# Efficient Multimodal Video Representation Learning Through Language

Thesis submitted in partial fulfilment
of the requirements for the degree of

*Master of Science in* ***Computer Science and Engineering*** *by Research*

by

Darshan Singh S
2022701013
darshan.singh@research.iiit.ac.in

Advisors: Prof. C. V. Jawahar and Prof. Makarand Tapaswi

International Institute of Information Technology
Hyderabad - 500 032, INDIA

December 2024

International Institute of Information Technology

Hyderabad, India

# CERTIFICATE

It is certified that the work contained in this thesis, titled "Efficient Multimodal Video Representation Learning Through Language" by Darshan Singh S, has been carried out under my supervision and is not submitted elsewhere for a degree.

| | |
|---|---|
| December 2024 | Advisors: Prof. C. V. Jawahar and Prof. Makarand Tapaswi |

To my Parents (Vinutha, Sudhir)

# Acknowledgments

## Gratitude

My journey at IIIT Hyderabad has been the most transformative period of my life. During my time here, I have faced more failures than successes, but as the saying goes, you only need to succeed once. I've hit rock bottom many times, but thanks to the incredible professors and peer support, I've been able to rise each time.

First and foremost, I want to express my deepest gratitude to Prof. Jawahar and Prof. Makarand. Prof. Jawahar's honest feedback and invaluable insights have shaped my research career. He was the one who introduced me to the world of research, and I will forever be indebted to him for the guidance and opportunities he provided. Prof. Makarand has been a role model for me. While his research skills are well-known, more than that, he is one of the most amazing human beings you will ever meet. He truly has a beautiful mind as well as a beautiful heart. I look up to him and aspire to be like him. I wouldn't have come this far without him. I am incredibly fortunate to have had both Prof. Jawahar and Prof. Makarand as my advisors. I would also like to thank Prof. Vineet Gandhi for his unwavering support and guidance.

I am immensely grateful to my CVIT labmates, who have been my backbone throughout this journey. The bond we share feels more like family. I want to thank Varun, Darshana, Balu, Hardik, Haran, Pranav, Ravi, Rudrabaha, Seshadri, Aditya Arun, Avijit, Aditya Agarwal, Bipasha, Siddhant, Shivanshu, and Madhav. Special thanks to Zeeshan, who has been like a big brother to me—I always to this date consult him before making any decision. Soumya, thank you for your constant encouragement and motivation. Siddharth, thank you for being someone I could always share everything with. Anagha, thanks for being a wonderful friend, travel buddy, and movie partner. A heartfelt thank you to Aparajitha for being a constant source of support. Talking to you always made me feel better. Thanks to Soumya (CSTAR) for your support.

Dhruv and Rodo, I owe you both so much for having my back during the most challenging times of my life. You both have been there with me (quite literally) at the most difficult time of my life. Words cannot express how grateful I am to have you in my life. Manu, thank you for lightening tense moments with your awful jokes—they always managed to divert my mind, even if only to wonder how anyone could make such bad jokes! I'd also like to thank Joy, Vivek, Kesav, Deb, Anindita, Jhansi, Jayaram, Harsha, Sanket, Bhoomendra, Dhaval, Anchit, BVK, Nayan, Nancy, Amruth, Naren, Kawshik, Vansh, and Akshat for making my time at IIIT such an incredible experience.

Thank you to my school friends, Aditya and Ronak, for your unwavering support and for sticking with me all these years.

Lastly, I want to thank my brother Tarun for always being there for me, and my sister Sameeksha, who ensured I never missed a meal, constantly looking out for my well-being. My heartfelt gratitude to my entire family for their unwavering support throughout this journey. And, of course, thanks to Amma and Daddy, though it almost seems redundant to thank them - they're as much a part of me as my own thoughts and feelings.

As I leave IIIT, I am filled with only gratitude. Gratitude for the opportunity to be here. Gratitude for the strength I discovered within myself during the darkest moments, and for the unwavering support from my professors and friends, who became my guiding light along the way. This place has been more than just an academic institution. This place has become my home.

I want to end with one of my favorite quotes from *Harry Potter*:

> "Happiness can be found in the darkest of times, if one only remembers to turn on the light."
>
> – Albus Dumbledore

# Abstract

This work presents several contributions to video representation learning and related multimodal tasks, addressing key challenges in datasets, efficient model adaptation using less data, and compositional and fine-grained visual understanding.

Despite the rapid growth of online lecture videos in the past several years, video-language research has primarily focused on instructional videos/movies, resulting in a scarcity of specialized datasets for educational lecture videos. To address this, we first introduce **AVLectures**, a large-scale dataset of STEM lecture videos. It consists of 86 courses with over 2,350 lectures covering various STEM subjects. Each course contains video lectures, transcripts, OCR for lecture frames, and optionally lecture notes, slides, assignments, and related educational content that can inspire a variety of tasks. Next, we propose a novel unsupervised temporal segmentation task to segment lecture videos into bite-sized topics. We show that multimodal cues can be effectively utilized to learn lecture-aware representations for this task, facilitating a richer analysis of educational content.

Next, we address the inefficiency of adapting pre-trained models like CLIP to videos. Existing methods typically rely on large-scale, sparsely annotated video caption datasets, resulting in slow and data-intensive adaptation. We propose **SRL-CLIP**, a novel approach that leverages the rich, structured semantic information within Semantic Role Labels (SRLs) for highly *efficient* adaptation. We use VidSitu for adaptation as it provides dense SRL annotations that holistically represent the entire video. SRL-CLIP achieves comparable or superior performance on various video understanding benchmarks (zero-shot retrieval, situation recognition, dense video captioning, and localization) compared to state-of-the-art models that possess $4{-}8\times$ more parameters and are post-pretrained on up to $4000\times$ times more data.

To further explore the models' understanding of visual content, we introduce three novel benchmarks. First, **VELOCITI** evaluates the compositional reasoning abilities of video-language models, focusing on their ability to bind semantic concepts through time. Second, we introduce **NDLB**, a frame-

work aimed at improving fine-grained image captioning, which uses self-retrieval as a key component along with a new benchmark to check if the model can capture subtle visual distinctions. Finally, we introduce $\mathbf{D_3}$, a benchmark specifically designed to evaluate the fine-grained visual discrimination capabilities of MLLMs using self-retrieval, further pushing the boundaries of fine-grained visual understanding.

These contributions, which include novel datasets, efficient training recipes, and insightful benchmarks, collectively advance the state of the art in multimodal and video representation learning.

# Contents

# List of Tables

# List of Figures

*Chapter 1*

# Introduction

Video and image content have become increasingly prevalent across various domains over the past several years, driving significant advances in the field of visual representation learning. From content retrieval to content generation, visual representation learning has become a cornerstone of numerous multimedia applications. In this thesis we propose novel approaches in visio-linguistic understanding, focusing on five key areas: educational video understanding, efficient video adaptation of vision-language models, compositional understanding in video-language tasks, fine-grained image captioning, and detailed visual difference detection.

Due to the exponential growth of online educational content, particularly lecture videos, students have a pressing need for tools to help them navigate and maximize the value of these resources. AVLectures [8] addresses this challenge by introducing a large-scale dataset of STEM lecture videos and proposing a novel video lecture segmentation task that leverages multimodal cues and unsupervised learning techniques. Lecture clip representations leverage visual, textual, and OCR cues and are trained on a pretext self-supervised task of matching the narration with the temporally aligned visual content. By focusing on the increasingly important domain of educational videos, AVLectures fills a gap in video-language research for educational content and provides valuable resources for developing tools to enhance online learning experiences.

Moving beyond the educational domain, we show how to efficiently adapt pre-trained vision-language models like CLIP for video understanding [9]. However, relying solely on narrations or captions for post-pretraining can be inefficient due to their inherent sparsity. SRL-CLIP [9] addresses this limitation by utilizing Semantic Role Labels (SRLs), which provide a more holistic and structured representation of video content. By generating captions from these dense SRL annotations, SRL-CLIP demonstrates efficient adaptation of CLIP with significantly less data than existing methods, achieving comparable or superior performance on various downstream tasks, highlighting the importance of structured, holistic video representations in improving model efficiency and generalization.

It is becoming increasingly important to assess whether video understanding models are truly capable of capturing video's complex compositional nature as they grow more sophisticated. VELOCITI [10] introduces a novel benchmark designed to evaluate the compositional understanding capabilities of video-language models, particularly focusing on perception and binding which is the ability to link entities through relationships within a video. VELOCITI highlights the current shortcomings of state-of-the-art models in this crucial aspect of video understanding.

There has been a persistent challenge in the realm of image captioning when it comes to generating fine-grained, accurate descriptions since they are trained on data that is either noisy (alt-text) or generic (human annotations). This is further amplified by maximum likelihood training that encourages generation of frequently occurring phrases. Previous approaches have used self-retrieval (SR) fine-tuning for captioners to address this but it often reduce caption faithfulness and cause hallucinations. To tackle this, we improve the initial MLE training of the captioning system and implement a curriculum-based approach for SR fine-tuning in this work [11]. We do this by introducing a novel framework to enhance the granularity of image captioning datasets while maintaining fidelity to human annotations. Next, we propose BagCurri, a curriculum for self-retrieval fine-tuning that balances fine-grained detail with caption faithfulness. Lastly, we introduce a novel benchmark to measure captioning models for capturing subtle differences between similar images. Our work emphasizes the importance of both training methodologies and evaluation metrics in achieving fine-grained understanding.

Similarly, accurately assessing the visual understanding capabilities of Multimodal Large Language Models (MLLMs) remains a challenge. In $D_3$ [12] we introduce a new benchmark focused on evaluating the ability of MLLMs to detect, describe, and discriminate fine-grained visual differences between highly similar image pairs. Using a self-retrieval approach, $D_3$ provides a white-box evaluation methodology that reveals the limitations of current MLLMs in discerning subtle visual details.

In summary, these works represent significant strides in video representation learning and related multimodal tasks, contributing valuable datasets, methods, and benchmarks that address challenges in data sparsity, fine-grained understanding, and compositional reasoning.

## 1.1 Contributions

- **AVLectures** [8]: Introduced a large-scale dataset of STEM lecture videos and a novel unsupervised lecture segmentation task, demonstrating improved segmentation performance using multimodal cues and self-supervised learning.

- **SRL-CLIP** [9]: Proposed an efficient CLIP adaptation method for video understanding using Semantic Role Labels, achieving state-of-the-art performance on several downstream tasks while minimizing the need for extensive post-pretraining.

- **VELOCITI** [10]: Introduced a benchmark to evaluate compositional understanding in video-language models, revealing limitations in current models' ability to bind semantic concepts and highlighting directions for future research. This work was jointly done with Darshana Saravanan and Varun Gupta.

- **NDLB** [11]: Developed a framework for enhancing datasets with fine-grained visual information, a new benchmark for evaluating captioning models, and a curriculum learning strategy for self-retrieval fine-tuning, leading to improved performance in fine-grained image captioning.

- **$D_3$** [12]: Introduced a benchmark for evaluating the fine-grained visual discrimination capabilities of MLLMs using a self-retrieval approach.

The NDLB and $D_3$ works were primarily led by Manu Gaur, with me contributing in a supporting role. The thesis will primarily focus on AVLectures and SRL-CLIP, as these represent my key individual works. The other contributions (VELOCITI, NDLB, and $D_3$) were completed in collaboration and will be discussed briefly in Chapter 4.

## 1.2 Related Publications

- **Unsupervised Audio-Visual Lecture Segmentation**, <u>*Darshan Singh S*</u>*, Anchit Gupta*, C.V. Jawahar, Makarand Tapaswi. *In IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2023.*

- **Seeing Beyond Captions: Efficient CLIP Video Adaptation via Structured Semantic Role Labels**, <u>*Darshan Singh S*</u>, Zeeshan Khan, Makarand Tapaswi. *(under review)*.

* indicates equal contribution

- **VELOCITI: Can Video-Language Models Bind Semantic Concepts through Time?**, Darshana Saravanan*, *Darshan Singh S**, Varun Gupta*, Zeeshan Khan, Vineet Gandhi, Makarand Tapaswi. *(under review)*.
- **No Detail Left Behind: Revisiting Self-Retrieval for Fine-Grained Image Captioning**, Manu gaur, *Darshan Singh S*, Makarand Tapaswi. *(under review)*.
- **Detect, Describe, Discriminate: Moving Beyond VQA for MLLM Evaluation**, Manu gaur, *Darshan Singh S*, Makarand Tapaswi. *In ECCVW'24, Emergent Visual Abilities and Limits of Foundation Models (EVAL-FoMo)*.

## 1.3 Related works: Educational Video Understanding

**Applications in educational videos.** Research in video-language domain has focused primarily on movies [13, 14, 15], and instructional videos [16, 17, 18], especially cooking videos [19, 20]. However, there are a few isolated works [21, 22, 23, 24, 25, 26] that attempt to solve various problems in the education domain that we highlight below. Mahapatra *et al*. [25] propose an approach to generate a hierarchical table of contents for a lecture video using multimodal information such as transcripts and associated metadata from video key frames. In the direction of localizing and recognizing text on a blackboard, Dutta *et al*. [23] introduce LectureVideoDB, a dataset consisting of frames from multiple lecture videos (including blackboard). Bulathwela *et al*. [21, 22] introduce datasets to understand learner engagement with educational videos.

**Joint representation learning of video and language.** In this section, we review popular works that address joint representation learning in video and language. A common self-supervised objective used to learn good representations is aligning video with its corresponding narrations [27, 17], which can then be used for a number of downstream tasks, such as text-to-video retrieval [28, 29, 17], visual question answering [30, 15, 31], video captioning [32, 33, 34], natural language guided video summarization [35] among others. Typically, representations from off-the-shelf pre-trained visual and language models are improved via a joint video-text embedding trained on the alignment task [17]. Recent approaches [36, 28, 29] also adopt Transformer-based models that learn in an end-to-end manner from raw video pixels.

