naming for recurring TV casts [Haurilet et al., 2016; Nagrani and Zisserman, 2018; Cour et al., 2010]. A complementary line assumes identities are known and analyses interactions via gaze and eyelines [Marin-Jimenez et al., 2014; Marin-Jimenez et al., 2020; Gu et al., 2018; Recasens et al., 2017]. We revisit this problem in the era of MLLMs. Prior dialogue-based approaches, including Haurilet et al. [2016], cast naming as an optimisation over many mentions across clips or episodes of a recurring TV cast; we instead resolve identity from the single clip surrounding the name as a human viewer does, generalising to feature-length movies in the wild where names are typically mined from one-off dialogue. In concurrent work, Pu et al. [2026] track identities via textual appearance descriptions, a lossy representation that fails to distinguish visually similar characters and degrades under costume or lighting changes. Our character bank instead grounds each identity in visual exemplars, retaining the fine-grained cues text abstracts away.

Visual Reasoning in Vision-Language Models (MLLMs). Recent work ex-

tends reasoning from Large Language Models (LLMs) [Wei et al., 2022; Kojima et al., 2022] to the multimodal setting. Early approaches prompt frozen MLLMs to produce intermediate reasoning steps [Wang et al., 2025c; Mitra et al., 2024], while more recent methods train explicit reasoning via reinforcement learning. Inspired by [Guo et al., 2025], Group Relative Policy Optimization (GRPO) has been adapted to MLLMs, showing that RL with verifiable rewards improves reasoning ability [Shen et al., 2025; Huang et al., 2025b; Zhou et al., 2025b; Meng et al., 2025], including across perception tasks such as detection [Liu et al., 2025b], segmentation [Liu et al., 2025a], and fine-grained perception [Yu et al., 2025]. For video understanding specifically, prior work incorporates temporal reasoning into this paradigm, *e.g.* via temporal-aware GRPO [Feng et al., 2025a], joint spatio-temporal reasoning [Li et al., 2025a], and long-form temporal grounding [Wang et al., 2025d]. In parallel, visual prompting introduces structured cues, such as colored masks [Yang et al., 2023b], boxes [Shtedritski et al., 2023; Cai et al., 2024b], region tokens [Chen et al., 2023b; You et al., 2023], or identity markers [Xie et al., 2024a], to link visual entities with language. Our approach builds on both lines of work, combining color-based visual prompting with a two-stage SFT + GRPO fine-tuning strategy to enable an MLLM to reason about character identity from visual and dialogue cues jointly.

Person Clustering in Long Videos. Grouping person appearances by identity is a long-standing problem in video understanding, with applications to cast listing, character-based browsing, and story understanding [Fitzgibbon and Zisserman, 2002; Cinbis et al., 2011; Everingham et al., 2006; Tapaswi et al., 2019]. Early work focuses on *face clustering*, using metric or representation learning to group face tracks under variations in pose, lighting, and expression [Sharma et al., 2019; Wang et al., 2019c; Yang et al., 2019a; Shen et al., 2021]. These methods exploit spatio-temporal structure in edited videos via co-occurrence constraints [Bauml et al., 2013; Kalogeiton and Zisserman, 2020; Wu et al., 2013], or apply nearest-neighbor clustering over pretrained embeddings [Sarfraz et al., 2019; Otto et al., 2017]. However, they rely on visible faces, limiting performance when characters are occluded, off-screen, or viewed from behind. To address this, Brown et al. [2021] introduce Video Person-Clustering, grouping face, body, and voice cues via multimodal hierarchical clustering, with later work extending this direction using

graph-based formulations [Nagrani et al., 2018]. Our pipeline follows a similar multimodal approach, using face embeddings as the primary signal with voice as fallback. Unlike prior work, we operate in a *streaming* setting, where clusters are instantiated, updated, and named as the film unfolds, extending online clustering [Aggarwal et al., 2003; Cao et al., 2006] to feature-length videos.

8.7 Discussion

In this paper, we introduce the task of *Character Naming* from dialogue and close the gap between open-source models and the best proprietary models. Building on this task, we further introduce a streaming pipeline to construct a *character bank*, enabling online, human-like movie understanding. For both tasks, we compare our method with Gemini [Google DeepMind, 2025], though neither task is fully solved, even by proprietary models. A natural question is whether this comparison is fully fair, given that Gemini may have been pre-trained on some movies in our benchmark. Notably, despite being explicitly instructed not to use external knowledge, Gemini can still correctly identify characters even when the video evidence is insufficient, suggesting a natural advantage on this task. Our own model shows substantially less of this behaviour. In future work, we will examine the performance gap on newly released or otherwise unseen movies, where we expect it to narrow further.

Appendices

Appendices for this chapter can be found in the online version of the paper.[1]

Statement of authorship

A statement of authorship for this paper is provided in Appendix A.

[1] https://www.robots.ox.ac.uk/~vgg/research/story_unfolds/

Chapter 9

Discussion

In this chapter, we first summarise the achievements and the impact of the presented works in this thesis (Section 9.1), and then briefly suggest directions for future work (Section 9.2).

9.1 Achievement and impact

9.1.1 Moving object segmentation

OCLR with Appearance-Based Refinement. In Chapter 2, we presented *Segmenting Moving Objects via an Object-Centric Layered Representation (OCLR)*, an early approach to multi-object motion segmentation that does not rely on human annotations. The work has received over 75 citations, and the released code[1] has served as a reference implementation for subsequent methods. Notable examples include Segment Any Motion in Videos [Huang et al., 2025a], which replaces OCLR's optical flow motion cue with long-range point trajectories to better handle deformation and occlusion, and VideoCutLER [Wang et al., 2023b], which replaces synthetic optical flow supervision with RGB video compositing to learn multi-object segmentation. Chapter 3 extended OCLR with an appearance-based refinement framework, demonstrating that appearance cues can be used to select and correct flow-based object proposals. Together, the two chapters introduced the *DAVIS17-motion* and *YTVOS18-motion* evaluation protocols, extending the established DAVIS17 and YouTubeVOS18 benchmarks to the setting of multi-

[1] https://github.com/Jyxarthur/OCLR_model

object motion segmentation. These protocols have subsequently been adopted by a range of motion segmentation methods, including Multi-Object Discovery by Low-Dimensional Object Motion [Safadoust and Güney, 2023], which segments multiple objects from a single image by reconstructing optical flow from per-pixel depth and a low-dimensional parameterisation of object motion.

FlowSAM. In Chapter 4, we presented *Moving Object Segmentation: All You Need Is SAM (and Flow)*, which was selected for *Oral Presentation* at ACCV 2024. The work demonstrated that the large-scale segmentation priors encoded in SAM can be effectively adapted for moving object segmentation through optical-flow-based inputs and prompts. The released code[2] has received over 330 stars on GitHub, the highest among the projects in this thesis, and the pretrained models have been widely adopted as off-the-shelf tools for motion segmentation. For example, MonoMobility [Zhou et al., 2025c] applies FlowI-SAM directly to obtain motion-part masks for monocular 3D mobility analysis. Notably, the SAM 2 technical report [Ravi et al., 2024] cites FlowSAM as a representative example of motion segmentation when motivating the broad applicability of the SAM family. The work has also influenced subsequent approaches, including EdgeTAM [Zhou et al., 2025a] and MoVieS [Lin et al., 2026], which build upon the paradigm of combining foundation-model segmentation priors with motion cues.

GMOS. In Chapter 5, we presented *Grounding Moving Object Segmentation in 3D Space and Time*, which is currently under review at NeurIPS 2026. The work represents the culmination of the moving object segmentation trajectory developed throughout Part I, moving beyond optical-flow-based representations to ground segmentation in 3D π^3 geometric representations. It introduces the *GMOS-2K* benchmark and the *MOS-I* (Instantaneous Moving Object Segmentation) evaluation protocol, the first benchmark to evaluate per-frame motion-state prediction rather than the sequence-level binary protocols that have dominated prior work. This finer temporal granularity is particularly important for applications such as robotics, real-time perception, and 4D scene understanding, where the precise timing of object motion is critical.

[2] https://github.com/Jyxarthur/flowsam

9.1.2 Movie understanding and Audio Description

AutoAD-Zero. In Chapter 6, we presented *AutoAD-Zero: A Training-Free Framework for Zero-Shot Audio Description*, one of the earliest training-free frameworks for Audio Description (AD) generation. The work introduced a character-aware visual prompting scheme together with a two-stage VLM–LLM pipeline for generating ADs without task-specific fine-tuning. It has since become a recurring training-free baseline in subsequent AD research. DistinctAD [Fang et al., 2024] builds upon its character-centric formulation by incorporating contextual modelling to generate more distinctive descriptions across neighbouring clips, while NarrAD [Park et al., 2025] enriches the training-free setting with narrative context derived from movie scripts. CoherentAD [Khandelwal et al., 2025] extends the framework towards coherent multi-sentence AD generation, and Uni-AD [Wang et al., 2025a] adopts AutoAD-Zero as a comparison baseline while improving character identification through dedicated character refinement and multimodal context modelling. The paper also introduced the *TV-AD* dataset, extending AD generation from films to the previously underexplored setting of television series. Beyond the AD task itself, the proposed character-aware visual prompting strategy was highlighted in the survey *Visual Prompting in Multimodal Large Language Models* [Wu et al., 2024b] as a representative example of prompting MLLMs with visual cues.

Shot-By-Shot. In Chapter 7, we presented *Shot-By-Shot: Film-Grammar-Aware Training-Free Audio Description Generation*, which extends AutoAD-Zero by incorporating film grammar into the AD generation process. The work treats shots as the fundamental units of temporal context and leverages thread structure and shot scale to guide description generation. It further introduces the character-independent *action score* metric for evaluating action recall in generated descriptions. Despite its recent publication, the framework has already been adopted as a primary baseline in ADQA [Kala et al., 2025], which recasts AD evaluation as a question-answering benchmark. Beyond AD generation, its shot-scale estimation component has been repurposed by VStoryGen [Li et al., 2026] to label training data and define a shot-type-alignment metric, and it is cited by recent work on camera-motion understanding [Feng et al., 2026] as evidence that shot-level cues

can steer description generation.

As the Story Unfolds. In Chapter 8, we presented *As the Story Unfolds: Watching a Film and Identifying Characters as a Human Does*, which is currently under review at NeurIPS 2026. The work introduces the *MCN* (Movie Character Naming) dataset together with a streaming framework that identifies and names characters as a film unfolds using only audiovisual cues and dialogue. Character naming provides a foundation for character-centric video understanding, as a character's visual identity must first be associated with a name before higher-level reasoning about actions, interactions, and narrative roles can be performed. The proposed benchmark establishes a new setting for studying character naming under the realistic constraints of sequential film viewing.

9.2 Future work

We outline several research directions for future exploration, organised around the two parts of the thesis.

9.2.1 Moving object segmentation

Scaling. The moving object segmentation systems developed in Part I remain limited in scale. On the model side, the current state of the art presented in Chapter 5 relies on a multi-stage pipeline that separates mask proposal from temporal propagation. A natural next step is to scale model capacity and unify these components within a single end-to-end architecture that jointly discovers, segments, and tracks moving objects directly from video. On the data side, existing benchmarks are concentrated in a relatively small set of object categories, particularly people, animals, and vehicles. Expanding both the scale and diversity of training data, especially to domains such as biomedical imaging, microscopy, robotics, industrial inspection, and remote sensing, would broaden the applicability of motion-based object perception.

Functionality. Beyond scaling, several functional extensions remain largely unexplored. First, the methods in Part I treat every independently moving region as an object, even when the motion is generated by non-object phenomena such as

shadows, water waves, or lens flares. Conversely, collective non-rigid motion, such as a flock of birds, a shoal of fish, or leaves moving in the wind, may be better represented as *group motion*, occupying an intermediate level between individual objects and the background. Second, current methods provide only a coarse description of motion, typically reducing it to a binary moving-or-static label. Richer motion representations could additionally capture magnitude, direction, and motion type, such as translation, rotation, or articulation. Such representations would naturally extend to motion forecasting, enabling the prediction of future object trajectories and behaviours over both short and long temporal horizons.

9.2.2 Movie understanding and Audio Description

From character naming to long-term character understanding. The character naming framework developed in Chapter 8 provides a foundation for streaming, character-centric understanding of films, but stops at identifying and naming the principal characters. A natural next step is to maintain richer character representations as a story unfolds, capturing relationships between characters, biographical information inferred from dialogue and visual cues, and the evolving roles they play in the narrative. Developing such representations requires reasoning over substantially longer temporal horizons than current Visual-Language Models can comfortably handle, making long-term character understanding a natural extension of the character-bank construction proposed in this thesis.

Character-aware downstream applications. A second direction is to leverage the character representations developed in Part II across a broader range of downstream tasks. Audio Description, studied in Chapters 6 and 7, is one such application, but the same character-centric foundations could also support movie captioning, plot-level retrieval, story understanding, character-centric question answering, and audience-adapted descriptions tailored to children, language learners, or viewers unfamiliar with a particular genre. The training-free design of the AutoAD-Zero family makes such adaptations particularly attractive as foundation models continue to improve.

References

Charu C Aggarwal, S Yu Philip, Jiawei Han, and Jianyong Wang (2003). "A framework for clustering evolving data streams". In: *Proceedings 2003 VLDB conference*. Elsevier.

AI@Meta (2024). "Llama 3 Model Card". In: URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Jean-Baptiste Alayrac, Joao Carreira, Relja Arandjelovic, and Andrew Zisserman (2019a). "Controllable Attention for Structured Layered Video Decomposition". In: *ICCV*.

Jean-Baptiste Alayrac, João Carreira, and Andrew Zisserman (2019b). "The Visual Centrifuge: Model-Free Layered Video Representations". In: *CVPR*.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould (2016). "SPICE: Semantic Propositional Image Caption Evaluation". In: *ECCV*.

Dawit Mureja Argaw, Fabian Caba Heilbron, Joon-Young Lee, Markus Woodson, and In So Kweon (2022). "The Anatomy of Video Editing: A Dataset and Benchmark Suite for AI-Assisted Video Editing". In: *ECCV*.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid (2021). "ViViT: A Video Vision Transformer". In: *ICCV*.

Yuki M Asano, Christian Rupprecht, Andrew Zisserman, and Andrea Vedaldi (2021). "PASS: An ImageNet replacement for self-supervised pretraining without humans". In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

AudioVault (2025). *AudioVault*. https://audiovault.net/.

Görkay Aydemir, Weidi Xie, and Fatma Güney (2023). "Self-supervised Object-Centric Learning for Videos". In: *NeurIPS*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. (2025). "Qwen3-vl technical report". In: *arXiv preprint arXiv:2511.21631*.

Renée Baillargeon and Julie DeVos (1991). "Object permanence in young infants: Further evidence". In: *Child Development*.

Renée Baillargeon, Elizabeth S. Spelke, and Stanley Wasserman (1985). "Object permanence in five-month-old infants". In: *Cognition*.

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

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

Simon Baker, Stefan Roth, Daniel Scharstein, Michael J. Black, J.P. Lewis, and Richard Szeliski (2007). "A Database and Evaluation Methodology for Optical Flow". In: *ICCV*.

Satanjeev Banerjee and Alon Lavie (2005). "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Zhipeng Bao, Pavel Tokmakov, Allan Jabri, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert (2022). "Discovering Objects That Can Move". In: *CVPR*.

Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros (2022). "Visual prompting via image inpainting". In: *NeurIPS*.

Martin Bauml, Makarand Tapaswi, and Rainer Stiefelhagen (2013). "Semi-supervised learning with constraints for person identification in multimedia data". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Daniel Bear, Chaofei Fan, Damian Mrowca, Yunzhu Li, Seth Alter, Aran Nayebi, Jeremy Schwartz, Li F Fei-Fei, Jiajun Wu, Josh Tenenbaum, et al. (2020). "Learning physical graph representations from visual scenes". In: *NeurIPS*.

Maksym Bekuzarov, Ariana Bermudez, Joon-Young Lee, and Hao Li (2023). "XMem++: Production-level Video Segmentation From Few Annotated Frames". In: *ICCV*.

Gedas Bertasius and Lorenzo Torresani (2020). "Classifying, Segmenting, and Tracking Object Instances in Video with Mask Propagation". In: *Proc. CVPR*.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani (2021). "Is Space-Time Attention All You Need for Video Understanding?" In: *Proc. ICML*.

Pia Bideau and Erik Learned-Miller (2016a). "A detailed rubric for motion segmentation". In: *arXiv preprint arXiv:1610.10033*.

Pia Bideau and Erik Learned-Miller (2016b). "It's moving! A probabilistic model for causal motion segmentation in moving camera videos". In: *ECCV*.

Pia Bideau, Aruni RoyChowdhury, Rakesh R. Menon, and Erik Learned-Miller (June 2018). “The Best of Both Worlds: Combining CNNs and Geometric Constraints for Hierarchical Motion Segmentation”. In: *CVPR*.

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

Thomas Brox and Jitendra Malik (2010). “Object segmentation by long term analysis of point trajectories”. In: *ECCV*.

Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles (2022). “Revisiting the ”Video” in Video-Language Understanding”. In: *CVPR*.

Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner (2019). “MONet: Unsupervised Scene Decomposition and Representation”. In: *arXiv preprint arXiv:1901.11390*.

Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool (2017). “One-shot video object segmentation”. In: *CVPR*.

Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool (2019). “The 2019 DAVIS Challenge on VOS: Unsupervised Multi-Object Segmentation”. In: *arXiv preprint arXiv:1905.00737*.

Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee (2024a). “Making large multimodal models understand arbitrary visual prompts”. In: *CVPR*.

Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee (2024b). “Vip-llava: Making large multimodal models understand arbitrary visual prompts”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Feng Cao, Martin Estert, Weining Qian, and Aoying Zhou (2006). “Density-based clustering over an evolving data stream with noise”. In: *Proceedings of the 2006 SIAM international conference on data mining*. SIAM.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou,

Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer (2026). "SAM 3: Segment Anything with Concepts". In: *ICLR*.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko (2020). "End-to-end object detection with transformers". In: *ECCV*.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin (2021). "Emerging Properties in Self-Supervised Vision Transformers". In: *ICCV*.

Joao Carreira and Andrew Zisserman (2017). "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset". In: *CVPR*.

Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian (2023). "Segment Any 3D Gaussians". In: *arXiv preprint arXiv:2312.00860*.

Boris Chen, Amir Ziai, Rebecca S. Tucker, and Yuchen Xie (2023a). "Match Cutting: Finding Cuts With Smooth Visual Transitions". In: *WACV*.

David Chen and William B Dolan (2011). "Collecting highly parallel data for paraphrase evaluation". In: *ACL*.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao (2023b). "Shikra: Unleashing multimodal llm's referential dialogue magic". In: *arXiv preprint arXiv:2306.15195*.

Shaoxiang Chen and Yu-Gang Jiang (2021). "Towards bridging event captioner and sentence localizer for weakly supervised dense event captioning". In: *CVPR*.

Shimin Chen, Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma (2024a). "TimeMarker: A Versatile Video-LLM for Long and Short Video Understanding with Superior Temporal Localization Ability". In: *arXiv preprint arXiv:2411.18211*.

Tianrun Chen, Lanyun Zhu, Chaotao Ding, Runlong Cao, Shangzhan Zhang, Yan Wang, Zejian Li, Lingyun Sun, Papa Mao, and Ying Zang (2023c). "SAM-Adapter: Adapting Segment Anything in Underperformed Scenes". In: *ICCV Workshop*.

Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and Sergey Tulyakov (2024b). "Panda-70M: Captioning 70M Videos with Multiple Cross-Modality Teachers". In: *CVPR*.

Xieyuanli Chen, Shijie Li, Benedikt Mersch, Louis Wiesmann, Jürgen Gall, Jens Behley, and Cyrill Stachniss (2021). "Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data". In: *IROS*.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen (2025). "Easi3R: Estimating Disentangled Motion from DUSt3R Without Training". In: *ICCV*.

Zeyu Chen, Yana Zhang, Suya Zhang, and Cheng Yang (2022). "Study on location bias of CNN for shot scale classification". In: *Multimedia Tools and Applications*.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar (2022a). "Masked-Attention Mask Transformer for Universal Image Segmentation". In: *Proc. CVPR*.

Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov (2021). "Per-Pixel Classification is Not All You Need for Semantic Segmentation". In: *NeurIPS*.

Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing (2024a). "Putting the Object Back into Video Object Segmentation". In: *CVPR*.

Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee (2023a). "Tracking Anything with Decoupled Video Segmentation". In: *ICCV*.

Ho Kei Cheng and Alexander G. Schwing (2022). "XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model". In: *ECCV*.

Xuelian Cheng, Huan Xiong, Deng-Ping Fan, Yiran Zhong, Mehrtash Harandi, Tom Drummond, and Zongyuan Ge (2022b). "Implicit Motion Handling for Video Camouflaged Object Detection". In: *CVPR*.

Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang (2023b). "Segment and track anything". In: *arXiv preprint arXiv:2305.06558*.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing (2024b). "VideoLLaMA 2: Advancing Spatial-Temporal Modeling and Audio Understanding in Video-LLMs". In: *arXiv preprint arXiv:2406.07476*.

Donghyeon Cho, Sungeun Hong, Sungil Kang, and Jiwon Kim (2019). "Key Instance Selection for Unsupervised Video Object Segmentation". In: *arXiv preprint arXiv:1906.07851*.

Suhwan Cho, Minhyeok Lee, Seunghoon Lee, Dogyoon Lee, Heeseung Choi, Ig-Jae Kim, and Sangyoun Lee (2024). "Dual Prototype Attention for Unsupervised Video Object Segmentation". In: *CVPR*.

Suhwan Cho, Minhyeok Lee, Seunghoon Lee, Chaewon Park, Donghyeong Kim, and Sangyoun Lee (2023). "Treating Motion as Option to Reduce Motion Dependency in Unsupervised Video Object Segmentation". In: *WACV*.

Subhabrata Choudhury, Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht (2022). "Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion". In: *BMVC*.