**Temporal video segmentation.** In temporal segmentation, fully supervised [37], weakly supervised [38, 39], and unsupervised [40, 16, 41, 42] approaches have been explored. In the unsupervised space, instructional videos are segmented by finding and grouping direct object relations in the narrations [16] or through the use of frame-level features that incorporate relative temporal information followed by K-means clustering (CTE) [41]. Proxy tasks such as future frame prediction are also used to perform temporal segmentation [40]. Recently, a temporally weighted version of a 1-nearest neighbor clustering algorithm is proposed to produce temporally consistent clusters (TW-FINCH) [42].

## 1.4 Related works: CLIP Video Adaptation

Several attempts have been made to adapt CLIP for videos [43, 44, 45, 2, 46, 47, 48, 49, 50, 51]. However, they are primarily focused on task-specific adaptation for action recognition [44, 47] or text-to-video retrieval [43, 46, 50, 2, 45, 51]. Task specific adaptations may lead to the loss of generalized representations. Different from above, we propose task-agnostic adaptation of CLIP, and focus on extending CLIP to videos using holistic video understanding datasets. This allows our model to be applied on a diverse range of video understanding tasks that demand different levels of perceptual granularity.

We discuss related works in two dimensions: (i) datasets used for adapting CLIP; and (ii) approaches for adapting CLIP for videos.

**1. Popular VL datasets for adapting CLIP.** The unparalleled success of training Large Language Models (LLMs), *e.g*. GPT2 [52], LLaMa [53], with massive datasets has brought similar trends to the vision community.

**Towards text-to-video retrieval.** Large-scale web scraped datasets such as HowTo100M [3] and HD-VILA [54] align video clips with narrations. More recently, WebVid-2.5M [55] was curated from stock footage with textual descriptions resulting in better captions, that align with the video. These datasets are used by several methods for adapting CLIP or training a video-text contrastive alignment approach from scratch [2, 43, 46, 56, 57].

**Towards action understanding.** Datasets typically consists of 10-second videos annotated with a single action. To adapt CLIP using supervised video-text contrastive learning, the widely used idea is to create CLIP-like prompts from the action labels [47, 44, 45] on datasets like Kinetics [58] or Something-Something [59].

**Towards holistic video understanding.** Action labels [58] or narrations [3] fail to capture the complex, hierarchical, multifaceted aspects of videos. To learn holistic and fine-grained representations, we propose to exploit the SRL annotations in VidSitu [5]. We observe that SRL-CLIP consistently outperforms base CLIP on multiple video understanding tasks that require varying degrees of granularity, indicating that it has improved holistic reasoning abilities. Moreover, a model trained on SRL captions outperforms methods that are post-pretrained on large-scale web video datasets (often 2-3 orders of magnitude larger) [3, 54, 55] on *zero-shot* text-to-video retrieval, indicating improved vision-language alignment.

**2. Approaches for adapting CLIP** for videos can be classified into direct fine-tuning or through adapters.

**Fine-tuning approaches** typically follow frame-level feature extraction from CLIP followed by temporal aggregation. The resulting video feature is aligned with the corresponding text prompt via contrastive loss. Either partial, or all the parameters of CLIP are fine-tuned [44, 47]. While a majority of the methods [44, 45, 2, 50] use a Transformer [60] for temporal aggregation of frame-level features, simple mean pooling has been shown to be effective for a narrow task of action recognition [47]. Other approaches use a weighted-mean of frame embeddings based on query-scoring [61]; compute frame-level attention based on text [62]; integrate temporal aggregation within the image encoder [46]; or suggest using a temporal model in parallel to the image encoder [63].

**Adapters**, the alternative to finetuning, are lightweight modules injected between layers of a pretrained model for efficient adaptation on a downstream task [64, 65, 66, 67]. The original parameters are usually frozen, and only the adapter is trained, allowing for efficient adaptation. Prior works in CLIP adaptation have used spatial adapters [68], spatio-temporal adapters [69], and cross-modal adapters [70] for efficient adaptation to downstream tasks. Recently there has been a surge of methods using low-rank adapters (LoRA) [65] and advances [71, 72] for their high efficiency enabled by low rank learnable matrices during training and no computational overhead during inference. We follow this approach instead of fine-tuning, allowing us to efficiently post-pretrain CLIP's `ViT-B/32` image and/or text encoders on a single 12GB GPU.

## 1.5 Organization of the thesis

The rest of the thesis is organized as follows:

1. In Chapter 2, we discuss AVLectures: How to learn lecture-aware representations and use them for downstream tasks such as temporal lecture segmentation?
2. In Chapter 3, we discuss SRLCLIP: How to efficiently Adapt CLIP for general and holistic video understanding using structured semantic role labels? and finally
3. In Chapter 4, we examine the capabilities and limitations of current multimodal models across a range of challenging tasks. We investigate fine-grained image captioning with NDLB, exploring the models' ability to generate detailed and specific descriptions. Further, we assess their capacity for subtle visual discrimination using the $D_3$benchmark. Finally, we probe their compositional reasoning abilities in the context of video understanding with VELOCITI.

*Chapter 2*

# AVLectures: Unsupervised Audio-Visual Lecture Segmentation

The last decade has seen a significant increase in online lectures in the form of Massive Open Online Courses (MOOCs) through platforms such as Coursera or EdX. Many high-quality recorded lectures are also published online, *e.g.*, MIT through MIT OpenCourseWare (OCW)[1], top Indian universities through NPTEL[2], and several professors that make their lectures publicly available[3]. This increase in online content is considered one of the biggest turning points in the history of education as anybody can learn any topic from the world's leading teachers from the comfort of their home [73, 24]. As the world moved to an online mode during the pandemic, there is absolutely no doubt that such online lecture content creation will only increase.

Creating an online course requires tremendous effort from the instructor and teaching assistants. Apart from designing and preparing the content itself, the mode of presentation poses challenges including segmenting the large videos into smaller topics to enhance the learning experience, adding quiz-like questions during the lecture to retain the student's engagement, summarizing the lecture at the end, *etc*. These tasks require carefully combing through the lecture several times, a time-consuming and error-prone process. Our goal is to encourage the community to address these tasks automatically or at least provide automatic recommendations for a human-in-the-loop system as they have the potential to reduce instructor's efforts, giving them more time and energy to improve the lecture content.

To build such solutions, machine understanding of audio-visual (AV) lectures is crucial. However, currently, there are no large-scale datasets of audio-visual lectures[4]. Our *first contribution* is *AVLectures*, a large-scale dataset to facilitate research in automatic understanding of lecture videos. By releas-

[1]MIT-OCW - https://ocw.mit.edu/

[2]NPTEL - https://nptel.ac.in/

[3]*e.g.* Statistics 110 or Stanford's CS231n.

[4]Despite educational videos being the fourth most consumed content on the Internet according to this survey, just behind "How-to" videos.

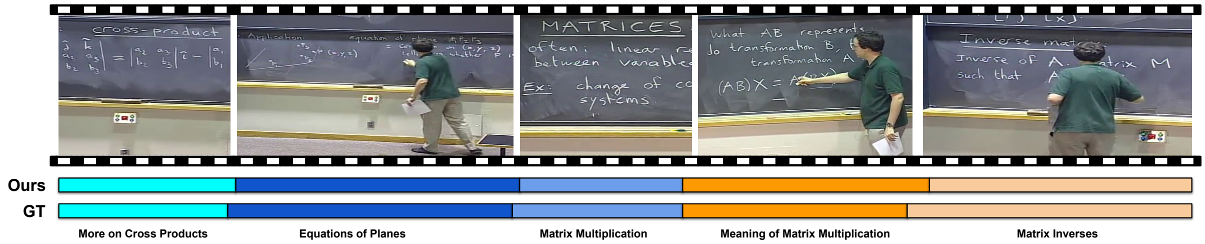



Figure 2.1: We address the task of lecture segmentation in an unsupervised manner. We show an example of a lecture segmented using our method. Our method predicts segments close to the ground-truth. Note that our method *does not predict the segment labels*, they are only shown so that the reader can appreciate the different topics.

ing *AVLectures*, we wish to ignite research in the largely overlooked applications in education to help manage the fast-growing online lecture content.

Our *second contribution* is the formulation and benchmarking of the *lecture segmentation* task, where, given a long video lecture, our goal is to temporally segment it into smaller bite-sized topics. Lecture segmentation can be more challenging than scene segmentation in movies [74] or cooking videos [41] as the differences across segments are subtle, in both the visual and transcribed narrations. For example, Fig. 2.1 shows a professor teaching on the blackboard and walking along the podium. A model trained on movies or instructional videos may find it hard to segment the lecture as the objects or actions in the video do not change significantly. Across segments, the visual boundaries are subtle changes such as clearing the board, while the narration may see a shift in the overall topic of discussion.

We propose lecture segmentation as an unsupervised task that leverages visual, textual, and OCR cues from the audio-visual lecture. We first split the lecture into small clips and extract each clip's visual and textual features using pre-trained models. To make our representations lecture-aware, we learn a joint text-video embedding in a self-supervised manner by matching the narration with the aligned visual content. Finally, we obtain clusters using a temporally consistent[5] 1-nearest neighbor algorithm, TW-FINCH [42].

We pick lecture segmentation as our first use case based on an insightful large-scale study conducted on the EdX platform [76]. They find that students who successfully complete an online course typically spend 4.4 minutes on a 12-15 minute long lecture clip, clearly demonstrating the need for simplified

[5]Temporally *consistent* here refers to temporally *contiguous*, *i.e*. the segment membership of clips looks like [0, 0, 1, 1, 1, 2, 2] rather than [0, 1, 0, 2, 2, 1, 1]. TW-FINCH [42] allows this over base FINCH [75].

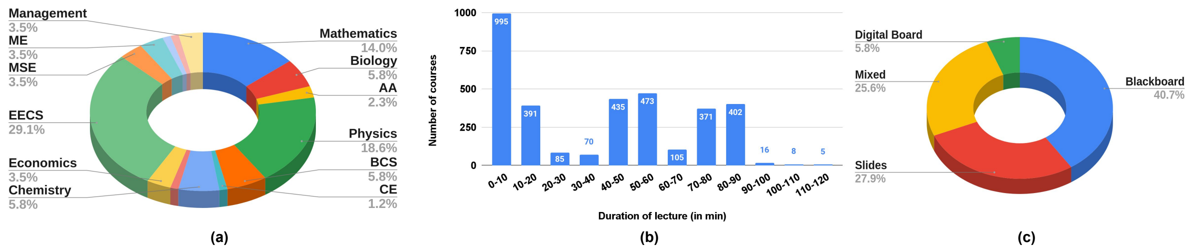



Figure 2.2: AVLectures statistics. (a) **Subject areas.** ME: Mechanical Eng., MSE: Materials Science and Eng., EECS: Electrical Eng. and Computer Science, AA: Aeronautics and Astronautics, BCS: Brain and Cognitive Sciences, CE: Chemical Eng. (b) **Lecture duration** distribution. (c) **Presentation modes** distribution.

navigation of long clips. Lecture segmentation is also a first step towards creating a multimodal table of contents to summarize a lecture [26]. Finally, there is evidence for segmentation to assist in enabling non-linear video consumption [77] and efficient previewing [78, 79, 80]. While segmentation is our first task, we emphasize that *AVLectures* can be used for various other tasks in the future such as generating automatic quizzes for the lecture, aligning lecture videos with the notes enabling generation of lecture notes, retrieving relevant clips of the lecture using text queries, summarizing long lecture videos, retrieving and aligning similar courses/lectures from different learning platforms, and many more.

Our key contributions are summarized below. (i) We introduce a novel educational audio-visual lectures dataset, *AVLectures*, that can facilitate several applications in the education domain. (ii) We formulate and benchmark the problem of *unsupervised lecture segmentation*. We show that self-supervised multimodal representations learned by matching the narration with temporally aligned video clips greatly help the task of segmentation. (iii) Our method outperforms several baselines. We also provide extensive ablation studies to understand prominent factors leading to the success of our approach. We will release code and data.

## 2.1 The AVLectures Dataset

We introduce *AVLectures*, a large-scale educational audio-visual lectures dataset to facilitate research in the domain of lecture video understanding. The dataset comprises of 86 courses with over 2,350 lectures for a total duration of 2,200 hours. Each course in our dataset consists of video lectures, corre-

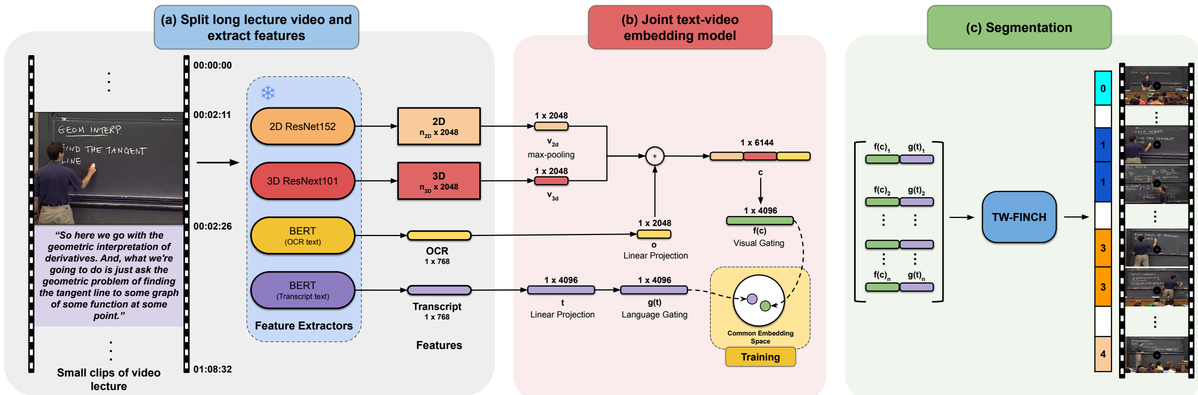


Figure 2.3: **Segmentation pipeline.** (a) *Video clip and feature extraction pipeline* used to extract visual and textual features from small clips of 10s-15s duration. The feature extractors are frozen and are not fine-tuned during the training process. (b) *Joint text-video embedding model* learns lecture-aware representations. (c) *Lecture segmentation process*, where we apply TW-FINCH at a clip-level to the learned (concatenated) visual and textual embeddings obtained from (b).

sponding transcripts, OCR outputs for frames, and optionally lecture notes, slides, and other metadata, making our dataset a rich multi-modality resource.