Peng Chu, Jiang Wang, and Andre Abrantes (2024). "LLM-AD: Large Language Model based Audio Description System". In: *arXiv preprint arXiv:2405.00983*.

Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid (2011). "Unsupervised metric learning for face identification in TV video". In: *2011 International Conference on Computer Vision*. IEEE.

Timothee Cour, Benjamin Sapp, Akash Nagle, and Ben Taskar (2010). "Talking pictures: Temporal grouping and dialog-supervised person recognition". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE.

Achal Dave, Pavel Tokmakov, and Deva Ramanan (2019). "Towards segmenting anything that moves". In: *ICCV workshops*.

Thanos Delatolas, Vicky Kalogeiton, and Dim P. Papadopoulos (2024). "Learning the What and How of Annotation in Video Object Segmentation". In: *WACV*.

Andong Deng, Zhongpai Gao, Anwesa Choudhuri, Benjamin Planche, Meng Zheng, Bin Wang, Terrence Chen, Chen Chen, and Ziyan Wu (2024). "Seq2Time: Sequential Knowledge Transfer for Video LLM Temporal Grounding". In: *arXiv preprint arXiv:2411.16932*.

Chaorui Deng, Shizhe Chen, Da Chen, Yuan He, and Qi Wu (2021). "Sketch, ground, and refine: Top-down dense video captioning". In: *CVPR*.

Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou (2020). "RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild". In: *CVPR*.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou (2019). "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". In: *CVPR*.

Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang (2022). "TAP-Vid: A Benchmark for Tracking Any Point in a Video". In: *NeurIPS*.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman (2023). "TAPIR: Tracking Any Point with Per-Frame Initialization and Temporal Refinement". In: *ICCV*.

Carl Doersch and Andrew Zisserman (2019). “Sim2real transfer learning for 3D human pose estimation: motion to the rescue”. In: *NeurIPS*.

Gamaleldin F. Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C. Mozer, and Thomas Kipf (2022). “SAVi++: Towards end-to-end object-centric learning from real-world videos”. In: *arXiv preprint arXiv:2206.07764*.

Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner (2020). “GENESIS: Generative scene inference and sampling with object-centric latent representations”. In: *ICLR*.

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, koray kavukcuoglu koray, and Geoffrey E Hinton (2016). “Attend, Infer, Repeat: Fast Scene Understanding with Generative Models”. In: *NeurIPS*.

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

Alon Faktor and Michal Irani (2014). “Video Segmentation by Non-Local Consensus voting.” In: *BMVC*.

Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen (2019). “Shifting more attention to video salient object detection”. In: *CVPR*.

Ke Fan, Zechen Bai, Tianjun Xiao, Dominik Zietlow, Max Horn, Zixu Zhao, Carl-Johann Simon-Gabriel, Mike Zheng Shou, Francesco Locatello, Bernt Schiele, Thomas Brox, Zheng Zhang, Yanwei Fu, and Tong He (2023). “Unsupervised Open-Vocabulary Object Localization in Videos”. In: *ICCV*.

Bo Fang, Wenhao Wu, Qiangqiang Wu, Yuxin Song, and Antoni B. Chan (2024). “DistinctAD: Distinctive Audio Description Generation in Contexts”. In: *arXiv preprint arXiv:2411.18180*.

Dennis Fedorishin, Lie Lu, Srirangaraj Setlur, and Venu Govindaraju (2024). “Audio Match Cutting: Finding and Creating Matching Audio Transitions in Movies and Videos”. In: *ICASSP*.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He (2019). “SlowFast Networks for Video Recognition”. In: *ICCV*.

Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman (2016). “Convolutional Two-Stream Network Fusion for Video Action Recognition”. In: *CVPR*.

Haoan Feng, Sri Harsha Musunuri, and Guan-Ming Su (2026). “Geometry-Guided Camera Motion Understanding in VideoLLMs”. In: *arXiv preprint arXiv:2603.13119*.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue (2025a). "Video-r1: Reinforcing video reasoning in mllms". In: *arXiv preprint arXiv:2503.21776*.

Yicheng Feng, Yijiang Li, Wanpeng Zhang, Sipeng Zheng, Hao Luo, Zihao Yue, and Zongqing Lu (2025b). "VideoOrion: Tokenizing Object Dynamics in Videos". In: *Proc. ICCV*.

Andrew Fitzgibbon and Andrew Zisserman (2002). "On affine invariant clustering and automatic cast listing in movies". In: *European Conference on Computer Vision*. Springer.

Peter R. Florence, Lucas Manuelli, and Russ Tedrake (2018). "Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation". In: *Proceedings of The 2nd Conference on Robot Learning*.

Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi (2012). "Video segmentation by tracing discontinuities in a trajectory embedding". In: *CVPR*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. (2025). "Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Yossi Gandelsman, Assaf Shocher, and Michal Irani (2019). ""Double-DIP": Unsupervised Image Decomposition via Coupled Deep-Image-Priors". In: *CVPR*.

Ridouane Ghermi, Xi Wang, Vicky Kalogeiton, and Ivan Laptev (2024). "Short Film Dataset (SFD): A Benchmark for Story-Level Video Understanding". In: *arXiv preprint arXiv:2406.10221*.

Lily Goli, Sara Sabour, Mark Matthews, Brubaker Marcus, Dmitry Lagun, Alec Jacobson, David J. Fleet, Saurabh Saxena, and Andrea Tagliasacchi (2025). "RoMo: Robust Motion Segmentation Improves Structure from Motion". In: *ICCV*.

Melvyn A. Goodale and A. David Milner (1992). "Separate visual pathways for perception and action". In: *Trends in Neurosciences*.

Google DeepMind (Dec. 2025). *Gemini 3 Pro Model Card*. Tech. rep. Google. URL: https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai,

Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi (2022). "Kubric: A Scalable Dataset Generator". In: *CVPR*.

Klaus Greff, Raphael Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner (2019). "Multi-object representation learning with iterative variational inference". In: *ICML*.

Klaus Greff, Antti Rasmus, Mathias Berglund, Tele Hao, Harri Valpola, and Jürgen Schmidhuber (2016). "Tagger: Deep Unsupervised Perceptual Grouping". In: *NeurIPS*.

Brent A. Griffin and Jason J. Corso (2019). "BubbleNets: Learning to Select the Guidance Frame in Video Object Segmentation by Deep Sorting Frames". In: *CVPR*.

Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. (2018). "Ava: A video dataset of spatio-temporally localized atomic visual actions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Zhongrui Gui, Junyu Xie, Tengda Han, Weidi Xie, and Andrew Zisserman (2025). "Character-Centric Understanding of Animated Movies". In: *Proceedings of the 33rd ACM International Conference on Multimedia*.

Guidelines for Audio Describers (n.d.). https://adp.acb.org/guidelines.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. (2025). "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning". In: *arXiv preprint arXiv:2501.12948*.

Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Xi Chen, and Bo Zhao (2024a). "VTG-LLM: Integrating Timestamp Knowledge into Video LLMs for Enhanced Video Temporal Grounding". In: *arXiv preprint arXiv:2405.13382*.

Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Qingbin Liu, and Xi Chen (2024b). "TRACE: Temporal Grounding Video LLM via Causal Event Modeling". In: *arXiv preprint arXiv:2410.05643*.

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*.

Adam W. Harley, Zhaoyuan Fang, and Katerina Fragkiadaki (2022). "Particle Video Revisited: Tracking Through Occlusions Using Point Trajectories". In: *ECCV*.

Monica-Laura Haurilet, Makarand Tapaswi, Ziad Al-Halah, and Rainer Stiefelhagen (2016). "Naming TV characters by watching and analyzing dialogs". In: *2016 IEEE winter conference on applications of computer vision (WACV)*. IEEE.

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick (2017). "Mask R-CNN". In: *ICCV*.

Xiankang He, Peile Lin, Ying Cui, Dongyan Guo, Chunhua Shen, and Xiaoqin Zhang (2026). "GeoMotion: Rethinking Motion Segmentation via Latent 4D Geometry". In: *CVPR*.

Yichen He, Yuan Lin, Jianchao Wu, Hanchong Zhang, Yuchen Zhang, and Ruicheng Le (2025). "StoryTeller: Improving Long Video Description through Global Audio-Visual Character Identification". In: *arXiv preprint arXiv:2411.07076*.

Zhen He, Jian Li, Daxue Liu, Hangen He, and David Barber (2019). "Tracking by Animation: Unsupervised Learning of Multi-Object Attentive Trackers". In: *CVPR*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell (2018). "Localizing Moments in Video with Temporal Language." In: *EMNLP*.

Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson (2022). "Object-Region Video Transformers". In: *Proc. CVPR*.

Minh Hoai and Andrew Zisserman (2014). "Thread-Safe: Towards Recognizing Human Actions Across Shot Boundaries". In: *ACCV*.

Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. (2024). "CogVLM2: Visual Language Models for Image and Video Understanding". In: *arXiv preprint arXiv:2408.16500*.

Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G. Schwing (2019). "SAIL-VOS: Semantic Amodal Instance Level Video Object Segmentation – A Synthetic Dataset and Baselines". In: *CVPR*.

De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz (2024a). "LITA: Language Instructed Temporal-Localization Assistant". In: *ECCV*.

Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu (2024b). "VTimeLLM: Empower LLM to Grasp Video Moments". In: *CVPR*.

Nan Huang, Wenzhao Zheng, Chenfeng Xu, Kurt Keutzer, Shanghang Zhang, Angjoo Kanazawa, and Qianqian Wang (2025a). "Segment Any Motion in Videos". In: *CVPR*.

Qingqiu Huang, Wentao Liu, and Dahua Lin (2018). "Person Search in Videos with One Portrait Through Visual and Temporal Links". In: *ECCV*.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin (2020). "MovieNet: A Holistic Dataset for Movie Understanding". In: *ECCV*.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Xu Tang, Yao Hu, and Shaohui Lin (2025b). "Vision-r1: Incentivizing reasoning capability in multimodal large language models". In: *arXiv preprint arXiv:2503.06749*.

Vladimir Iashin and Esa Rahtu (2020a). "A better use of audio-visual cues: Dense video captioning with bi-modal transformer". In: *BMVC*.

Vladimir Iashin and Esa Rahtu (2020b). "Multi-modal dense video captioning". In: *CVPR Workshops on Multimodal Learning*.

Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius (2024). "Video ReCap: Recursive Captioning of Hour-Long Videos". In: *CVPR*.

Allan Jabri, Andrew Owens, and Alexei A Efros (2020). "Space-Time Correspondence as a Contrastive Random Walk". In: *NeurIPS*.

Suyog Dutt Jain, Bo Xiong, and Kristen Grauman (2017). "Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos". In: *CVPR*.

Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles (2020). "Action Genome: Actions As Compositions of Spatio-Temporal Scene Graphs". In: *Proc. CVPR*.

Ge-Peng Ji, Keren Fu, Zhe Wu, Deng-Ping Fan, Jianbing Shen, and Ling Shao (2021). "Full-Duplex Strategy for Video Object Segmentation". In: *ICCV*.

Yatai Ji, Shilong Zhang, Jie Wu, Peize Sun, Weifeng Chen, Xuefeng Xiao, Sidi Yang, Yujiu Yang, and Ping Luo (2025). "IDA-VLM: Towards Movie Understanding via ID-Aware Large Vision-Language Model". In: *Proc. ICLR*.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim (2022). "Visual prompt tuning". In: *ECCV*.

Jindong Jiang, Fei Deng, Gautam Singh, Minseung Lee, and Sungjin Ahn (2024). "Slot State Space Models". In: *NeurIPS*.

Jindong Jiang, Sepehr Janghorbani, Gerard de Melo, and Sungjin Ahn (2020). "SCALOR: Generative World Models with Scalable Object Representations". In: *ICLR*.

Xuekun Jiang, Libiao Jin, Anyi Rao, Linning Xu, and Dahua Lin (2022). "Jointly Learning the Attributes and Composition of Shots for Boundary Detection in Videos". In: *Trans. Multi.*

Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie (2022). "Prompting visual-language models for efficient video understanding". In: *ECCV*.

Yeong Jun Koh and Chang-Su Kim (2017). "Primary object segmentation in videos based on region augmentation and reduction". In: *CVPR*.

Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matthew Botvinick, Alexander Lerchner, and Christopher P. Burgess (2021). "SIMONe: View-Invariant, Temporally-Abstracted Object Representations via Unsupervised Video Decomposition". In: *arXiv preprint arxiv.2106.03849*.

Divy Kala, Eshika Khandelwal, and Makarand Tapaswi (2025). "What You See is What You Ask: Evaluating Audio Descriptions". In.

Vicky Kalogeiton and Andrew Zisserman (2020). "Constrained video face clustering using 1nn relations". In: *https://ora.ox.ac.uk/objects/uuid:23f369b3-794b-42a6-a6af-ab4b0d726ba4/files/spn89d688d*.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht (2025). "CoTracker3: Simpler and Better Point Tracking by Pseudo-Labeling Real Videos". In: *ICCV*.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht (2023). "DynamicStereo: Consistent Dynamic Depth from Stereo Videos". In: *CVPR*.

Laurynas Karazija, Iro Laina, Christian Rupprecht, and Andrea Vedaldi (2024). "Learning segmentation from point trajectories". In: *NeurIPS*.

Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel (2021). "Layered neural atlases for consistent video editing". In: *ACM Trans. Graph.*

Steve D. Katz (2019). *Film Directing Shot by Shot: Visualizing from Concept to Screen*. Michael Wiese Productions, ISBN: 1615932976.

Margret Keuper, Bjoern Andres, and Thomas Brox (2015). “Motion trajectory segmentation via minimum cost multicuts”. In: *ICCV*.

Eshika Khandelwal, Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Andrew Zisserman, Gül Varol, and Makarand Tapaswi (2025). “More than a Moment: Towards Coherent Sequences of Audio Descriptions”. In: *arXiv preprint arXiv:2510.25440*.

Minkuk Kim, Hyeon Bae Kim, Jinyoung Moon, Jinwoo Choi, and Seong Tae Kim (2024). “Do You Remember? Dense Video Captioning with Cross-Modal Memory Retrieval”. In: *CVPR*.

Diederik P. Kingma and Jimmy Ba (2015). “Adam: A Method for Stochastic Optimization”. In: *ICLR*.

Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff (2022). “Conditional Object-Centric Learning from Video”. In: *ICLR*.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick (2023). “Segment Anything”. In: *ICCV*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa (2022). “Large language models are zero-shot reasoners”. In: *Advances in neural information processing systems*.

Bruno Korbar, Jaesung Huh, and Andrew Zisserman (2024). “Look, Listen and Recognise: Character-Aware Audio-Visual Subtitling”. In: *arXiv preprint arXiv:2401.12039*.

Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner (2018). “Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects”. In: *NeurIPS*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles (2017). “Dense-captioning events in videos”. In: *ICCV*.

Alina Kuznetsova, Aakrati Talati, Yiwen Luo, Keith Simmons, and Vittorio Ferrari (2021). “Efficient Video Annotation With Visual Interpolation and Frame Selection Guidance”. In: *WACV*.

Zihang Lai, Erika Lu, and Weidi Xie (2020). “MAST: A Memory-Augmented Self-Supervised Tracker”. In: *CVPR*.

Zihang Lai and Weidi Xie (2019). “Self-supervised Learning for Video Correspondence Flow”. In: *BMVC*.

Hala Lamdouar, Weidi Xie, and Andrew Zisserman (2021). “Segmenting Invisible Moving Objects”. In: *BMVC*.

Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman (2020). "Betrayed by Motion: Camouflaged Object Discovery via Motion Segmentation". In: *ACCV*.

Dong Lao, Zhengyang Hu, Francesco Locatello, Yanchao Yang, and Stefano Soatto (2023). "Divided Attention: Unsupervised Multi-Object Discovery with Contextually Separated Slots". In: *arXiv preprint arxiv:2304.01430*.

Minhyeok Lee, Suhwan Cho, Dogyoon Lee, Chaewon Park, Jungho Lee, and Sangyoun Lee (2024). "Guided Slot Attention for Unsupervised Video Object Segmentation". In: *CVPR*.

Minhyeok Lee, Suhwan Cho, Seunghoon Lee, Chaewon Park, and Sangyoun Lee (2023). "Unsupervised Video Object Segmentation via Prototype Memory Network". In: *WACV*.

Yao-Chih Lee, Erika Lu, Sarah Rumbley, Michal Geyer, Jia-Bin Huang, Tali Dekel, and Forrester Cole (2025). "Generative Omnimatte: Learning to Decompose Video into Layers". In: *Proc. CVPR*.

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

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal (2020). "TVR: A Large-Scale Dataset for Video-Subtitle Moment Retrieval". In: *ECCV*.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud (2024). "Grounding image matching in 3d with mast3r". In: *ECCV*.

Brian Lester, Rami Al-Rfou, and Noah Constant (2021). "The power of scale for parameter-efficient prompt tuning". In: *arXiv preprint arXiv:2104.08691*.

Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum (2023a). "Mask DINO: Towards a Unified Transformer-Based Framework for Object Detection and Segmentation". In: *CVPR*.

Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg (2013). "Video Segmentation by Tracking Many Figure-Ground Segments". In: *ICCV*.

Ke Li and Jitendra Malik (2016). "Amodal Instance Segmentation". In: *ECCV*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling.

Qipeng Li, Yuan Zhuang, You Chen, Jianzhu Huai, Miaomiao Li, Tianxiang Ma, Yufei Tang, and Xinlian Liang (2023b). "Multi-sensor Fusion for Robust Localization with Moving Object Segmentation in Complex Dynamic 3D Scenes". In: *International Journal of Applied Earth Observation and Geoinformation*.

Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qin Huang, and C.-C. Jay Kuo (2018). "Instance Embedding Transfer to Unsupervised Video Object Segmentation". In: *CVPR*.

Wei-Hua Li, Cheng Sun, and Chu-Song Chen (2026). “Customized Visual Storytelling with Unified Multimodal LLMs”. In: *arXiv preprint arXiv:2603.27690*.

Xiang Lisa Li and Percy Liang (2021). “Prefix-tuning: Optimizing continuous prompts for generation”. In: *arXiv preprint arXiv:2101.00190*.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang (2025a). “Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning”. In: *arXiv preprint arXiv:2504.06958*.

Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang (2019a). “Joint-task self-supervised learning for temporal correspondence”. In: *NeurIPS*.

Yuzhi Li, Feng Tian, Haojun Xu, and Tianfeng Lu (2023c). “Toward Unified and Quantitative Cinematic Shot Attribute Analysis”. In: *Electronics*.

Zhaowei Li, Qi Xu, Dong Zhang, Hang Song, Yiqing Cai, Qi Qi, Ran Zhou, Junting Pan, Zefeng Li, Vu Tu, et al. (2024). “Groundinggpt: Language enhanced multi-modal grounding model”. In: *ACL*.

Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T. Freeman (2019b). “Learning the Depths of Moving People by Watching Frozen People”. In: *CVPR*.

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely (2025b). “MegaSaM: Accurate, Fast and Robust Structure and Motion from Casual Dynamic Videos”. In: *CVPR*.

Long Lian, Zhirong Wu, and Stella X. Yu (2023). “Bootstrapping Objectness From Videos by Relaxed Common Fate and Visual Grouping”. In: *CVPR*.

Chenguo Lin, Yuchen Lin, Panwang Pan, Yifan Yu, Tao Hu, Honglei Yan, Katerina Fragkiadaki, and Yadong Mu (2026). “MoVieS: Motion-Aware 4D Dynamic View Synthesis in One Second”. In: *Proc. CVPR*.

Chin-Yew Lin (2004). “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*.

Huaijia Lin, Ruizheng Wu, Shu Liu, Jiangbo Lu, and Jiaya Jia (2021). “Video Instance Segmentation with a Propose-Reduce Paradigm”. In: *ICCV*.

Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, Ce Liu, and Lijuan Wang (2023). “MM-Vid: Advancing Video Understanding with GPT-4V(ision)”. In: *arXiv preprint arXiv:2310.19773*.

Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang (2022). "SwinBERT: End-to-end transformers with sparse attention for video captioning". In: *CVPR*.

Kevin Qinghong Lin, Pengchuan Zhang, Difei Gao, Xide Xia, Joya Chen, Ziteng Gao, Jinheng Xie, Xuhong Xiao, and Mike Zheng Shou (2024). "Learning Video Context as Interleaved Multimodal Sequences". In: *ECCV*.

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

Huan Ling, David Acuna, Karsten Kreis, Seung Wook Kim, and Sanja Fidler (2020). "Variational Amodal Object Completion". In: *NeurIPS*.

Daizong Liu, Dongdong Yu, Changhu Wang, and Pan Zhou (2021). "F2Net: Learning to Focus on the Foreground for Unsupervised Video Object Segmentation". In: *AAAI*.

Qihao Liu, Junfeng Wu, Yi Jiang, Xiang Bai, Alan Yuille, and Song Bai (2023). "InstMove: Instance Motion for Object-centric Video Segmentation". In: *arXiv preprint arXiv:2303.08132*.

Weihuang Liu, Xi Shen, Haolun Li, Xiuli Bi, Bo Liu, Chi-Man Pun, and Xiaodong Cun (2024). "Depth-aware Test-Time Training for Zero-shot Video Object Segmentation". In: *CVPR*.

Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi (2022). "HOI4D: A 4D Egocentric Dataset for Category-Level Human-Object Interaction". In: *CVPR*.

Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia (2025a). "Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement". In: *arXiv preprint arXiv:2503.06520*.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang (2025b). "Visual-rft: Visual reinforcement fine-tuning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf (2020). "Object-Centric Learning with Slot Attention". In: *NeurIPS*.

Ilya Loshchilov and Frank Hutter (2019). "Decoupled Weight Decay Regularization". In: *ICLR*.

Erika Lu, Forrester Cole, Tali Dekel, Weidi Xie, Andrew Zisserman, David Salesin, William T. Freeman, and Michael Rubinstein (2020). "Layered Neural Rendering for Retiming People in Video". In: *SIGGRAPH Asia*.