Courses span a broad range of subjects, including Mathematics, Physics, EECS, and Economics (see Fig. 2.2a). While the average duration of a lecture in the dataset is about 55 minutes, Fig. 2.2b shows a significant variation in the duration. We broadly categorize lectures based on their presentation modes into four types: (i) Blackboard, (ii) Slides, (iii) Digital Board, and (iv) Mixed, a combination of blackboard and slides. Fig. 2.2c depicts a healthy distribution of presentation modes in our dataset.

**Courses with Segmentation.** Among the 86 courses in AVLectures, a significant subset of 15 courses also have temporal segmentation boundaries. We refer to this subset as the *Courses with Segmentation* (CwS) and the remainder 71 courses as the *Courses without Segmentation* (CwoS).

### 2.1.1 Dataset Collection Procedure

Our dataset is primarily sourced from MIT-OCW [81]. We curated a list of courses by browsing the OCW website and used web scraping tools to download the video lectures and accompanying metadata such as narration transcripts, assignments, lecture notes/slides, *etc*. Non-lecture videos (*e.g.* instructor interviews) that were found in some courses are manually discarded. We process and store the OCR outputs of video frames in each lecture using Google Cloud Vision API. As sudden changes in the visual content of a lecture are rare, we process one frame at every 10 seconds.

### 2.1.2 Curating the Lecture Segmentation Dataset

It is shown that partitioning a long duration lecture into shorter topic-based clips helps in capturing students' attention and improves the overall learning experience [76, 77]. However, manually segmenting lecture recordings is a time-consuming and costly task. To evaluate automatic methods for lecture segmentation, we create a subset of our dataset, called *Courses with Segmentation* (CwS), that includes courses in which long lecture videos are segmented into multiple smaller clips. We curate 15 such courses with 350 lectures in total, where temporal segmentation *ground-truth* (for each lecture) is obtained in one of two ways. (i) Out of the 15 courses, 5 courses[6] have topics in the table of contents that refer to various temporal segments in a long lecture video. We obtain the segmentation timestamps for such courses directly by web scraping. (ii) The rest of the 10 courses[7] have concepts that are presented as pre-segmented short videos. Here, we re-assemble the small segments to build the original complete lecture. We trim the intro and outro from short video clips to avoid biasing the models to identify the segments easily.

## 2.2 Lecture Segmentation

Our lecture segmentation approach involves three stages (Fig. 2.3). In the first stage, we extract features from diverse modalities of the lecture (Sec. 2.2.1 and Fig. 2.3a). In the second stage, we learn lecture-aware representations by aligning the visual content with the corresponding narration using self-supervision (Sec. 2.2.2 and Fig. 2.3b). Finally, we perform segmentation using TW-FINCH [42] on the learned representations (Sec. 2.2.3 and Fig. 2.3c).

### 2.2.1 Video clip feature extraction

We divide a lecture into small clips of 10-15 seconds while ensuring that subtitles are not split. This clip is a basic unit for segmentation, *i.e.* segmentation boundaries can be placed before or after, not in

[6](i) *e.g.* Single Variable Calculus
[7](ii) *e.g.* Classical Mechanics

between. The chosen duration is small enough to not introduce boundary errors for segmentation but big enough to contain meaningful information about the lecture, as will also be shown empirically.

**Video feature extraction.** The visual clip representation consists of three feature types: OCR, 2D, and 3D. The *OCR feature* encodes the output text from an OCR API using the BERT sentence transformer model. Specifically, we use MPNet (`all-mpnet-base-v2`) [82, 83] from HuggingFace to obtain a 768-dimensional vector that captures the semantic information of the recognized text. The *2D and 3D features* are extracted using a video feature extraction pipeline [17]. An ImageNet pre-trained Resnet-152 [84] model produces 2D features at 1 fps while the 3D features are extracted using the Kinetics [85] pre-trained ResNeXt-101 [86] to obtain 1.5 features per second. We apply max-pooling across the temporal dimension to obtain 2048-dimensional vectors, $\mathbf{v_{2d}}$ and $\mathbf{v_{3d}}$ respectively.

**Text feature extraction** uses the same model as used for OCR. The text feature encodes the instructor's spoken words or subtitles corresponding to each video clip.

### 2.2.2 Learning joint text-video embeddings

Our approach transforms features from off-the-shelf models into lecture-aware embeddings and is inspired by popular works on instructional videos [17, 18].

**Model architecture.** Fig. 2.3b depicts our model used to learn lecture-aware embeddings by matching the visual feature of a clip with its corresponding text pair. We first extract the visual and textual features for a video clip $C$ and transcript (text) $T$ using the feature extraction pipelines described above. We pass the OCR feature through a fully-connected layer to obtain a 2048-dimensional vector $\mathbf{o}$, and concatenate it with $\mathbf{v_{2d}}$ and $\mathbf{v_{3d}}$ to form a 6144-dimensional vector $\mathbf{c}$ describing the clip $C$. Similarly, the text feature vector (output of the transformer) is passed through a fully connected layer to obtain a 4096-dimensional vector $\mathbf{t}$, representing text $T$. Next, we learn a projection using the non-linear context gating [87, 17] defined as follows:

$$f(\mathbf{c}) = (W_1^c\mathbf{c} \,+\, b_1^c) \;\odot\; \sigma(W_2^c(W_1^c\mathbf{c} \,+\, b_1^c) \,+\, b_2^c)\,, \tag{2.1}$$

$$g(\mathbf{t}) = (W_1^t\mathbf{t} \,+\, b_1^t) \;\odot\; \sigma(W_2^t(W_1^t\mathbf{t} \,+\, b_1^t) \,+\, b_2^t)\,, \tag{2.2}$$

where $W_1^c, W_2^c, W_1^t, W_2^t$ and $b_1^c, b_2^c, b_1^t, b_2^t$ are learnable parameters, $\odot$ is element-wise multiplication and $\sigma$ is an element-wise sigmoid. $f(\mathbf{c})$ and $g(\mathbf{t})$ are 4096-dimensional embeddings, which are used later for the segmentation task.

**Loss function.** We train our embedding model's parameters with the max-margin ranking loss [88, 89]. Specifically, we consider the (cosine) similarity score between a clip $C_i$ and transcript $T_j$ as $s_{ij} = \langle f(\mathbf{c}_i), g(\mathbf{t}_j)\rangle$. We loop over paired samples of a mini-batch $\mathcal{B}$ and compute the loss as

$$\sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{N}(i)} \max(0, \delta + s_{ij} - s_{ii}) + \max(0, \delta + s_{ji} - s_{ii})\,, \tag{2.3}$$

where $s_{ii}$ corresponds to a positive (aligned) clip-transcript pair $(C_i, T_i)$ and should score high, while $\mathcal{N}(i)$ is the set of negative pairs such that half the negative pairs are from the same lecture and act as hard negatives, while the others stem from other lectures [90, 17]. Our mini-batch size is $|\mathcal{B}| = 32$ and the margin is set at $\delta = 0.1$.

### 2.2.3 Lecture segmentation with learned embeddings

We extract clip and transcript embeddings from our joint text-video model and concatenate them to obtain an overall representation $\phi_i = [f(\mathbf{c}_i), g(\mathbf{t}_i)]$. All such representations of a lecture with $N$ clips, $\{\phi_1, \dots, \phi_N\}$, are passed to the TW-FINCH algorithm [42] that encodes feature similarity and temporal proximity as a 1-nearest-neighbor graph and produces a clustering as shown in Fig. 2.3c. Specifically, we denote the feature similarity between clips as $E_s$ and temporal proximity as $E_\tau$.

$$E_s(m, n) = \begin{cases} 1 - \langle \phi_m, \phi_n \rangle & \text{if } m \neq n\,, \\ 1 & \text{otherwise}\,. \end{cases} \tag{2.4}$$

$$E_\tau(m, n) = \begin{cases} |\tau_m - \tau_n|/T & \text{if } m \neq n\,, \\ 1 & \text{otherwise}\,, \end{cases} \tag{2.5}$$

where $m, n \in [1, \dots, N]$, $\tau_m$ and $\tau_n$ are timestamps for the clips $m$ and $n$ and $T$ is the total lecture duration.

We construct a fully-connected graph $\mathcal{G}$ with $N$ nodes that have edge distances obtained as a combination of feature-space distances and temporal proximity

$$E(m, n) = E_s(m, n) \cdot E_\tau^\alpha(m, n)\,, \tag{2.6}$$

where $\alpha$ acts as a further modulating factor. The graph $\mathcal{G}$ is converted to a 1-nearest-neighbor graph by keeping only one edge to the *nearest* node for each node based on the edge distances defined in $E$, resulting in the first clustering partition. TW-FINCH [42] operates recursively and merges clusters (nodes) by averaging their representations and timestamps until the desired number of clusters (connected components) is obtained. It employs two algorithms: Temporally Weighted Clustering Hierarchy (Algorithm 1) and Final Action Segmentation (Algorithm 2).

**Algorithm 1: Temporally Weighted Clustering Hierarchy:** This algorithm aims to create a hierarchy of partitions for a given video, with each partition containing clusters of clips from the previous partition. It works as follows:

1. It takes as input a video represented as a set of feature vectors, one for each clip.
2. Initializes timestamps for each clip and computes a temporally weighted 1-NN graph (as described in Equations 2.4 - 2.6). This graph links each clip to its closest neighbor in terms of appearance and temporal proximity.
3. The connected components of the graph form the first partition of clips into clusters.
4. The algorithm recursively merges clusters based on their average features and timestamps.
5. This merging process continues until only one cluster remains. The output is a set of hierarchical partitions.

**Algorithm 2: Final Action Segmentation:** This algorithm refines the output of Algorithm 1 to produce the desired number of action segments (K).

1. It takes as input the number of desired action segments (K), the video, and a specific partition from the hierarchy produced by Algorithm 1.
2. It iteratively merges clusters, prioritizing those with minimal temporal distance until the number of clusters equals K.
3. This algorithm ensures the temporal consistency of the final action segments by considering the average timestamps of the clusters during merges.

For more details, we request the reader to refer to Algorithm 1 and 2 in [42].

Note that the original algorithm [42] does not include an $\alpha$ scaling factor, or considers it to be 1 (*cf*. Eq. 2.6). However, we observed a few cases where this is unable to produce temporally consistent segments using our learned embeddings. As higher values of alpha amplify the strength of the temporal proximity factor, incrementing it progressively (*e.g.* by 0.1 steps) yields temporally consistent clusters.

## 2.3 Experiments.

We evaluate our proposed approach for lecture segmentation and present extensive ablation studies.

| | | Feature modality | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Method** | **visual** | **textual** | **learned** | **NMI ↑** | **MOF ↑** | **IOU ↑** | **F1 ↑** | **BS@30 ↑** |
| 1 | Naïve (Equal Splits) | - | - | - | 71.8 | 75.5 | 62.7 | 74.0 | 32.5 |
| 2 | Content-Aware Detector [91] | ✓ | - | - | 72.9 | 73.3 | 59.4 | 65.9 | 57.0 |
| 3 | Text Tiling [92] | - | ✓ | - | 67.9 | 64.7 | 46.3 | 50.9 | 33.7 |
| 4 | LDA [93] | - | ✓ | - | 70.0 | 72.4 | 57.6 | 68.2 | 38.8 |
| 5 | K-Means | - | - | ✓ | 63.9 | 66.8 | 48.2 | 55.7 | 44.9 |
| 6 | CTE [41] | - | - | ✓ | 67.2 | 67.3 | 48.1 | 57.3 | 41.5 |
| 7 | | ✓ | - | - | 71.6 | 71.3 | 56.5 | 66.4 | 46.9 |
| 8 | Vanilla TW-FINCH [42] | - | ✓ | - | 74.6 | 75.4 | 62.0 | 71.2 | 48.9 |
| 9 | | ✓ | ✓ | - | 74.9 | 75.1 | 61.7 | 70.9 | 52.1 |
| 10 | **Ours** | - | - | ✓ | **79.8** | **80.3** | **69.2** | **76.9** | **58.7** |

Table 2.1: Segmentation performance on all 350 lectures from 15 courses. Our approach outperforms all baselines. Here, *learned feature modality* refers to the features extracted from our joint text-video embedding model (Sec. 2.2.2). For rows 2-4, the *visual and textual feature modalities* refer to the unprocessed lecture video or transcripts respectively. For rows 7-9, *visual and textual feature modalities* refer to the features obtained from pre-trained backbones (ResNet or BERT, Sec. 2.2.1).

### 2.3.1 Experiment setup

**Training procedure** involves two stages. In the first stage, we pre-train the embedding model (Sec. 2.2.2) on the Courses without Segmentation (CwoS). In the second stage, we fine-tune our embedding model on the Courses with Segmentation (CwS) in an unsupervised manner. Note that we do not update the feature extraction backbones (BERT, ResNet, *etc*.). Next, we extract the visual and textual embeddings from the trained model, which are used to perform segmentation using the TW-FINCH algorithm. We evaluate the segments obtained from TW-FINCH using five different metrics described below.