Erika Lu, Forrester Cole, Tali Dekel, Andrew Zisserman, William T. Freeman, and Michael Rubinstein (June 2021). "Omnimatte: Associating Objects and Their Effects in Video". In: *CVPR*.

Fengtian Lu, Yuzhi Li, and Feng Tian (2024). "Exploring challenge and explainable shot type classification using SAM-guided approaches". In: *Signal, Image and Video Processing*.

Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli (2019). "See More, Know More: Unsupervised Video Object Segmentation With Co-Attention Siamese Networks". In: *CVPR*.

Zheng Lu and Kristen Grauman (2013). "Story-driven summarization for egocentric video". In: *CVPR*.

Jonathon Luiten, Idil Esen Zulfikar, and Bastian Leibe (2020). "UnOVOST: Unsupervised Offline Video Object Segmentation and Tracking". In: *WACV*.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou (2020). "UniViLM: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation". In: *arXiv preprint arXiv:2002.06353*.

Chih-Yao Ma, Min-Hung Chen, Zsolt Kira, and Ghassan AlRegib (2018). "TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition". In: *Signal Processing: Image Communication*.

Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang (2024). "Segment anything in medical images". In: *Nature Communications*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan (2024). "Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models". In.

Aravindh Mahendran, James Thewlis, and Andrea Vedaldi (2018). "Self-Supervised Segmentation by Grouping Optical-Flow". In: *ECCV*.

Anna Manasyan, Maximilian Seitzer, Filip Radovic, Georg Martius, and Andrii Zadaianchuk (2025). "Temporally Consistent Object-Centric Learning by Contrasting Slots". In: *Proc. CVPR*.

Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool (2018). "Video Object Segmentation Without Temporal Information". In: *TPAMI*.

Manuel Marin-Jimenez, Vicky Kalogeiton, Pablo Medina-Suarez, and Andrew Zisserman (Dec. 2020). "LAEO-Net++: revisiting people Looking At Each Other in videos". In: *IEEE TPAMI*.

Manuel Jesús Marin-Jimenez, Andrew Zisserman, Marcin Eichner, and Vittorio Ferrari (2014). "Detecting people looking at each other in videos". In: *International Journal of Computer Vision*.

David R. Martin, Charless C. Fowlkes, and Jitendra Malik (2004). "Learning to detect natural image boundaries using local brightness, color, and texture cues". In: *TPAMI*.

Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell (2020). "Something-Else: Compositional Action Recognition With Spatial-Temporal Interaction Networks". In: *Proc. CVPR*.

Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi (2022). "Deep Spectral Methods: A Surprisingly Strong Baseline for Unsupervised Semantic Segmentation and Localization". In: *CVPR*. IEEE.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. (2025). "Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning". In: *arXiv preprint arXiv:2503.07365*.

Meta (2024). "The Llama 3 Herd of Models". In: *arXiv preprint arXiv: 2407.21783*.

Etienne Meunier, Anaïs Badoual, and Patrick Bouthemy (2023). "EM-driven unsupervised learning for efficient motion segmentation". In: *TPAMI*.

Etienne Meunier and Patrick Bouthemy (2023a). "Unsupervised motion segmentation in one go: Smooth long-term model over a video". In: *arXiv preprint arXiv:2310.01040*.

Etienne Meunier and Patrick Bouthemy (2023b). "Unsupervised Space-Time Network for Temporally-Consistent Segmentation of Multiple Motions". In: *CVPR*.

Bo Miao, Mohammed Bennamoun, Yongsheng Gao, and Ajmal Mian (2022). "Self-Supervised Video Object Segmentation by Motion-Aware Mask Propagation". In: *IEEE International Conference on Multimedia and Expo (ICME)*. IEEE.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic (2019). "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips". In: *ICCV*.

Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig (2024). "Compositional chain-of-thought prompting for large multimodal models". In:

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

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

James Monaco (2009). *How to Read a Film: Movies, Media, and Beyond*. OUP, ISBN: 0571168973.

Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han (2019). "Streamlined dense video captioning". In: *CVPR*.

Shehan Munasinghe, Hanan Gani, Wenqi Zhu, Jiale Cao, Eric Xing, Fahad Shahbaz Khan, and Salman Khan (2025). "VideoGLaMM : A Large Multimodal Model for Pixel-Level Visual Grounding in Videos". In: *Proc. CVPR*.

Walter Murch (2021). *In the Blink of an Eye: A Perspective on Film Editing*. Silman-James Press, ISBN: 1879505622.

Arsha Nagrani, Samuel Albanie, and Andrew Zisserman (2018). "Learnable pins: Cross-modal embeddings for person identity". In: *Proceedings of the European conference on computer vision (ECCV)*.

Arsha Nagrani and Andrew Zisserman (2018). "From benedict cumberbatch to sherlock holmes: Character identification in tv series without a script". In: *arXiv preprint arXiv:1801.10442*.

Nasaruddin Nasaruddin, Kahlil Muchtar, Afdhal Afdhal, and Alvin Prayuda Juniarta Dwiyantoro (2020). "Deep anomaly detection through visual attention in surveillance videos". In: *Journal of Big Data*.

Joel Norman (2002). "Two visual systems and two theories of perception: An attempt to reconcile the constructivist and ecological approaches". In: *The Behavioral and brain sciences*.

David Nukrai, Ron Mokady, and Amir Globerson (2022). "Text-Only Training for Image Captioning using Noise-Injected CLIP". In: *ENNLP*.

Peter Ochs and Thomas Brox (2011). "Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions". In: *ICCV*.

Peter Ochs, Jitendra Malik, and Thomas Brox (2014). "Segmentation of moving objects by long term video analysis". In: *TPAMI*.

Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim (2019). "Video object segmentation using space-time memory networks". In: *ICCV*.

OpenAI (2024). "GPT-4 Technical Report". In: *arXiv preprint arXiv: 2303.08774*.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa,

Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski (2024). "DINOv2: Learning Robust Visual Features without Supervision". In: *TMLR*.

Charles Otto, Dayong Wang, and Anil K Jain (2017). "Clustering millions of faces by identity". In: *IEEE transactions on pattern analysis and machine intelligence*.

Xiao Pan, Peike Li, Zongxin Yang, Huiling Zhou, Chang Zhou, Hongxia Yang, Jingren Zhou, and Yi Yang (2022). "In-N-Out Generative Learning for Dense Unsupervised Video Segmentation". In: *ACM MM*.

Anestis Papazoglou and Vittorio Ferrari (2013). "Fast object segmentation in unconstrained video". In: *ICCV*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). "BLEU: a method for automatic evaluation of machine translation". In: *ACL*.

Alejandro Pardo, Fabian Caba Heilbron, Juan León Alcázar, Ali Thabet, and Bernard Ghanem (2022). "Moviecuts: A new dataset and benchmark for cut type recognition". In: *ECCV*.

Alejandro Pardo, Fabio Pizzati, Tong Zhang, Alexander Pondaven, Philip Torr, Juan Camilo Perez, and Bernard Ghanem (2024). "MatchDiffusion: Training-free Generation of Match-Cuts". In: *arXiv preprint arXiv:2411.18677*.

Jaehyeong Park, Junchel Ye, Seungkook Lee, Hyun W. Ka, and Dongsu Han (2025). "NarrAD: Automatic Generation of Audio Descriptions for Movies with Rich Narrative Context". In: *Proc. WACV*.

Gensheng Pei, Fumin Shen, Yazhou Yao, Guo-Sen Xie, Zhenmin Tang, and Jinhui Tang (2022). "Hierarchical Feature Alignment Network for Unsupervised Video Object Segmentation". In: *ECCV*.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung (2016). "A benchmark dataset and evaluation methodology for video object segmentation". In: *CVPR*.

Toby Perrett, Tengda Han, Dima Damen, and Andrew Zisserman (2024). "It's Just Another Day: Unique Video Captioning by Discriminative Prompting". In: *ACCV*.

Jean Piaget (1954). "The Construction of Reality in the Child". In: *New York: Basic Books*.

Stefano Pini, Marcella Cornia, Federico Bolelli, Lorenzo Baraldi, and Rita Cucchiara (2019). "M-VAD names: a dataset for video captioning with naming". In: *Multimedia Tools and Applications*.

Georgy Ponimatkin, Nermin Samet, Yang Xiao, Yuming Du, Renaud Marlet, and Vincent Lepetit (2023). “A Simple and Powerful Global Optimization for Unsupervised Video Object Segmentation”. In: *WACV*.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool (2017). “The 2017 davis challenge on video object segmentation”. In: *arXiv preprint arXiv:1704.00675*.

Will Price and Dima Damen (2020). “Play Fair: Frame Attributions in Video Models”. In: *ACCV*.

Junfu Pu, Yuxin Chen, Teng Wang, and Ying Shan (2026). “OmniScript: Towards Audio-Visual Script Generation for Long-Form Cinematic Video”. In: *arXiv preprint arXiv:2604.11102*.

Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip Torr, and Song Bai (2022). “Occluded Video Instance Segmentation: A Benchmark”. In: *IJCV*.

Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia (2019). “Amodal Instance Segmentation With KINS Dataset”. In: *CVPR*.

Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang (2024). “Momentor: Advancing Video Large Language Model with Fine-Grained Temporal Reasoning”. In: *ICML*.

Mengxue Qu, Xiaodong Chen, Wu Liu, Alicia Li, and Yao Zhao (2024). “ChatVTG: Video Temporal Grounding via Chat with Video Dialogue Large Language Models”. In: *CVPRW*.

Haran Raajesh, Naveen Reddy Desanur, Zeeshan Khan, and Makarand Tapaswi (2024). “MICap: A Unified Model for Identity-aware Movie Descriptions”. In: *CVPR*.

Tanzila Rahman, Bicheng Xu, and Leonid Sigal (2019). “Watch, listen and tell: Multi-modal weakly supervised dense event captioning”. In: *ICCV*.

Vignesh Ramanathan, Armand Joulin, Percy Liang, and Li Fei-Fei (2014). “Linking people in videos with “their” names using coreference resolution”. In: *European conference on computer vision*. Springer.

Anyi Rao, Xuekun Jiang, Sichen Wang, Yuwei Guo, Zihao Liu, Bo Dai, Long Pang, Xiaoyu Wu, Dahua Lin, and Libiao Jin (2022). “Temporal and Contextual Transformer for Multi-Camera Editing of TV Shows”. In: *Proceedings of the CVEU Workshop at ICCV*.

Anyi Rao, Jiaze Wang, Linning Xu, Xuekun Jiang, Qingqiu Huang, Bolei Zhou, and Dahua Lin (2020). “A Unified Framework for Shot Type Classification Based on Subject Centric Lens”. In: *ECCV*.

Michalis Raptis and Leonid Sigal (2013). “Poselet Key-Framing: A Model for Human Activity Recognition”. In: *CVPR*.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer (2024). “SAM 2: Segment Anything in Images and Videos”. In: *arXiv preprint arXiv:2408.00714*.

Adria Recasens, Carl Vondrick, Aditya Khosla, and Antonio Torralba (2017). “Following gaze in video”. In: *Proceedings of the IEEE International Conference on Computer Vision*.