**Evaluation dataset.** We evaluate all 15 courses of CwS to report performance. Our self-supervised fine-tuning process can be easily extended to a new course that needs segmentation. Further impact of pre-training and fine-tuning strategies is evaluated in Sec. 2.3.3, Ablation 2.

**Evaluation metrics.** Normalized Mutual Information (NMI) is a standard clustering metric [94]; Mean over Frames (MoF), F1-score, and Intersection over union (IoU) or the Jaccard index are standard metrics used in segmentation (*e.g.* [42]); and Boundary Score @ $k$ (BS@$k$), is the average number of predicted boundaries matching with the ground truth boundaries within a $k$ second interval. Different from the above metrics, BS@$k$ measures the localization of boundaries rather than the overlap of segments.

### 2.3.2 Comparison against Segmentation Baselines

We briefly describe the baselines below:

**1. Naïve.** The video lecture is split into equal parts based on the number of ground-truth (GT) segments.

**2. Content-Aware Detector** [91] is a shot/scene detection algorithm that detects jump cuts in a video by finding areas of high difference between two adjacent frames. While there is no direct way to set the number of segments, we search across several thresholds to generate the GT number of segments to ensure a fair comparison.

**3. Text Tiling** utilizes only the transcripts to predict the segments. We implement text tiling using the NLTK [92] library. As there is no way to set the number of clusters, we let the algorithm decide the appropriate number of clusters.

**4. Latent Dirichlet Allocation (LDA)** [95, 93] is a generative probabilistic model that automatically discovers hidden topics based on a text corpora. LDA is used as a baseline in identifying topic transitions in educational videos [24] and many other topic modeling works [96, 97]. We train the LDA model on the transcripts of AVLectures and represent each clip as a distribution over topics. Finally, we use TW-FINCH to perform lecture segmentation using these vectors.

**5. K-Means** clustering algorithm is applied to the learned embeddings from our joint text-video embedding model.

**6. CTE** [41] is a *strong unsupervised approach* that infuses features with relative temporal information and clusters them using K-Means. We report CTE scores using learned embeddings from our joint model.

**7. Vanilla TW-FINCH** [42]. Visual and textual features from the feature extraction pipeline in Sec. 2.2.1 are adopted here (no lecture-awareness). We apply the TW-FINCH segmentation algorithm directly on these features.

We compare all baselines against our approach and report performance in Table 2.1. For K-Means (row 5) and CTE (row 6), we report the best performance with learned features. We observe that the

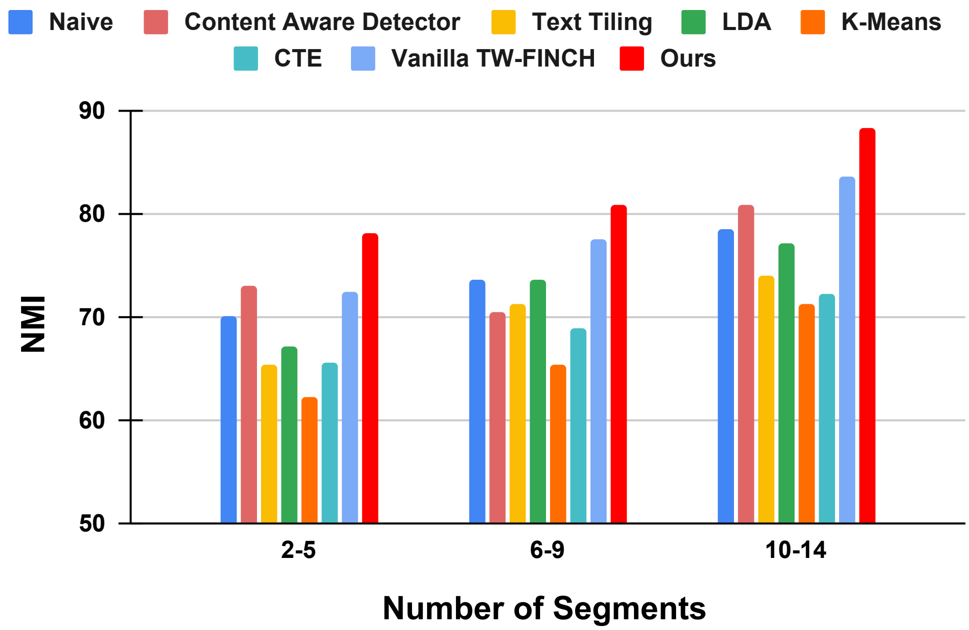


Figure 2.4: Comparing NMI across all methods grouped by the number of ground-truth segments.

Naïve baseline (row 1) performs quite well, and in fact outperforms strong baselines with learned features such as K-Means (row 5) and CTE (row 6). This may be due to an inherent bias of the instructor spending close to equal amounts of time on various sub-topics of the lecture. The text-only approach, Text Tiling (row 3) lags behind the visual-only approach Content-Aware Detector (row 2) as the latter performs specially well on non-blackboard courses (see Fig. 2.5). An additional factor is that we are unable to select the ground-truth number of clusters for Text Tiling. Our approach (row 10) outperforms all baselines. In fact, the gap between our approach and Vanilla TW-FINCH baselines (rows 7-9) highlights the importance of training lecture-aware representations using the joint text-video embedding model, as even a combination of both modalities (row 9) falls short of our approach by almost 5% on NMI. This emphasizes the importance of learning lecture-aware embeddings in a self-supervised manner.

We further analyze the results by slicing lectures based on the number of GT segments in Fig. 2.4. Our method outperforms all the other baselines irrespective of the number of segments in the ground truth, indicating the robustness of our approach. Another way is to slice the data based on presentation mode, specifically blackboard and non-blackboard. Fig. 2.5 shows a similar trend, our approach outperforms all baselines in both scenarios. Interestingly, the Naïve baseline works well for blackboard lectures (perhaps indicative of relatively equal time allocation across sub-topics), while slide-based lectures with clear transitions are segmented well by the visual Content-Aware Detector.

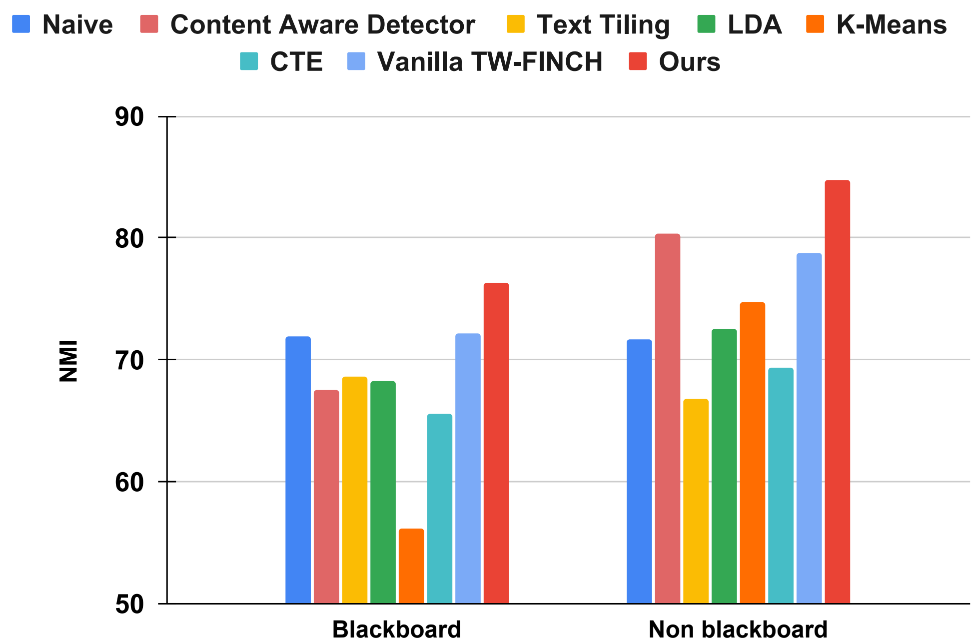



Figure 2.5: Comparing NMI across all methods grouped by presentation mode: blackboard and non-blackboard.

### 2.3.3 Ablation Studies

We present various ablation studies to understand the contributing factors to our approach's performance.

**1. How important is each visual feature?** To understand the impact of each individual visual feature, we train separate models on all combinations of visual features and report performance in Table 2.2. We observe that although the individual features perform reasonably well, OCR outperforms 2D and 3D representations, and it is the combination of all features that outperforms all other variations.

**2. Impact of training datasets.** Educational lecture videos are very different compared to instructional videos or movies. Lecture videos typically have much less dynamic visual content and compensate for this through substantial amounts of textual information, both accompanying (narrated speech/transcripts) and even inside the video (which we extract using OCR). As a result, the representations learned from instructional videos may not transfer well to the tasks in the education domain, necessitating a collection of lecture videos for learning representations.

We validate the above claim by showing that pre-training on AVLectures is more effective than pre-training on the general instructional videos (*e.g.* HowTo100M) for the lecture segmentation task, see Table 2.3. While using a model to improve representations is clearly better than the naïve baseline

| Features | | | Metrics | | | | |
|---|---|---|---|---|---|---|---|
| **2D** | **3D** | **OCR** | **NMI ↑** | **MOF ↑** | **IOU ↑** | **F1 ↑** | **BS@30 ↑** |
| ✓ | - | - | 76.6 | 76.8 | 64.4 | 73.0 | 54.4 |
| - | ✓ | - | 75.1 | 76.0 | 62.9 | 72.2 | 50.7 |
| - | - | ✓ | 78.9 | 79.7 | 68.2 | 76.2 | 57.7 |
| ✓ | ✓ | - | 76.6 | 77.0 | 64.7 | 73.5 | 53.9 |
| ✓ | - | ✓ | 79.5 | 80.3 | 69.1 | 76.9 | 58.6 |
| - | ✓ | ✓ | 78.4 | 79.5 | 68.3 | 76.4 | 57.9 |
| ✓ | ✓ | ✓ | **79.8** | **80.3** | **69.2** | **76.9** | **58.7** |

Table 2.2: Impact of visual features.

| | **PT** | **FT** | **NMI ↑** | **MOF ↑** | **IOU ↑** | **F1 ↑** | **BS@30 ↑** |
|---|---|---|---|---|---|---|---|
| 1 | HowTo100M | - | 73.0 | 58.8 | 68.3 | 73.0 | 48.5 |
| 2 | HowTo100M | CwS | 74.5 | 75.1 | 61.5 | 71.0 | 49.7 |
| 3 | - | CwS | 78.5 | 79.0 | 67.2 | 75.3 | 57.2 |
| 4 | CwoS | - | 77.7 | 78.0 | 66.0 | 74.2 | 57.1 |
| 5 | **CwoS** | **CwS** | **79.8** | **80.3** | **69.2** | **76.9** | **58.7** |

Table 2.3: Impact of pre-training (PT) on HowTo100M or CwoS. The second column indicates whether unsupervised fine-tuning (FT) is performed on CwS.

(NMI 73.0 *vs*. 71.8), we can see that a model pre-trained on AVLectures (rows 3-5) outperforms a model pre-trained on HowTo100M (rows 1-2) consistently. This strengthens our dataset contribution and highlights the importance of pre-training on AVLectures for tasks in the education domain. In row 4, though the model is trained only on CwoS, it is able to generalize well to unseen courses and predict reasonable segmentation boundaries. After fine-tuning the model on CwS we get a slight boost in performance (row 5). Row 5 outperforms row 3 that is trained only on CwS, justifying our adoption of pre-training on CwoS followed by fine-tuning on CwS. Note that all the training is performed in an unsupervised manner and only applies to the text-video embedding model.

**3. Impact of modalities.** From the joint text-video embedding model we can extract visual and textual embeddings. We compare visual-only, textual-only, and a concatenation of visual and textual learned embeddings in Table 2.4. A combination of both modalities shows best results.

| Embed. type | NMI ↑ | MOF ↑ | IOU ↑ | F1 ↑ | BS@30 ↑ |
|---|---|---|---|---|---|
| Visual | 78.6 | 79.1 | 67.7 | 75.7 | 57.9 |
| Textual | 75.6 | 77.0 | 64.4 | 73.5 | 50.3 |
| **Visual + Textual** | **79.8** | **80.3** | **69.2** | **76.9** | **58.7** |

Table 2.4: Impact of different embedding modalities.

| PT | FT | Duration | NMI ↑ | MOF ↑ | IOU ↑ | F1 ↑ | BS@30 ↑ |
|---|---|---|---|---|---|---|---|
| | | 4-8 | 53.2 | 58.7 | 53.0 | 40.9 | 26.4 |
| ✓ | - | **10-15** | **77.7** | **78.0** | **66.0** | **74.2** | **57.1** |
| | | 20-25 | 73.9 | 77.0 | 64.6 | 74.8 | 36.7 |
| | | 4-8 | 54.6 | 60.0 | 54.1 | 42.2 | 26.6 |
| ✓ | ✓ | **10-15** | **79.8** | **80.3** | **69.2** | **76.9** | **58.7** |
| | | 20-25 | 74.5 | 77.7 | 65.6 | 75.6 | 36.8 |

Table 2.5: Performance for different clip durations (in seconds). PT: Pre-training on CwoS, FT: Fine-tuning on CwS.

**4. Impact of lecture clip duration.** Works on instructional videos such as [27, 17] typically split videos into short clips of 4s. We perform an experiment to determine an appropriate clip duration for lecture videos: 4-8s, 10-15s, or 20-25s. The results reported in Table 2.5 coincide with our expectations that 4-8s clips are too short to capture meaningful information while 20-25s clips are harder to represent due to the pooling operation and also cause a significant drop in BS@30 due to their longer duration. Clips of 10-15s are a good compromise and span meaningful lecture content while not losing information to pooling.

**5. What if the number of segments is unknown?** It is not trivial to guess the ideal number of segments for the unseen lectures. In such cases, we let the TW-FINCH algorithm decide the appropriate number of clusters. TW-FINCH produces a hierarchy of partitions where the number of clusters reduces with successive partitions. We use the $2^{nd}$- and the $3^{rd}$-last partitions to estimate the number of segments automatically and report performance in Table 2.6. We also report scores for the Naïve baseline on the above partitions as well.