Dima Rekesh, Nithin Rao Koluguri, Samuel Kriman, Somshubra Majumdar, Vahid Noroozi, He Huang, Oleksii Hrinchuk, Krishna Puvvada, Ankur Kumar, Jagadeesh Balam, et al. (2023). “Fast conformer with linearly scalable attention for efficient speech recognition”. In: *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun (2015). “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Simiao Ren, Francesco Luzi, Saad Lahrichi, Kaleb Kassaw, Leslie M Collins, Kyle Bradbury, and Jordan M Malof (2024). “Segment anything, from space?” In: *WACV*.

Sucheng Ren, Wenxi Liu, Yongtuo Liu, Haoxin Chen, Guoqiang Han, and Shengfeng He (2021). “Reciprocal Transformations for Unsupervised Video Object Segmentation”. In: *CVPR*.

Stephen Robertson (2004). “Understanding inverse document frequency: on theoretical arguments for IDF”. In: *Journal of Documentation*.

Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele (2015). “A dataset for movie description”. In: *CVPR*.

Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele (2017). “Movie Description”. In: *IJCV*.

Martin Runz, Maud Buffier, and Lourdes Agapito (2018). “MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects”. In: *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*.

Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, Jitendra Malik, Yanghao Li, and Christoph Feichtenhofer (2023). "Hiera: A Hierarchical Vision Transformer without the Bells-and-Whistles". In: *ICML*.

Sadra Safadoust and Fatma Güney (2023). "Multi-Object Discovery by Low-Dimensional Object Motion". In: *ICCV*.

Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata (2024). "In-context impersonation reveals Large Language Models' strengths and biases". In: *NeurIPS*.

Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen (2019). "Efficient parameter-free clustering using first neighbor relations". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Mattia Savardi, András Bálint Kovács, Alberto Signoroni, and Sergio Benini (2023). "Recognition of Camera Angle and Camera Level in Movies from Single Frames". In: *Proceedings of the ACM International Conference on Interactive Media Experiences Workshops*.

SceneDetect (2025). *SceneDetect*. https://www.scenedetect.com/.

Christian Schmidt, Ali Athar, Sabarinath Mahadevan, and Bastian Leibe (2022). "D^{2}Conv3D: Dynamic Dilated Convolutions for Object Segmentation in Videos". In: *WACV*.

Gerald E. Schneider (1969). "Two Visual Systems". In: *Science*.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello (2023). "Bridging the Gap to Real-World Object-Centric Learning". In: *ICLR*.

Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid (2022). "End-to-end generative pretraining for multimodal video captioning". In: *CVPR*.

Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua (2017). "Video Visual Relation Detection". In: *ACM Multimedia*.

Vivek Sharma, Makarand Tapaswi, M Saquib Sarfraz, and Rainer Stiefelhagen (2019). "Self-supervised learning of face representations for video face clustering". In: *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. (2025). "Vlm-r1: A

stable and generalizable r1-style large vision-language model". In: *arXiv preprint arXiv:2504.07615*.

Shuai Shen, Wanhua Li, Zheng Zhu, Guan Huang, Dalong Du, Jiwen Lu, and Jie Zhou (2021). "Structure-aware face clustering on a large-scale graph with 107 nodes". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Gyungin Shin, Samuel Albanie, and Weidi Wie (2022). "Unsupervised Salient Object Detection with Spectral Cluster Voting". In: *CVPR workshops*.

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi (2023). "What does CLIP know about a red circle? Visual prompt engineering for VLMs". In: *ICCV*.

Nina Shvetsova, Anna Kukleva, Xudong Hong, Christian Rupprecht, Bernt Schiele, and Hilde Kuehne (2023). "HowToCaption: Prompting LLMs to Transform Video Annotations at Scale". In: *arXiv:2310.04900*.

Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce (2021). "Localizing Objects with Self-Supervised Transformers and no Labels". In: *BMVC*.

Karen Simonyan and Andrew Zisserman (2014a). "Two-Stream Convolutional Networks for Action Recognition in Videos". In: *NeurIPS*.

Karen Simonyan and Andrew Zisserman (2014b). "Two-Stream Convolutional Networks for Action Recognition in Videos". In: *NeurIPS*.

Gautam Singh, Yi-Fu Wu, and Sungjin Ahn (2022). "Simple Unsupervised Object-Centric Learning for Complex and Naturalistic Videos". In: *NeurIPS*.

Silky Singh, Shripad Deshmukh, Mausoom Sarkar, and Balaji Krishnamurthy (2023). "LOCATE: Self-supervised Object Discovery via Flow-guided Graph-cut and Bootstrapped Self-training". In: *BMVC*.

Josef Sivic, Mark Everingham, and Andrew Zisserman (2009). ""Who are you?"-Learning person specific classifiers from video". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem (2022). "MAD: A scalable dataset for language grounding in videos from movie audio descriptions". In: *CVPR*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. (2024a). "Moviechat: From dense token to sparse memory for long video understanding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Huihui Song, Tiankang Su, Yuhui Zheng, Kaihua Zhang, Bo Liu, and Dong Liu (2024b). "Generalizable Fourier Augmentation for Unsupervised Video Object Segmentation". In: *AAAI*.

ELIZABETH Spelke (1990). "Principles of object perception." In: *Cognitive science*.

Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely (2019). "Pushing the Boundaries of View Extrapolation with Multiplane Images". In: *CVPR*.

Dhruv Srivastava, Aditya Kumar Singh, and Makarand Tapaswi (2023). "How You Feelin'? Learning Emotions and Mental States in Movie Scenes". In: *CVPR*.

Yanpeng Sun, Jiahui Chen, Shan Zhang, Xinyu Zhang, Qiang Chen, Gang Zhang, Errui Ding, Jingdong Wang, and Zechao Li (2024). "VRP-SAM: SAM with Visual Reference Prompt". In: *CVPR*.

Yihong Sun, Adam Kortylewski, and Alan Yuille (2021). "Amodal Segmentation through Out-of-Task and Out-of-Distribution Generalization with a Bayesian Model". In: *CVPR*.

Lv Tang, Haoke Xiao, and Bo Li (2023). "Can SAM Segment Anything? When SAM Meets Camouflaged Object Detection". In: *arXiv preprint arXiv:2304.04709*.

Yunlong Tang, Junjia Guo, Hang Hua, Susan Liang, Mingqian Feng, Xinyang Li, Rui Mao, Chao Huang, Jing Bi, Zeliang Zhang, Pooyan Fazli, and Chenliang Xu (2024). "VidComposition: Can MLLMs Analyze Compositions in Compiled Videos?" In: *arXiv preprint arXiv:2411.10979*.

Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, and Haizhou Li (2021). "Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection". In: *Proceedings of the 29th ACM international conference on multimedia*.

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen (2012). ""Knock! Knock! Who is it?" probabilistic person identification in TV-series". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Makarand Tapaswi, Marc T Law, and Sanja Fidler (2019). "Video face clustering with unknown number of clusters". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Makarand Tapaswi, Omkar M. Parkhi, Esa Rahtu, Eric Sommerlade, Rainer Stiefelhagen, and Andrew Zisserman (2014). "Total Cluster: A person agnostic clustering method for broadcast videos". In: *Proceedings of the Indian Conference on Computer Vision Graphics and Image Processing*.

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*.

Zachary Teed and Jia Deng (2020). "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow". In: *ECCV*.

Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid (2017). "Learning motion patterns in videos". In: *CVPR*.

Pavel Tokmakov, Cordelia Schmid, and Karteek Alahari (2019). "Learning to segment moving objects". In: *IJCV*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom (2023). "Llama 2: Open Foundation and Fine-Tuned Chat Models". In: *arXiv preprint arXiv: 2307.09288*.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri (2015). "Learning spatiotemporal features with 3d convolutional networks". In: *Proc. ICCV*.

Vladimír Ulman, Martin Maška, Klas E. G. Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, Ihor Smal, Karl Rohr, Joakim Jaldén, Helen M. Blau, Oleh Dzyubachyk, Boudewijn Lelieveldt, Pengdong Xiao, Yuexiang Li, Siu-Yeung Cho, Alexandre C. Dufour, Jean-Christophe Olivo-Marin, Constantino C. Reyes-Aldasoro, Jose A. Solis-Lemus, Robert Bensch, Thomas Brox, Johannes Stegmaier, Ralf Mikut, Steffen Wolf, Fred A. Hamprecht,

Tiago Esteves, Pedro Quelhas, Ömer Demirel, Lars Malmström, Florian Jug, Pavel Tomancak, Erik Meijering, Arrate Muñoz-Barrutia, Michal Kozubek, and Carlos Ortiz-de-Solorzano (2017). "An objective comparison of cell-tracking algorithms". In: *Nature Methods*.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky (2016). "Instance normalization: The missing ingredient for fast stylization". In: *arXiv preprint arXiv:1607.08022*.

Bartolomeo Vacchetti and Tania Cerquitelli (2022). "Cinematographic Shot Classification with Deep Ensemble Learning". In: *Electronics*.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh (2015). "CIDEr: Consensus-Based Image Description Evaluation". In: *CVPR*.

Rishi Veerapaneni, John D. Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine (2019). "Entity Abstraction in Visual Model-Based Reinforcement Learning". In: *CoRL*.

Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i-Nieto (2019). "RVOS: End-to-End Recurrent Network for Video Object Segmentation". In: *CVPR*.

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*.

Angel Villar-Corrales and Sven Behnke (2025). "PlaySlot: Learning Inverse Latent Dynamics for Controllable Object-Centric Video Prediction and Planning". In.

Vlog: Video as A Long Document (2023). https://github.com/showlab/VLog.

Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen (2019). "Feelvos: Fast end-to-end embedding learning for video object segmentation". In: *CVPR*.

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy (2018). "Tracking Emerges by Colorizing Videos". In: *ECCV*.

Haibo Wang, Zhiyang Xu, Yu Cheng, Shizhe Diao, Yufan Zhou, Yixin Cao, Qifan Wang, Weifeng Ge, and Lifu Huang (2024a). "Grounded-VideoLLM: Sharpening Fine-grained Temporal Grounding in Video Large Language Models". In: *arXiv preprint arXiv:2410.03290*.

Hanlin Wang, Zhan Tong, Kecheng Zheng, Yujun Shen, and Limin Wang (2025a). "Contextual AD Narration with Interleaved Multimodal Sequence". In: *CVPR*.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny (2025b). "VGGT: Visual Geometry Grounded Transformer". In: *CVPR*.

Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu (2018). "Bidirectional attentive fusion with context gating for dense video captioning". In: *CVPR*.

John Y. A. Wang and Edward H. Adelson (1994). "Representing Moving Images with Layers". In: *The IEEE Transactions on Image Processing Special Issue: Image Sequence Compression*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin (2024b). "Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution". In: *arXiv preprint arXiv:2409.12191*.

Renhao Wang, Yu Sun, Yossi Gandelsman, Xinlei Chen, Alexei A. Efros, and Xiaolong Wang (2023a). "Test-Time Training on Video Streams". In: *arXiv:2307.05014*.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud (2024c). "DUSt3R: Geometric 3D Vision Made Easy". In: *CVPR*.

Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo (2021a). "End-to-end dense video captioning with parallel decoding". In: *ICCV*.

Teng Wang, Huicheng Zheng, Mingjing Yu, Qian Tian, and Haifeng Hu (2020). "Event-centric hierarchical representation for dense video captioning". In: *IEEE Transactions on Circuits and Systems for Video Technology*.

Wenguan Wang, Xiankai Lu, Jianbing Shen, David J Crandall, and Ling Shao (2019a). "Zero-shot video object segmentation via attentive graph neural networks". In: *ICCV*.

Xiaolong Wang, Allan Jabri, and Alexei A. Efros (2019b). "Learning Correspondence from the Cycle-Consistency of Time". In: *CVPR*.

Xudong Wang, Ishan Misra, Zizun Zeng, Rohit Girdhar, and Trevor Darrell (2023b). "VideoCutLER: Surprisingly Simple Unsupervised Video Instance Segmentation". In: *arXiv preprint arXiv:2308.14710*.

Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz (2022). "Self-supervised Transformers for Unsupervised Object Discovery using Normalized Cut". In: *CVPR*.

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei (2025c). "Multimodal chain-of-thought reasoning: A comprehensive survey". In: *arXiv preprint arXiv:2503.12605*.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. (2025d). "Time-r1: Post-training large vision language model for temporal video grounding". In: *arXiv preprint arXiv:2503.13377*.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He (2026). "π^3: Permutation-Equivariant Visual Geometry Learning". In: *ICLR*.

Yueqian Wang, Xiaojun Meng, Jianxin Liang, Yuxuan Wang, Qun Liu, and Dongyan Zhao (2024d). "HawkEye: Training Video-Text LLMs for Grounding Text in Videos". In: *arXiv preprint arXiv:2403.10228*.

Yujia Wang, Wei Liang, Haikun Huang, Yongqi Zhang, Dingzeyu Li, and Lap-Fai Yu (2021b). "Toward automatic audio description generation for accessible videos". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.

Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia (2021c). "End-to-End Video Instance Segmentation With Transformers". In: *Proc. CVPR*.

Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang (2019c). "Linkage based face clustering via graph convolution network". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. (2022). "Chain-of-thought prompting elicits reasoning in large language models". In: *NeurIPS*.

Marissa A. Weis, Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S. Ecker (2021). "Benchmarking Unsupervised Object Representations for Video Sequences". In: *JMLR*.

Zhengkui Weng, Zhipeng Jin, Shuangxi Chen, Quanquan Shen, Xiangyang Ren, and Wuzhao Li (2021). "Attention-Based Temporal Encoding Network with Background-Independent Motion Mask for Action Recognition". In: *Computational Intelligence and Neuroscience*.

Max Wertheimer (1923). "Untersuchungen zur Lehre von der Gestalt. II". In: *Psychologische forschung*.

Baoyuan Wu, Siwei Lyu, Bao-Gang Hu, and Qiang Ji (2013). "Simultaneous clustering and tracklet linking for multi-face tracking in videos". In: *Proceedings of the IEEE international conference on computer vision*.

Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan (2021). "STAR: A Benchmark for Situated Reasoning in Real-World Videos". In: *NeurIPS*.

Chao-Yuan Wu and Philipp Krahenbuhl (2021). "Towards Long-Form Video Understanding". In: *Proc. CVPR*.

Chaoyi Wu, Jiayu Lei, Qiaoyu Zheng, Weike Zhao, Weixiong Lin, Xiaoman Zhang, Xiao Zhou, Ziheng Zhao, Ya Zhang, Yanfeng Wang, et al. (2023a). "Can GPT-4V(ision) serve medical applications? Case studies on GPT-4V for multimodal medical diagnosis". In: *arXiv preprint arXiv:2310.09909*.

Hao Wu, Huabin Liu, Yu Qiao, and Xiao Sun (2024a). "DIBS: Enhancing Dense Video Captioning with Unlabeled Videos via Pseudo Boundary Enrichment and Online Refinement". In: *CVPR*.

Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum (2015). "Galileo: Perceiving Physical Object Properties by Integrating a Physics Engine with Deep Learning". In: *NeurIPS*.

Junda Wu, Zhehao Zhang, Yu Xia, Xintong Li, Zhaoyang Xia, Aaron Chang, Tong Yu, Sungchul Kim, Ryan A. Rossi, Ruiyi Zhang, Subrata Mitra, Dimitris N. Metaxas, Lina Yao, Jingbo Shang, and Julian J. McAuley (2024b). "Visual Prompting in Multimodal Large Language Models: A Survey". In: *arXiv preprint arXiv:2409.15310*.

Junde Wu, Wei Ji, Yuanpei Liu, Huazhu Fu, Min Xu, Yanwu Xu, and Yueming Jin (2023b). "Medical SAM Adapter: Adapting Segment Anything Model for Medical Image Segmentation". In: *arXiv preprint arXiv:2304.12620*.

Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai (2022). "In Defense of Online Models for Video Instance Segmentation". In: *ECCV*.

Ruizheng Wu, Huaijia Lin, Xiaojuan Qi, and Jiaya Jia (2020). "Memory Selection Network for Video Propagation". In: *ECCV*.

Tz-Ying Wu, Sharath Nittur Sridhar, and Subarna Tripathi (2026). "Harnessing Object Grounding for Time-Sensitive Video Understanding". In: *Proc. WACV*.

Yongliang Wu, Xinting Hu, Yuyang Sun, Yizhou Zhou, Wenbo Zhu, Fengyun Rao, Bernt Schiele, and Xu Yang (2025). "Number it: Temporal Grounding Videos like Flipping Manga". In: *CVPR*.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg (2023c). "SlotFormer: Unsupervised Visual Dynamics Simulation with Object-Centric Models". In: *Proc. ICLR*.

Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg (2023d). "SlotDiffusion: Object-Centric Generative Modeling with Diffusion Models". In: *NeurIPS*.

Yuting Xiao, Yanyu Xu, Ziming Zhong, Weixin Luo, Jiawei Li, and Shenghua Gao (2021). "Amodal Segmentation Based on Visible Region Segmentation and Shape Prior". In: *AAAI*.

Christopher Xie, Yu Xiang, Zaid Harchaoui, and Dieter Fox (2019). "Object discovery in videos as foreground motion clustering". In: *CVPR*.

Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Eshika Khandelwal, Gül Varol, Weidi Xie, and Andrew Zisserman (2025). "Shot-by-Shot: Film-Grammar-Aware Training-Free Audio Description Generation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman (2024a). "AutoAD-Zero: A Training-Free Framework for Zero-Shot Audio Description". In: *ACCV*.

Junyu Xie, Weidi Xie, and Andrew Zisserman (2022). "Segmenting Moving Objects via an Object-Centric Layered Representation". In: *NeurIPS*.

Junyu Xie, Weidi Xie, and Andrew Zisserman (2024b). "Appearance-Based Refinement for Object-Centric Motion Segmentation". In: *ECCV*.

Junyu Xie, Charig Yang, Weidi Xie, and Andrew Zisserman (2024c). "Moving Object Segmentation: All You Need Is SAM (and Flow)". In: *ACCV*.

Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, Raghuraman Krishnamoorthi, and Vikas Chandra (2023). "EfficientSAM: Leveraged Masked Image Pretraining for Efficient Segment Anything". In: *arXiv:2312.00863*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui (2016). "MSR-VTT: A large video description dataset for bridging video and language". In: *CVPR*.

Kai Xu, Tze Ho Elden Tse, Jizong Peng, and Angela Yao (2024). "DAS3R: Dynamics-Aware Gaussian Splatting for Static Scene Reconstruction". In: *arXiv preprint arxiv:2412.19584*.

Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang (2018). "Youtube-vos: A large-scale video object segmentation benchmark". In: *ECCV*.

Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo (2022). "Advancing High-Resolution Video-Language Representation with Large-Scale Video Transcriptions". In: *CVPR*.

Zhiling Yan, Kai Zhang, Rong Zhou, Lifang He, Xiang Li, and Lichao Sun (2023). "Multimodal ChatGPT for medical applications: an experimental study of GPT-4V". In: *arXiv preprint arXiv:2310.19061*.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid (2023a). "Vid2Seq: Large-Scale Pretraining of a Visual Language Model for Dense Video Captioning". In: *arXiv preprint arXiv:2302.14115*.

Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie (2021a). "Self-supervised Video Object Segmentation by Motion Grouping". In: *ICCV*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao (2023b). "Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V". In: *arXiv preprint arXiv:2310.11441*.

Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin (2019a). "Learning to cluster faces on an affinity graph". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Linjie Yang, Yuchen Fan, and Ning Xu (2019b). "Video Instance Segmentation". In: *Proc. ICCV*.

Shu Yang, Lu Zhang, Jinqing Qi, Huchuan Lu, Shuo Wang, and Xiaoxing Zhang (2021b). "Learning Motion-Appearance Co-Attention for Zero-Shot Video Object Segmentation". In: *ICCV*.

Yanchao Yang, Brian Lai, and Stefano Soatto (2021c). "DyStaB: Unsupervised Object Segmentation via Dynamic-Static Bootstrapping". In: *CVPR*.

Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto (2019c). "Unsupervised Moving Object Detection via Contextual Information Separation". In: *CVPR*.

Zhao Yang, Qiang Wang, Luca Bertinetto, Song Bai, Weiming Hu, and Philip H.S. Torr (2019d). "Anchor Diffusion for Unsupervised Video Object Segmentation". In: *ICCV*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang (2023c). "The dawn of LMMs: Preliminary explorations with gpt-4v (ision)". In: *arXiv preprint arXiv:2309.17421*.

Zongxin Yang and Yi Yang (2022). "Decoupling Features in Hierarchical Propagation for Video Object Segmentation". In: *NeurIPS*.

Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely (2022). "Deformable Sprites for Unsupervised Video Decomposition". In: *CVPR*.

Xiaojun Ye, Junhao Chen, Xiang Li, Haidong Xin, Chao Li, Sheng Zhou, and Jiajun Bu (2024). "MMAD: Multi-modal Movie Audio Description". In: *LREC-COLING*.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum (2020). "CLEVRER: Collision Events for Video Representation and Reasoning". In: *Proc. ICLR*.

Zhaoyuan Yin, Jia Zheng, Weixin Luo, Shenhan Qian, Hanling Zhang, and Shenghua Gao (2021). "Learning To Recommend Frame for Interactive Video Object Segmentation in the Wild". In: *CVPR*.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang (2023). "Ferret: Refer and ground anything anywhere at any granularity". In: *arXiv preprint arXiv:2310.07704*.

En Yu, Kangheng Lin, Liang Zhao, Jisheng Yin, Yana Wei, Yuang Peng, Haoran Wei, Jianjian Sun, Chunrui Han, Zheng Ge, et al. (2025). "Perception-r1: Pioneering perception policy with reinforcement learning". In: *arXiv preprint arXiv:2504.07954*.

Keunwoo Peter Yu (2023). *VideoBLIP*. URL: https://github.com/yukw777/VideoBLIP.