In addition to the usual metrics we also compute the L1 distance between the ground-truth number of clusters and the number of automatically estimated clusters for both the partitions. The L1 distance between the last and $2^{nd}$-last partition is 8.554 and that of $3^{rd}$-last is 4.614. The $3^{rd}$-last partition has a

| Method | Partition | NMI ↑ | MOF ↑ | IOU ↑ | F1 ↑ | BS@30 ↑ |
|---|---|---|---|---|---|---|
| Ours | $2^{\text{nd}}$ last | 63.7 | 61.7 | 59.5 | 42.6 | 42.3 |
| | $3^{\text{rd}}$ last | 72.1 | 59.7 | 39.1 | 42.7 | 65.2 |
| | GT | 79.8 | 80.3 | 69.2 | 76.9 | 58.7 |
| Naive | $2^{\text{nd}}$ last | 58.6 | 58.9 | 54.1 | 40.5 | 27.0 |
| | $3^{\text{rd}}$ last | 66.9 | 51.2 | 33.8 | 39.3 | 38.9 |
| | GT | 71.8 | 75.5 | 62.7 | 74.0 | 32.5 |

Table 2.6: Allowing TW-FINCH to estimate the number of clusters.

lower L1 score compared to the $2^{\text{nd}}$-last partition. This, along with the other metrics, indicates that the number of clusters generated by the $3^{\text{rd}}$-last partition is closer to the ground-truth.

| Language Model | NMI ↑ | MOF ↑ | IOU ↑ | F1 ↑ | BS@30 ↑ |
|---|---|---|---|---|---|
| Word2Vec | 78.9 | 79.7 | 68.4 | 76.4 | 58.2 |
| mpnet-v1 | 79.1 | 79.7 | 68.3 | 76.2 | 58.4 |
| **mpnet-v2** | **79.8** | **80.3** | **69.2** | **76.9** | **58.7** |

Table 2.7: Impact of different Language Models.

**6. Using different language embedding models.** In this study, we experiment with three different text embeddings,

1. word2vec: We first preprocess the transcripts by removing the most common stop words. Next, we extract the word embeddings from the GoogleNews pre-trained word2vec model [98]. word2vec encodes each word into to a 300-dimensional vector.
2. `multi-qa-mpnet-base-dot-v1` (mpnet-v1 in Table 2.7): This is a sentence transformer BERT model that uses the pre-trained MPNet [82] model and is trained on 215M (question, answer) pairs from diverse sources. This model encodes the transcripts into a 768-dimensional vector.
3. `all-mpnet-base-v2` (mpnet-v2 in Table 2.7): This model uses the pre-trained MPNet [82] model and is fine-tuned on a 1B sentence pairs dataset using a contrastive learning objective: given a sentence from the sentence pairs, the model should predict which sentence from a randomly sampled

other sentences was paired with it. This is the same model that was described in the Main paper Sec. 2.2.1.

The results of all three models are reported in Table 2.7. Although, the `all-mpnet-base-v2` model performs slightly better when compared to the other two text embedding models the scores are almost similar in all three variations. The results show that there is no significant impact on the type of text embeddings that are used to train the model.

| **Embed. dim.** | **NMI ↑** | **MOF ↑** | **IOU ↑** | **F1 ↑** | **BS@30 ↑** |
|---|---|---|---|---|---|
| 512 | 79.3 | 79.7 | 68.3 | 76.1 | **59.7** |
| 1024 | 79.3 | 80.3 | 68.9 | 76.7 | 59.0 |
| 2048 | **79.8** | **80.4** | **69.4** | **77.1** | 59.6 |
| 4096 | **79.8** | 80.3 | 69.2 | 76.9 | 58.7 |

Table 2.8: Impact of different embedding dimension.

**7. How does the model's embedding dimension affect the performance of segmentation?** We train the model with four different output embedding dimensions: 512, 1024, 2048 and 4096. It can be seen from Table 2.8 that the learned features are robust and independent of the feature dimension and therefore has little impact on the overall performance of the model on the segmentation task. Although the embedding dimensions 2048 and 4096 perform slightly better than the rest.

| | Feature modality | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | visual | textual | learned | NMI ↑ | MOF ↑ | IOU ↑ | F1 ↑ | BS@30 ↑ |
| 1 | ✓ | - | ✗ | 53.1 | 58.6 | 38.2 | 46.2 | 37.5 |
| 2 | - | ✓ | ✗ | 48.5 | 55.1 | 33.5 | 41.0 | 34.3 |
| 3 | ✓ | ✓ | ✗ | 53.1 | 58.9 | 38.6 | 46.5 | 37.9 |
| 4 | ✓ | - | ✓ | **63.9** | **66.8** | **48.2** | **55.7** | **44.9** |
| 5 | - | ✓ | ✓ | 49.2 | 56.4 | 35.0 | 42.4 | 33.7 |
| 6 | ✓ | ✓ | ✓ | 60.2 | 64.9 | 46.0 | 53.3 | 44.1 |

Table 2.9: Impact of different feature modalities on K-Means

| | Feature modality | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | visual | textual | learned | NMI ↑ | MOF ↑ | IOU ↑ | F1 ↑ | BS@30 ↑ |
| 1 | ✓ | - | ✗ | 65.0 | 65.4 | 45.9 | 55.4 | 38.6 |
| 2 | - | ✓ | ✗ | **67.2** | **68.1** | **49.6** | **59.4** | 35.3 |
| 3 | ✓ | ✓ | ✗ | 66.3 | 66.5 | 47.4 | 57.0 | 39.8 |
| 4 | ✓ | - | ✓ | 67.1 | 67.2 | 48.2 | 57.6 | 41.0 |
| 5 | - | ✓ | ✓ | 64.7 | 65.7 | 45.4 | 54.8 | 35.6 |
| 6 | ✓ | ✓ | ✓ | **67.2** | 67.3 | 48.1 | 57.3 | **41.5** |

Table 2.10: Impact of different feature modalities on CTE

**8. Impact of different feature modalities on K-Means and CTE [41]** We show the segmentation results for K-means and Continuous Temporal Embedding [41] (CTE) on the features extracted using the pipeline (Sec. 2.2.1 Main Paper) as well as on the learned embeddings from our joint text-video model. The scores are shown in Table 2.9 and 2.10. For K-Means, the learned visual embeddings (row 4) and the combination of learned visual and textual embeddings (row 6) outperforms all other variations by a good margin. The results highlight the importance of training lecture-aware representations using our joint text-video embedding model. For CTE, even though all the scores are relatively closer to each other, the one that uses text features (BERT embeddings) (row 2) and a combination of learned visual and textual embeddings (row 6) perform the best. Note that using a combination of learned visual and textual embeddings results in the highest boundary score, highlighting the importance of our learned representations in predicting better boundaries.

**9. Deeper analysis on Naïve method performing well.** As discussed in the paper, one reason why the Naïve method is effective is due to an inherent bias of the instructor spending almost equal amounts of time on different topics in certain lectures. For example, consider a lecture on Multivariate Calculus[8]. Here each of the segment is approximately 16 minutes, thus giving an upper-hand to the naive method. Upon further analysis, we observe that 73 of 350 lectures (nearly 20 % of CwS) have GT segment boundaries within 3 minutes to the boundaries suggested by the Naïve baseline. We perform an ablation study by varying the number of splits obtained by automatically clustering lectures with TW-FINCH. The results indicate that splitting lectures at the ground truth number of segments gives a better segmentation performance than splitting it in any other way, as seen in Table 2.6.

[8]Multivariate Calculus - segment-1, segment-2, segment-3

| Method | NMI ↑ | MOF ↑ | IOU ↑ | F1 ↑ | BS@30 ↑ |
|---|---|---|---|---|---|
| NCE | 70.6 | 71.5 | 56.3 | 66.3 | 43.2 |
| Ours | **79.8** | **80.3** | **69.2** | **76.9** | **58.7** |

Table 2.11: Segmentation performance when lecture-transcript alignment is done using Noise Contrastive Estimation (NCE) loss.

**10. Boundary scores at various intervals.** We also perform an ablation study by computing Boundary Scores at various values of K, and it's plot is shown in Fig. 2.6. Typically, the instructor spends at least 25-30 seconds (in answering student's questions, erasing the blackboard etc.) before switching to new a topic. This was the reason behind reporting the scores for BS@30 in the paper. As expected, all methods perform worse for lower values of K and as K approaches 15, the use of 10-15s clip sizes hurts performance.

**11. Impact of lecture-transcript alignment strategies.** We also compare our approach with a more popular approach that uses Noise Contrastive Estimation (NCE) loss for aligning video-text pairs [27]. The results are reported in Table 2.11. Our approach, which uses max-margin ranking loss outperforms the NCE loss perhaps due to the scale of the dataset and the limited number of negative samples in the batch. We were unable to train with larger batch sizes due to GPU memory restrictions.

## 2.4 Qualitative results

We visualize segmentation outputs for three video lectures from different courses in Fig. 2.7 and compare our method with all other baselines. It is clear that our method yields better segments (overlap) and boundaries as opposed to other methods that produce noisy segments. In the third lecture, the first and second predicted segments of our approach are different from the GT while the other boundaries are detected correctly.

An additional problem that can be addressed using the embeddings learned from our joint text-video model is the text-to-video retrieval task. Given a text query, we retrieve a list of lecture clips for which the similarity scores with the text query are the highest. Fig. 2.8 shows some of the retrieved clips for various text queries. We can see that our model is able to relate the visual notion of graphs with the word. Similar results are observed for the other queries.

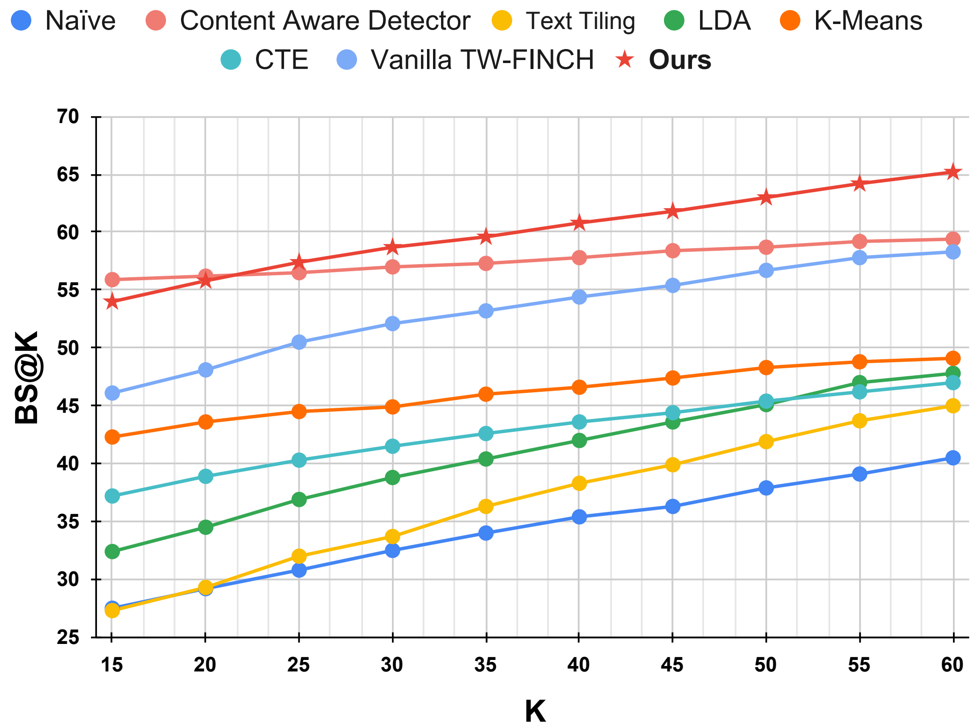



Figure 2.6: Boundary scores at different values of K.

Fig. 2.9 shows some of the retrieved clips for different text queries like *graphs coloring*, *operating systems*, etc. We also tested a query *erasing board* to check the model's comprehension of non-conceptual keywords, as shown in the last example of the figure. Although this query is not present in the transcript, it still correctly retrieves the clips in which the professor erases the blackboard. This demonstrates the importance of pre-training on the CwoS dataset.

Fig. 2.10 shows more qualitative results from the lecture segmentation task for lectures from different courses. Regardless of the number of segments, our method yields better segmentation length and boundaries when compared with the other baselines.

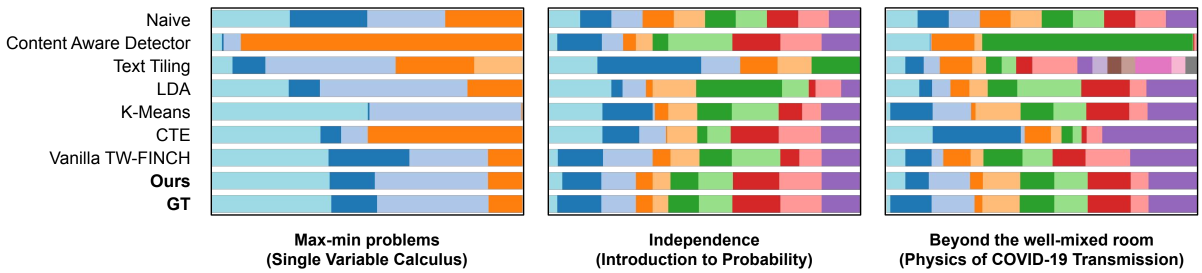



Figure 2.7: Segmentation examples for three lectures. Our approach closely resembles the ground-truth. Best viewed in color.

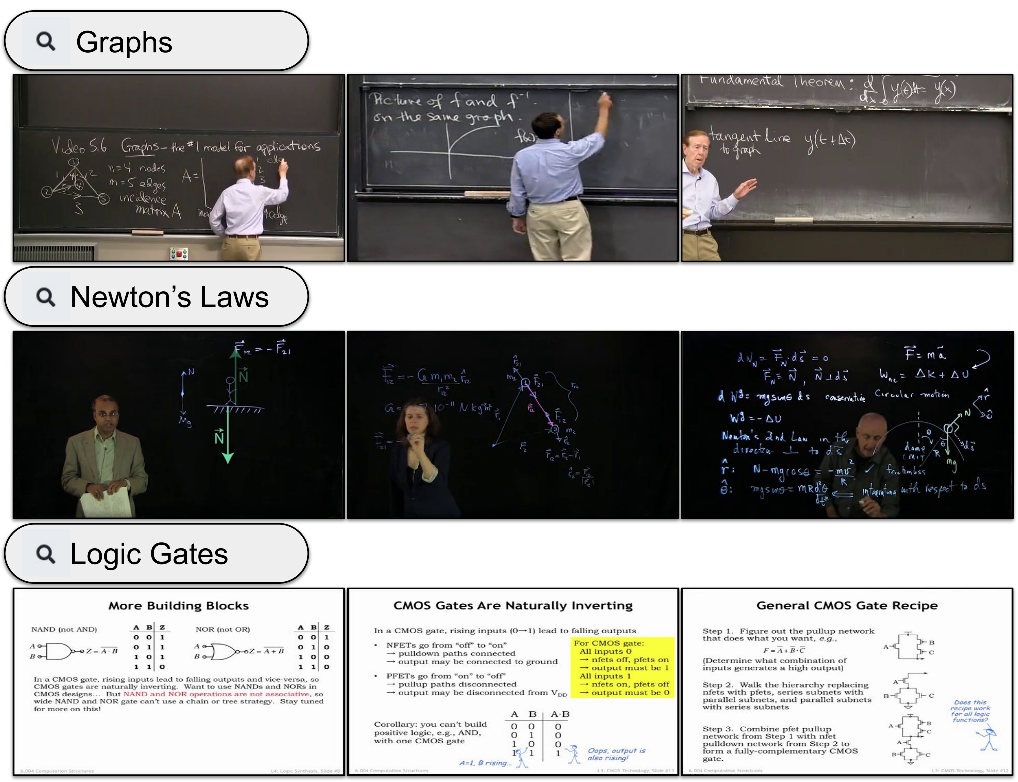



Figure 2.8: Examples of text-to-video retrieval for different queries using our learned joint embeddings. Our model is able to retrieve relevant lecture clips based on the query.

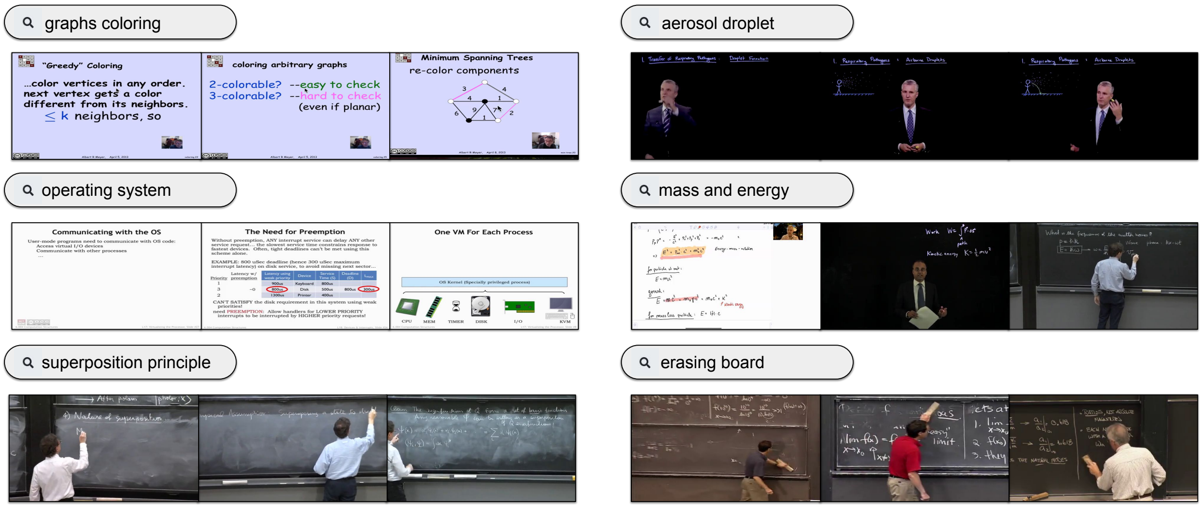



Figure 2.9: Text-to-video retrieval results on six queries. The figure shows the thumbnails of the top 3 retrieved lecture clips from our model. Our model is able to retrieve relevant lecture clips according to the query.

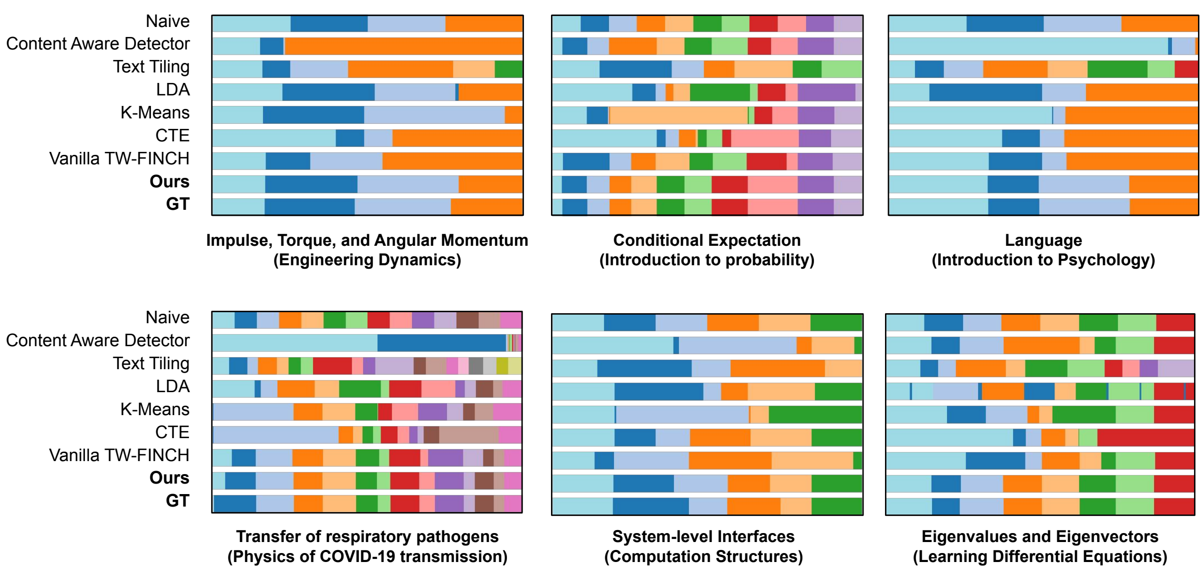



Figure 2.10: Segmentation examples for six lectures from different courses with varying a number of segments.

# Chapter 3

# Seeing Beyond Captions: Efficient CLIP Video Adaptation via Structured Semantic Role Labels

Large-scale vision language pretraining has proved effective in learning generalized and transferable visual representations, with powerful zero-shot capabilities [1, 99, 100, 101]. Among them, Contrastive Language-Image Pretraining (CLIP) [1] is a popular approach for learning rich semantic representations, thanks to the image-text alignment. As the representations generalize well, CLIP sees wide adoption in several downstream tasks ranging from simple classification to being used in state-of-the-art Vision-Language Models (VLMs) [102, 103, 104, 105]. This has spurred a lot of interest in adapting CLIP for video understanding [2, 106, 47, 107]. While these methods show promising results, they often come at the expense of large scale post-pretraining on hundreds of millions of videos.

We argue that as CLIP is pretrained on 400+ million image-language pairs, it already has the world knowledge required for general visual perception, and post-pretraining on hundreds of millions of videos can be wasteful. We hypothesize that the need for such large-scale post-pretraining is due to the contradiction between rich videos and sparse descriptions. Videos are a highly complex modality with multidimensional information about people, objects, states, actions, relations, that change over time. But the textual description representing the video is generally very sparse. Narrations [3] or captions [55] often fail to capture such details and therefore the capacity for visual processing is underutilized. The adaptation process only extracts sparse information from each sample and hence, requires millions of samples.

In this work, we revisit the large-scale post-pretaining paradigm for CLIP video adaptation and propose to use small-scale, but dense captions for efficient and holistic representation learning. Specifically, we use the VidSitu dataset [5] that is annotated using Semantic Role Labels (SRLs) that capture

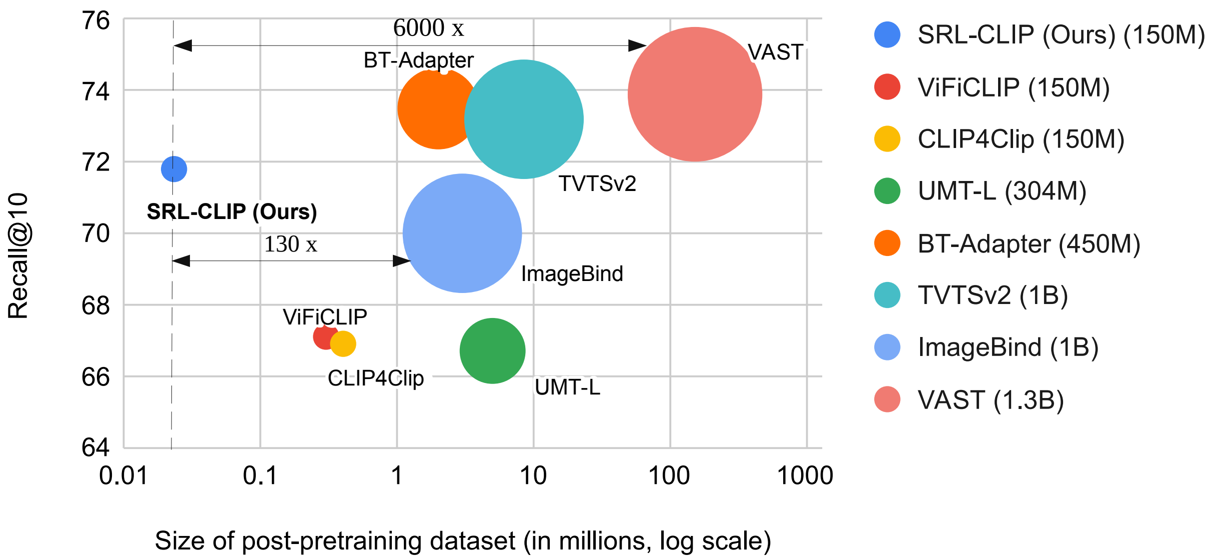



Figure 3.1: Zero-shot text-to-video retrieval performance on the MSR-VTT dataset. We compare SRL-CLIP (Ours) against various CLIP-based approaches that use orders of magnitude more post-pretraining data samples and/or have significantly larger models.

the holistic situation. In VidSitu, every 10 second video clip is divided into 2 second *events*. Each event contains structured annotations answering who (subject) is doing what (action), to/with whom (patient), where (scene), how (manner, adverbs), and why (purpose, goal). We believe that such details in a video description represent the visual concepts well and can provide a strong learning signal.

We propose to adapt CLIP through alignment between video-text SRLs, highlighting the power of rule-based high quality prompts generated from SRL. Our approach has multiple advantages: (i) The structured nature of SRL-based prompts (who, what, where, *etc*.) facilitate learning holistic visual concepts consistently across all the videos. (ii) VidSitu videos have many shot changes, while the annotated captions may represent visual concepts across all the events. This presents a challenging scenario with a strong signal to learn object permanence and temporal consistency, by meaningfully combining information across frames and events. (iii) Multiple events from the same videos show small differences creating *natural hard negatives* for learning. In addition, due to the structured nature of SRL, we can add/swap/replace verbs, nouns, or roles in an SRL-based prompt, easily creating *hard negative text prompts* that further boost the model's performance. We show that SRL-CLIP, adapted on a small dataset of 23k videos with dense annotations, learns powerful general purpose representations, improving over base CLIP on multiple video-language tasks. In particular, we achieve these improvements through *efficient adaptation* across multiple aspects – model size (150M params), post-pretraining dataset size (23k video clips), compute (one 12GB RTX 2080 GPU), and training time (5 hours)!

**Contributions.** In summary, we: (i) Favor efficient post-pretraining using small-scale but dense prompts, as compared to the current inefficient trend of using large-scale datasets. (ii) Propose to use SRL to create rule-based dense captions that capture holistic concepts in a video, through details such as actions, persons, objects, attributes, relations, manner, location, reasons, *etc*. (iii) Show that post-pretraining CLIP on mere 23k videos with dense SRL captions is efficient and on zero-shot text-to-video retrieval (see Fig. 3.1), it also performs better or on-par with state-of-the-art CLIP based video models that have $4{-}8\times$ more parameters and are trained on upto $4000\times$ more data. (iv) Consistently outperform the original CLIP model on various tasks: video situation recognition, dense video captioning and localization, and text-to-video retrieval, demonstrating that the representations learned with dense holistic captions generalize well across multiple tasks requiring different levels of perceptual granularity.

We emphasize that our innovation lies in an efficient and effective recipe to adapt CLIP for video data rather than architectural or modeling modifications.

## 3.1 Adaptation: From CLIP to SRL-CLIP

We present our approach to post-pretrain the CLIP model on VidSitu, a densely annotated video dataset. We start with background information related to CLIP (Section 3.1.1), followed by our adaptation strategy with specific emphasis on the architecture modifications for training and SRL prompts (Section 3.1.2). We end this section with a discussion on how SRL prompts support the creation of difficult negatives (Section 3.1.3) and some details about the adaptation process (Section 3.1.5).