Ye Yu, Jialin Yuan, Gaurav Mittal, Li Fuxin, and Mei Chen (2022). "BATMAN: Bilateral Attention Transformer in Motion-Appearance Neighboring Space for Video Object Segmentation". In: *ECCV*.

Haobo Yuan, Xiangtai Li, Tao Zhang, Yueyi Sun, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, and Ming-Hsuan Yang (2025). "Sa2VA: Marrying SAM2 with LLaVA for Dense Grounded Understanding of Images and Videos". In: *arXiv preprint arXiv:2501.04001*.

Yichen Yuan, Yifan Wang, Lijun Wang, Xiaoqi Zhao, Huchuan Lu, Yu Wang, Weibo Su, and Lei Zhang (2023). "Isomer: Isomerous Transformer for Zero-shot Video Object Segmentation". In: *ICCV*.

Zihao Yue, Qi Zhang, Anwen Hu, Liang Zhang, Ziheng Wang, and Qin Jin (2023). "Movie101: A new movie understanding benchmark". In: *Proceedings of the 61st*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius (2023). "Object-Centric Learning for Real-World Videos by Predicting Temporal Feature Similarities". In: *NeurIPS*.

Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi (2022). "MERLOT Reserve: Multimodal Neural Script Knowledge through Vision and Language and Sound". In: *CVPR*.

Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy (2020). "Self-Supervised Scene De-occlusion". In: *CVPR*.

Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong (2023a). "Faster Segment Anything: Towards Lightweight SAM for Mobile Applications". In: *arXiv preprint arXiv:2306.14289*.

Chaoyi Zhang, Kevin Lin, Zhengyuan Yang, Jianfeng Wang, Linjie Li, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang (2024a). "MM-Narrator: Narrating Long-form Videos with Multimodal In-Context Learning". In: *CVPR*.

Hang Zhang, Xin Li, and Lidong Bing (2023b). "Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding". In.

Hang Zhang, Xin Li, and Lidong Bing (2023c). "Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding". In: *ENNLP*.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang (2025). "MonST3R: A Simple Approach for Estimating Geometry in the Presence of Motion". In: *ICLR*.

Kaihua Zhang, Zicheng Zhao, Dong Liu, Qingshan Liu, and Bo Liu (2021). "Deep Transport Network for Unsupervised Video Object Segmentation". In: *ICCV*.

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman (2016). "Video summarization with long short-term memory". In: *ECCV*.

Xiaofeng Zhang, Chaochen Gu, and Shanying Zhu (2023d). "SAM-helps-Shadow:When Segment Anything Model meet shadow removal". In: *arXiv preprint arXiv:2306.06113*.

Yuqi Zhang, Bin Guo, Nuo Li, Ying Zhang, Qianru Wang, and Zhiwen Yu (2023e). "Representation Learning of Next Shot Selection for Vlog Editing". In: *Proceedings of the CVEU Workshop at ICCV*.

Zhenghao Zhang, Shengfan Zhang, Zhichao Wei, Zuozhuo Dai, and Siyu Zhu (2024b). "UVOSAM: A Mask-free Paradigm for Unsupervised Video Object Segmentation via Segment Anything Model". In: *arXiv preprint arXiv:2305.12659*.

Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao (2020). "Where Does It Exist: Spatio-Temporal Video Grounding for Multi-Form Sentences". In: *Proc. CVPR*.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi (2020). "BERTScore: Evaluating Text Generation with BERT". In: *ICLR*.

Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang (2022). "CenterCLIP: Token Clustering for Efficient Text-Video Retrieval". In: *The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang (2023a). "Fast Segment Anything". In: *arXiv preprint arXiv:2306.12156*.

Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar (2023b). "Learning Video Representations from Large Language Models". In: *CVPR*.

Minghang Zheng, Xinhao Cai, Qingchao Chen, Yuxin Peng, and Yang Liu (2024a). "Training-free Video Temporal Grounding using Large-scale Pre-trained Models". In: *arXiv preprint arXiv:2408.16219*.

Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas (2023). "PointOdyssey: A Large-Scale Synthetic Dataset for Long-Term Point Tracking". In: *ICCV*.

Zhuo Zheng, Yanfei Zhong, Liangpei Zhang, and Stefano Ermon (2024b). "Segment Any Change". In: *arXiv:2402.01188*.

Chong Zhou, Chenchen Zhu, Yunyang Xiong, Saksham Suri, Fanyi Xiao, Lemeng Wu, Raghuraman Krishnamoorthi, Bo Dai, Chen Change Loy, Vikas Chandra, and Bilge Soran (2025a). "EdgeTAM: On-Device Track Anything Model". In: *Proc. CVPR*.

Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh (2025b). "R1-Zero's" Aha Moment" in Visual Reasoning on a 2B Non-SFT Model". In: *arXiv preprint arXiv:2503.05132*.

Hongyi Zhou, Yulan Guo, Xiaogang Wang, and Kai Xu (2025c). "MonoMobility: Zero-Shot 3D Mobility Analysis from Monocular Videos". In: *arXiv preprint arXiv:2505.11868*.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. (2025d). "Mlvu:

Benchmarking multi-task long video understanding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu (2022a). "Conditional Prompt Learning for Vision-Language Models". In: *CVPR*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu (2022b). "Learning to prompt for vision-language models". In: *IJCV*.

Luowei Zhou, Chenliang Xu, and Jason J Corso (2018a). "Towards Automatic Learning of Procedures From Web Instructional Videos". In: *AAAI*.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong (2018b). "End-to-end dense video captioning with masked transformer". In: *CVPR*.

Tianfei Zhou, Jianwu Li, Xueyi Li, and Ling Shao (2021). "Target-Aware Object Discovery and Association for Unsupervised Video Multi-Object Segmentation". In: *CVPR*.

Tianfei Zhou, Shunzhou Wang, Yi Zhou, Yazhou Yao, Jianwu Li, and Ling Shao (2020). "Motion-attentive transition for zero-shot video object segmentation". In: *AAAI*.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely (2018c). "Stereo Magnification: Learning View Synthesis using Multiplane Images". In: *SIGGRAPH*.

Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid (2024). "Streaming Dense Video Captioning". In: *CVPR*.

Xingyi Zhou, Anurag Arnab, Chen Sun, and Cordelia Schmid (2025e). "Dense Video Object Captioning from Disjoint Supervision". In: *Proc. ICLR*.

Wangjiang Zhu, Jie Hu, Gang Sun, Xudong Cao, and Yu Qiao (2016). "A Key Volume Mining Deep Framework for Action Recognition". In: *CVPR*.

Yan Zhu, Yuandong Tian, Dimitris Mexatas, and Piotr Dollár (2017). "Semantic Amodal Segmentation". In: *CVPR*.

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee (2023). "Segment everything everywhere all at once". In: *arXiv preprint arXiv:2304.06718*.

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 "Segmenting Moving Objects via an Object-Centric Layered Representation" in Chapter 2.

Paper title	Segmenting Moving Objects via an Object-Centric Layered Representation
Authors	Junyu Xie, Weidi Xie, Andrew Zisserman
Publication status	Published
Publication details	Neural Information Processing Systems (NeurIPS), 2022.

Student Confirmation

Student name	Junyu Xie	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • running of large-scale experiments • writing and presentation of the paper	
Signature and Date	Junyu Xie	15/06/2026

Supervisor Confirmation

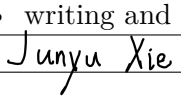
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date	A. Zisserman	17/06/2026

Statement of Authorship for the paper "Appearance-Based Refinement for Object-Centric Motion Segmentation" in Chapter 3.

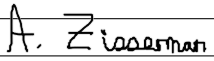
Paper title	Appearance-Based Refinement for Object-Centric Motion Segmentation
Authors	Junyu Xie, Weidi Xie, Andrew Zisserman
Publication status	Published
Publication details	European Conference on Computer Vision (ECCV), 2024.

Student Confirmation

Student name	Junyu Xie	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • running of large-scale experiments • writing and presentation of the paper	
Signature and Date		15/06/2026

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		17/06/2026

Statement of Authorship for the paper "Moving Object Segmentation: All You Need Is SAM (and Flow)" in Chapter 4.

Paper title	Moving Object Segmentation: All You Need Is SAM (and Flow)
Authors	Junyu Xie, Charig Yang, Weidi Xie, Andrew Zisserman
Publication status	Published
Publication details	Asian Conference on Computer Vision (ACCV), 2024.

Student Confirmation

Student name	Junyu Xie	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • running of large-scale experiments • writing and presentation of the paper	
Signature and Date	Junyu Xie	15/06/2026

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	17/06/2026

Statement of Authorship for the paper "Grounding Moving Object Segmentation in 3D Space and Time" in Chapter 5.

Paper title	Grounding Moving Object Segmentation in 3D Space and Time
Authors	Junyu Xie, Tengda Han, Weidi Xie, Andrew Zisserman
Publication status	Under review
Publication details	Submitted to Neural Information Processing Systems (NeurIPS), 2026.

Student Confirmation

Student name	Junyu Xie	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • running of large-scale experiments • writing and presentation of the paper	
Signature and Date	Junyu Xie	15/06/2026

Supervisor Confirmation

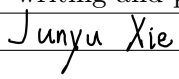
By signing the Statement of Authorship, the supervisor is certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date	A. Zisserman	17/06/2026

Statement of Authorship for the paper "AutoAD-Zero: A Training-Free Framework for Zero-Shot Audio Description" in Chapter 6.

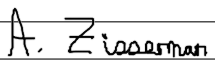
Paper title	AutoAD-Zero: A Training-Free Framework for Zero-Shot Audio Description
Authors	Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, Andrew Zisserman
Publication status	Published
Publication details	Asian Conference on Computer Vision (ACCV), 2024.

Student Confirmation

Student name	Junyu Xie	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • running of large-scale experiments • writing and presentation of the paper	
Signature and Date		15/06/2026

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		17/06/2026

Statement of Authorship for the paper "Shot-by-Shot: Film-Grammar-Aware Training-Free Audio Description Generation" in Chapter 7.

Paper title	Shot-by-Shot: Film-Grammar-Aware Training-Free Audio Description Generation
Authors	Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Eshika Khandelwal, Gül Varol, Weidi Xie, Andrew Zisserman
Publication status	Published
Publication details	International Conference on Computer Vision (ICCV), 2025.

Student Confirmation

Student name	Junyu Xie	
Contribution to the paper	First-author contribution: • conception of research ideas • design and implementation of models • running of large-scale experiments • writing and presentation of the paper	
Signature and Date	Junyu Xie	15/06/2026

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	17/06/2026

Statement of Authorship for the paper "As the Story Unfolds: Watching a Film and Identifying Characters as a Human Does" in Chapter 8.

Paper title	As the Story Unfolds: Watching a Film and Identifying Characters as a Human Does
Authors	Zhongrui Gui, Junyu Xie, Tengda Han, Weidi Xie, Andrew Zisserman
Publication status	Under review
Publication details	Submitted to Neural Information Processing Systems (NeurIPS), 2026.

Student Confirmation

Student name	Junyu Xie	
Contribution to the paper	Second-author contribution: • conception of research ideas • writing and presentation of the paper	
Signature and Date	Junyu Xie	15/06/2026

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	17/06/2026