### 3.1.1 Preliminaries

Consider a batch $B$ of paired image-text data: $\{(f_i, t_i)\}_{i=1}^{B}$, where $f_i$ is the image and $t_i$ describes $f_i$. The CLIP model [1] consists of an image encoder $\mathbf{f}_i = \Phi_I(f_i)$ and a text encoder $\mathbf{t}_i = \Phi_T(t_i)$ that are trained in a contrastive manner by applying the InfoNCE loss [108]:

$$L(\mathbf{f}_i, \mathbf{t}_i) = -\log \frac{\exp(\mathbf{f}_i^T \mathbf{t}_i)}{\sum_{j=1}^{B} \exp(\mathbf{f}_i^T \mathbf{t}_j)}\,, \tag{3.1}$$

and the corresponding symmetric version $L(\mathbf{t}_i, \mathbf{f}_i)$. The loss for the entire batch is $\mathcal{L} = \sum_{i=1}^{B}(L(\mathbf{f}_i, \mathbf{t}_i) + L(\mathbf{t}_i, \mathbf{f}_i))$.

Previous works have adapted CLIP using video-text pairs by mean pooling across multiple video frames [47]. Instead of an image $f_i$, consider a video-text pair $(V_i, t_i)$ where $V_i = \{f_{ij}\}_{j=1}^{L_i}$ has $L_i$ frames. We can adapt CLIP by computing a video representation $\mathbf{v}_i = \text{mean}_j(\mathbf{f}_{ij})$ and using the same loss $L(\mathbf{v}_i, \mathbf{t}_i)$.

**Video contextualizer (VC).** Instead of mean pooling, a VC may be used for learning video-text representations [45, 2]. To contextualize all frames, a simple Transformer encoder (Tx) [60] ingests frame representations as tokens. A learnable CLS token, $\mathbf{h}_{\mathsf{CLS}}$, is inserted before all frames. The input to the VC is: $[\mathbf{h}_{\mathsf{CLS}}, \mathbf{f}_{i1}, \dots, \mathbf{f}_{iL_i}]$. Position encoding [60] is added to the video frame tokens to specify their temporal order. Finally, the output at the CLS token is considered as the video representation, *i.e.*, $\mathbf{v}_i = \hat{\mathbf{h}}_{\mathsf{CLS}}$. Note, the VC can be trained jointly with adaptation of the backbone through the same loss $L(\mathbf{v}_i, \mathbf{t}_i)$.

In the next part, we show how VC can be adopted for VidSitu annotations.

### 3.1.2 Adaptation with SRLs

The VidSitu dataset [5] features videos that are split into $P$=5 contiguous short *events*, *i.e.* $V_i = [E_{ik}]_{k=1}^{P}$. Each event contains a detailed annotation: an action label and corresponding semantic role labels (SRL) with the role-noun pairs. For example, in Fig. 3.3, we show the action *drive*, with roles *driver*, *vehicle*, *manner* (of driving), and *scene*, each described through a short caption (noun).

We use such fine-grained labels to create a prompt for each event, $t_{ik}$, leading to event and text pairs $(E_{ik}, t_{ik})$. Fig. 3.2 illustrates the overall adaptation strategy.

**Video contextualizer (VC) for encoding VidSitu events.** We modify the VC to account for VidSitu's structured annotations. During training, $B$ videos $\{V_i\}_{i=1}^{B}$ are fed to the model at once. Each video is split into $P$ events. From each event, we sub-sample $T$ frames, *i.e.*, for each video, we have $L$=$P \cdot T$ frames. The VC operates over a sequence of all frames, passed through the CLIP image encoder. Let $f_{ik}^{j}$ be the $j^{\text{th}}$ frame for event $E_{ik}$ of video $V_i$.

Similar to CLS in BERT [109], we create two types of learnable tokens that collect information about the video. $\mathbf{v}_i$ represents the overall video $V_i$ and $\mathbf{e}_{ik}$ represents event $E_{ik}$. Note, these embeddings are shared across all videos. To indicate the type of token, we augment visual/learnable encodings with type embeddings $\mathbf{e}_v^{\text{typ}}$ for video, $\mathbf{e}_e^{\text{typ}}$ for event, and $\mathbf{e}_f^{\text{typ}}$ for the frame. Furthermore, we encode position with two embeddings, $\mathbf{e}^{\text{e-pos}}$ for event position and $\mathbf{e}^{\text{f-pos}}$ for frame position within the event. Overall, our

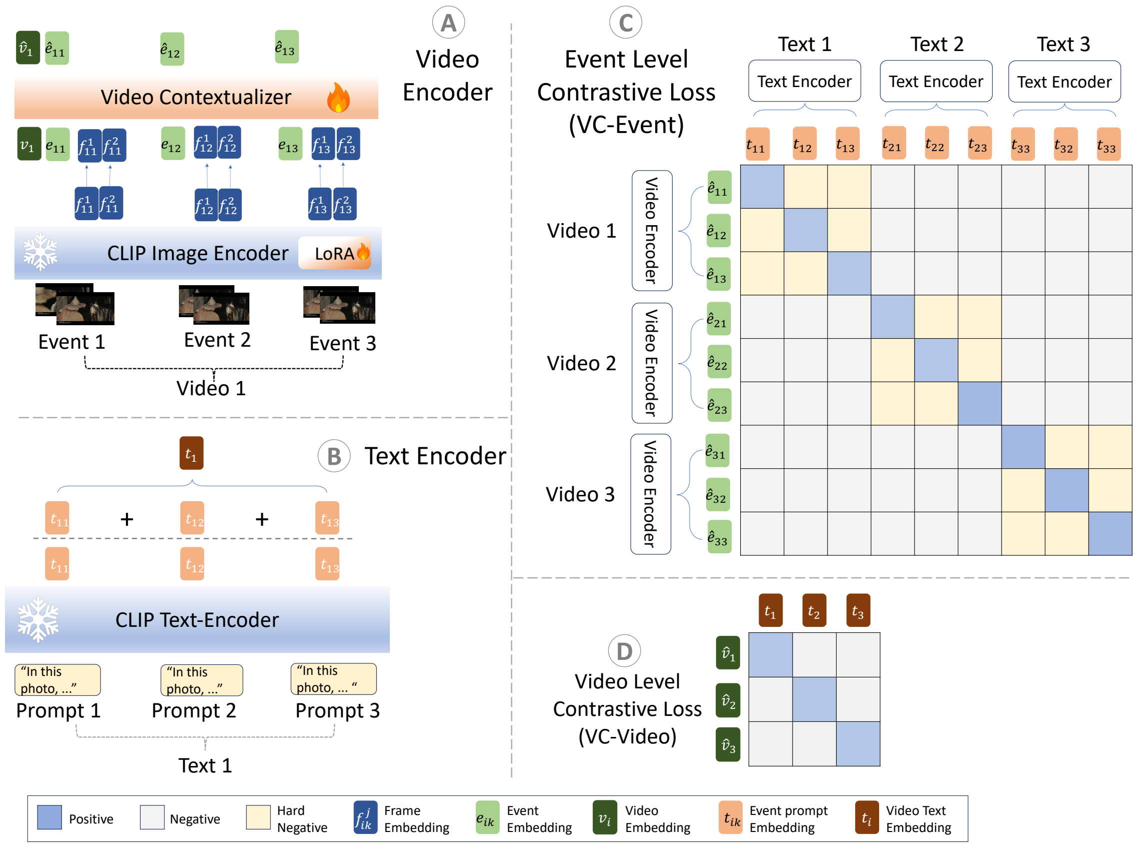



Figure 3.2: Overview of our CLIP adaptation strategy. **Top-left** (A) shows the visual encoder, consisting of the CLIP backbone and the video contextualizer (VC), applied to a single video with $P{=}3$ events for illustration. **Bottom-left** (B) shows the frozen CLIP text encoder extracting event-level representations. **Top-right** (C) shows the event-level contrastive loss, VC-Event, with natural hard negatives due to multiple events within a single video. **Bottom-right** (D) shows the video-level constrastive loss, VC-Video.

input tokens are:

$$\mathbf{v}_i = \mathbf{v}_i + \mathbf{e}_v^{\text{typ}}\,, \tag{3.2}$$

$$\mathbf{e}_{ik} = \mathbf{e}_{ik} + \mathbf{e}_e^{\text{typ}} + \mathbf{e}_k^{\text{e-pos}}\,, \tag{3.3}$$

$$\mathbf{f}_{ik}^j = \mathbf{f}_{ik}^j + \mathbf{e}_f^{\text{typ}} + \mathbf{e}_k^{\text{e-pos}} + \mathbf{e}_j^{\text{f-pos}}\,, \tag{3.4}$$

and passed to the VC, $\Phi_V$, after LayerNorm [110]:

$$\Phi_V([\mathbf{v}_i, \mathbf{e}_{i1}, \mathbf{f}_{i1}^1, \ldots, \mathbf{f}_{i1}^T, \ldots, \mathbf{e}_{iP}, \mathbf{f}_{iP}^1, \ldots, \mathbf{f}_{iP}^T])\,. \tag{3.5}$$

We denote outputs after the VC as $\hat{\mathbf{v}}_i, \hat{\mathbf{e}}_{ik}, \hat{\mathbf{f}}_{ik}^j$ for video, event, and frame tokens respectively. Fig. 3.2 (top-left) illustrates this process.

**Creating event-level prompts.** Our prompt $t_{ik}$ is a simple template that enumerates over the action and semantic role labels for each event. An example is shown below. Template words in gray, label type in *italics*, the label in line:

> In this photo, the *action* is walk where, the *walker* is man with short hair wearing collared shirt, *direction* is forward, *manner* is slowly, and *scene* of the event is apartment.

We also consider generating natural language prompts using a language model (LLaMa [53]). Labels are underlined:

> In this photo, a man with short hair wearing a collared shirt is walking slowly in an apartment.

However, these show worse performance. Telling the model that walking is the *action*, or the role played by a person with collared shirt is *walker*, and the apartment is a *scene* allows for holistic and dense representation learning.

**Losses.** We train our model with losses at multiple levels. For this part, we will recall some notations: the prompt encoding $\mathbf{t}_{ik} = \Phi_T(t_{ik})$; $\mathbf{f}_{ik}^j$ is the frame encoding before VC; $\hat{\mathbf{e}}_{ik}$ is the event encoding after VC; and $\hat{\mathbf{v}}_i$ the video encoding after VC. We also consider a prompt representation for the full video obtained by mean pooling over all event-level prompts, $\mathbf{t}_i = \text{mean}_k(\mathbf{t}_{ik})$.

All losses below are of the contrastive loss type shown in Eq. (3.1):

(i) **CLIP-Event** applies a loss on the event representation obtained by mean pooling raw CLIP frame encodings: $L_{\text{event}}^{\text{CLIP}} = L(\text{mean}_j(\mathbf{f}_{ik}^j), \mathbf{t}_{ik})$.

(ii) **CLIP-Video** applies a loss on the video representation obtained by mean pooling raw CLIP frame encodings across the video and contrasting against the video-level prompt: $L_{\text{video}}^{\text{CLIP}} = L(\text{mean}_{jk}(\mathbf{f}_{ik}^j), \mathbf{t}_i)$.

(iii) **VC-Event** applies a loss on the event representation post VC and the event-level prompt: $L_{\text{event}}^{\text{VC}} = L(\hat{\mathbf{e}}_{ik}, \mathbf{t}_{ik})$. Fig. 3.2 (top-right) represents this loss. Note how multiple events from the same video are used as negatives.

(iv) **VC-Video**: applies a loss on the video representation post VC and video-level prompt: $L_{\text{video}}^{\text{VC}} = L(\hat{\mathbf{v}}_i, \mathbf{t}_i)$. Fig. 3.2 (bottom-right) represents this loss.

We also use the symmetric version of the losses, *e.g.* $L(\mathbf{t}_{ik}, \hat{\mathbf{e}}_{ik})$, which are not shown here for brevity. We train the VC and adapt the backbone through a combination of all losses: $\mathcal{L} = L_{\text{event}}^{\text{CLIP}} + L_{\text{event}}^{\text{VC}} + \lambda(L_{\text{video}}^{\text{CLIP}} + L_{\text{video}}^{\text{VC}})$.

### 3.1.3 SRL Contains and Facilitates Hard Negatives

Contrastive learning requires hard negatives (HN) during training to prevent the model from finding the easy difference between the image and negative prompt.

**Natural hard negatives.** Training with dense template prompts along with our batch creation strategy gives us natural hard negatives which promotes fine-grained, holistic, and efficient representation learning. For the CLIP-Event and VC-Event losses, negative prompts are obtained from different events of the same video which are generally quite similar (perhaps differing in only some of the verb/roles/nouns). Fig. 3.2 top-right shows these natural HNs in light yellow background. For example, a template prompt from a different event of the same video as described above (walking example) looks like this:

> In this photo, the *action* is sit where, the *thing sitting* is man with short hair wearing collared shirt, *manner* is casually, and *scene* of the event is apartment.

These subtle differences in verbs, roles or their corresponding nouns presents a challenging learning scenario for the model.

**Artificial hard negatives by replacing verb-role pairs.** Although natural HNs provide challenging samples to the model, template-based prompts can be used easily to create artificial HNs by replacing verb-role pairs. Starting from the positive prompt, we replace the verb with a randomly sampled verb from the batch. We also replace its corresponding roles, but keep the nouns unchanged. Note, common roles such as *direction*, *manner*, *scene* remain unchanged; making the prompts quite hard. This allows the model to focus more on the *action*. We use $\mathcal{N}_{vr}$ such negatives. Differences to our walking example are in red:

> In this photo, the *action* is jog where, the *jogger* is man with short hair wearing collared shirt, *direction* is forward, and *scene* of the event is apartment.

**Incorporating negatives in the loss.** HNs are only added to the event-level losses (VC-Event and CLIP-Event). The loss function in Eq. (3.1) is extended by including similarity scores between the visual information and the negative prompts in the denominator.

### 3.1.4 Going beyond VidSitu: Creating synthetic SRL data using Kinetics-700.

We advocate the use of dense text prompts for efficient post-pretraining of CLIP for videos. Thus, while VidSitu is a natural fit, our work is *not* specific to VidSitu. While sparse labels generally perform worse (*e.g.* ViFi-CLIP), we present an approach to leverage action labels in Kinetics-700 and create dense SRL prompts that are effective for adaptation. Given video frames, we prompt LLaVA-1.6 (`llava-hf/llava-v1.6-mistral-7b-hf`) to generate role labels through questions: *e.g.* for the action label *driving*, we ask "who is driving?" or "what are they driving?". Thus, we create *K-SRL*: 40k Kinetics videos with sparse action labels, augmented with dense and automatic (albeit noisy) SRL annotations. Subsequently, we fine-tune our model with this data without VC.

### 3.1.5 Implementation Details

We use the OpenAI CLIP implementation and its associated checkpoints [1], restricting experiments to `ViT-B/32` and `ViT-B/16` models. We add LoRA adapters to the CLIP image encoder and freeze both the CLIP image and text encoders. We find $r$=64 rank to work well in our experiments. Our VC module consists of 6 Tx encoder layers. We use $\lambda$=0.25 for combining video- and event-level losses. The number of artificial hard negatives $\mathcal{N}_{vr}$=4. We use a learning rate of $10^{-6}$ and the AdamW [111] optimizer. Each video in VidSitu has $P$=5 events, and we sub-sample $T$=4 frames from each event, for a total of $L$=20 frames for a 10 s video. We post-pretrain for 40 epochs on *one 12GB RTX2080 GPU* with a batch size of $B$=20 videos (100 event-text pairs).

## 3.2 Experiments

We evaluate SRL-CLIP on a variety of video understanding tasks that require different levels of perceptual granularity. This is followed by thorough ablations on the VidSitu dataset, providing insights into the adapation parameters, loss functions, prompt creation strategies, and benefits of hard negatives.

### 3.2.1 Zero-shot Text-to-Video (T2V) Retrieval

We present results on zero-shot T2V retrieval. To evaluate SRL-CLIP's representations for coarse video level understanding, the VC is ignored for this part and retrieval scoring is performed by extracting frame-level features from SRL-CLIP's backbone, followed by simple mean pooling, similar to [47].

We evaluate on two popular datasets: MSRVTT [124] and LSMDC [125]. Results obtained using standard retrieval metrics (recall and mean/median rank) are presented in Table 3.1. We also report post-pretraining efficiency by showing the dataset size and model size. We categorize and describe the competing methods in three sections:

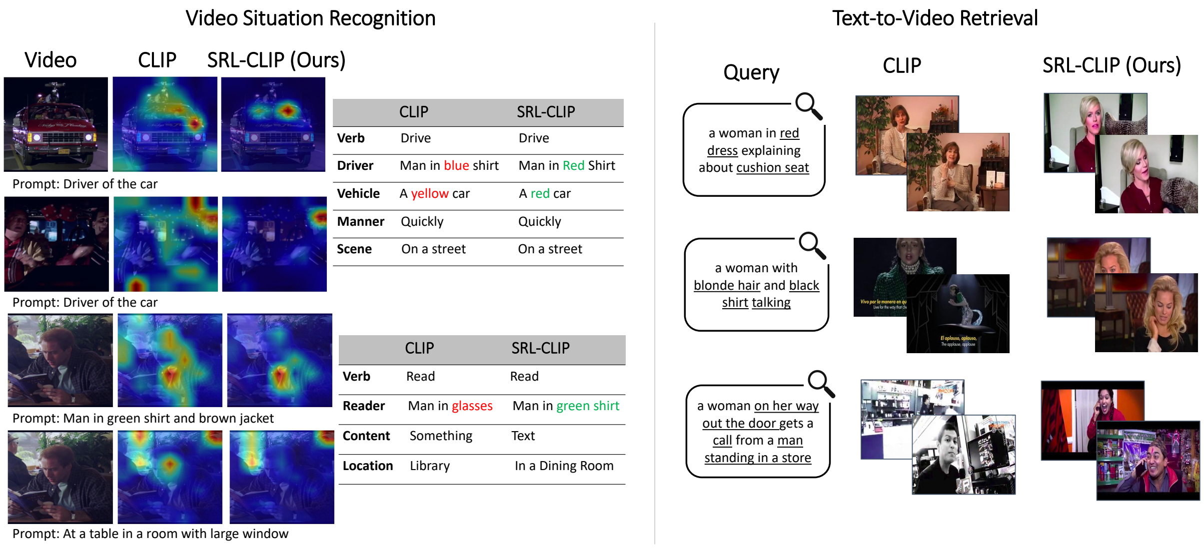



Figure 3.3: Qualitative results comparing CLIP and SRL-CLIP. **VidSitu** (left) shows improved attention maps resulting in better noun (SRL) captions; and **T2V Retrieval** (right) shows that SRL-CLIP has better awareness to details.

**A. Non-CLIP based models** generally require large models and large datasets to perform on par with CLIP based models. SoTA methods like VideoCoca [117] and Florence [119] are trained on 100M and 900M video samples and adopt ViT-H as a backbone, resulting in 2.1B and 637M parameters respectively. They require pretraining from scratch on multiple A100 GPUs for several days.

On the other hand we build on top of CLIP-base of 150M parameters, pretrained on 400M *images* (an order of magnitude smaller when compared to the number of *video frames* used above). Our post-pretraining (LoRA adaptation) is on 23k video clips from VidSitu and can be performed on a single 12GB RTX 2080 GPU within 5 hours. On MSRVTT, w outperform VideoCoCa by 4.8% R@10, and are comparable to Florence (0.8% worse R@10). We outperform all other models in this category (Table 3.1A).

**B. CLIP-based model with simple finetuning.** Our approach, SRL-CLIP, belongs to this category. Comparison to other methods allows us to disentangle the significance of the structured SRL annotations from other aspects (model pretraining, model size, *etc*.). CLIP4Clip uses a Transformer-based frame feature aggregator, and is post-pretrained on 380k (0.4M) videos from HowTo100M [3] (about 40M video-narration pairs). ViFi-IFT use simple mean pooling, and is fine-tuned on 300k video-action prompts from Kinetics [4]. Our approach, SRL-CLIP, fine-tuned on 23k videos, and using simple mean pooling over frame representations, improves over both approaches on all metrics. *E.g.* R@10 improves by 7.4% and 3.9% on LSMDC and 4.9% and 4.7% on MSRVTT in comparison to CLIP4Clip and ViFi-IFT respectively. Compared to the original CLIP model (ViT B/16), R@10 improves by 3.4% on LSMDC and 6.5% on MSRVTT. Fig. 3.3 (right) shows some qualitative results. We observe that SRL-CLIP is able to correctly understand fine-grained details such as red dress (example 1) or associate multi-person events across multiple shots (example 3) better than base CLIP.

**C. CLIP-based models with sophisticated extensions.** In this part, we compare against models with much larger sizes (*e.g*. BT-Adapter [107] at 450M, VAST [123] at 1.3B parameters) that are often trained with multiple modalities (*e.g*. ImageBind [120], VAST), with post-pretraining datasets $85-6700\times$ larger than our 23k videos. Compared against SRL-CLIP, we see some mixed results: (i) TVTSv2 [122] is worse on LSMDC (2.4% R@10), but better on MSRVTT (1.4% R@10). (ii) ImageBind [120], on MSRVTT, has worse R@10 by 1.8%, but higher R@5 by 2.1%. (iii) UMT-L [121] has comparable R@10 for LSMDC, but is worse by 5.1% R@10 on MSRVTT. Finally, BT-Adapter and VAST perform better than SRL-CLIP (1.7% and 2.1% R@10 on MSRVTT), however, they have significantly higher model footprints (450M and 1.3B) and are post-pretrained on 2M and 154M videos (compared to our 23k videos).

These results highlight the efficacy achievable through post-pretraining on dense and structured SRL, as opposed to narrations or captions.

**Using synthetic SRLs from Kinetics-700.**

Table 3.2 shows the results for adapting CLIP with K-SRL and VidSitu. Even though the K-SRL annotations are noisy, we improve R@5 by 3.2% for CLIP B/32 (row3 *vs*. row1), and 5.1% for CLIP L/14 (row7 *vs*. row5). Training on K-SRL followed by VidSitu further improves over only training on VidSitu. For ViT L/14, row8 *vs*. row6 shows 1.4% reduction in MnR and 1.2% increase in R@10.

*New SotA among CLIP-based models.* Our model trained on K-SRL and VidSitu achieves a new SoTA on zero-shot MSRVTT text-to-video retrieval, outperforming the best visual CLIP-based model BT-Adapter [107] that also uses CLIP ViT L/14. Our model also outperforms VAST [123] on R@10, even though VAST is a multimodal model that uses video, audio, and subtitles.

**Larger models show consistent improvements.** Table 3.2 shows additional zero-shot (ZS) text-to-video retrieval (T2V) experiments with CLIP ViT L/14 on MSRVTT. All CLIP variants show a consistent $\sim$5% improvement in R@5 even though the dataset is the same (23k videos): (i) B/32 53.2 to 58.2 (row1, row2), (ii) B/16 55.0 to 59.7 (in main paper), (iii) L/14 59.0 to 64.7 (row5, row6). SRL-

CLIP with ViT L/14 (450M params) trained efficiently on dense text prompts from 23k videos surpasses Florence, a sophisticated model (637M params) trained on 900M samples.

### 3.2.2 Holistic Video Understanding

We evaluate SRL-CLIP's holistic video understanding ability from two perspectives: (i) VidSitu [5] requires fine-grained perception to generate complex structured outputs; and (ii) Dense video captioning [6] requires localizing and describing actions in longer (few minute) videos.

**Video situation recognition** (VidSitu) [5] requires a model to predict the primary action verb in each event and generate noun captions for the verb-appropriate set of roles. We adopt VideoWhisperer's three-stage architecture [127] as it is the SoTA on VidSitu and remove the 165 object (Faster-RCNN) features and 5 event/action (SlowFast) features. Instead, as a baseline, we only represent the $P$=5 events as mean-pooled CLIP encoded frames. In Table 3.3 we see that CLIP features perform on par with previous best methods on verbs (action understanding) achieving close to 45% Vb@1 accuracy. However, they perform much worse on CIDEr (55%) that evaluates quality of semantic role labels. This is expected as SRL requires fine-grained understanding which CLIP lacks [128].

Next, we evaluate SRL-CLIP by representing the $P$ event features with event-level contextualized tokens (post VC). Verb accuracy improves by 2.95% for ViT B/16, while we see a large increase in SRL prediction performance with CIDEr improving by 17.9%. We acknowledge that a part of this gain may be due to the in-domain post-pretraining on VidSitu, especially training VC from scratch. Nevertheless, this demonstrates the effectiveness of our adaptation process and the video contextualizer that compresses information effectively. With this, we also set a new SoTA on VidSitu. Finally, in Fig. 3.3, we show that our adapted model is better suited for fine-grained reasoning with focused attention maps and accurate SRL prediction.

**Dense captioning and temporal localization** is evaluated on ActivityNet [6]. We adopt PDVC [129, 130], a SoTA approach for dense captioning, where frame-level features are obtained using CLIP or SRL-CLIP. PDVC is a single-stage model that performs video localization and captioning using a DETR-style [131] architecture and two separate heads for event localization and captioning. Table 3.4 shows that a slight drop in CIDEr by 1.2% is compensated by a significant improvement in METEOR by 5%. In addition, we see improvements in temporal localization that requires fine-grained understanding as grounding recall and precision improve by 1.05% and 2.64% respectively.

### 3.2.3 Ablations

We now show the impact of various design choices, with results on VidSitu. All ablations are performed on the CLIP ViT B/32 model.

**How to adapt CLIP?** We start with identifying how and which layers of the backbone should be adapted. Table 3.5 shows various options with full (F) or partial fine-tuning (P), and using LoRA modules (L) on the image encoder (IE) and text encoder (TE). Row 2 (R2) *vs*. R3 and R4 *vs*. R5 show that freezing TE improves performance. We suspect this is because it encourages IE to align with fine-grained descriptions. R5 shows good SRL CIDEr performance with LoRA for IE and frozen TE. Comparing R1 *vs*. R5 we see that full fine-tuning (F) is not only costly, but also performs slightly worse than LoRA (1.1% Vb@1 and 1% CIDEr), likely due to concept forgetting. Hard negatives (R6) further improve results over R5 slightly, especially 1.2% on Vb@1.

**Impact of loss functions** is presented in Table 3.6. Directly adapting the CLIP backbone with the CLIP-Event and CLIP-Video losses (R1-2) results in poor SRL performance on VidSitu. The benefits of a video contextualizer (VC), both during training and downstream evaluation, are seen in R4 that uses VC-Event and VC-Video losses, achieving good results on VidSitu SRL (R4: 70.0% *vs*. R2: 57.3% CIDEr). R5-7, combine both CLIP and VC losses. We choose R7 as the default model as it achieves the highest geometric mean (good trade-off) between Vb@1 and CIDEr scores.

**How to create text prompts?** In Table 3.7, we show the results with different prompting strategies. We see a small performance gain of 1.6% CIDEr when we move from natural language prompts generated by LLaMa2 (R1) to template-based prompts (R2). This shows the benefit of explicitly mentioning the role labels (*e.g*. walker, thing sitting, manner) in the prompt. Including verb-role hard negatives (R3) boosts verb prediction performance (Vb@1 by 1.2% over R2) and SRL CIDEr by a small amount.

**When should VC be used?** We investigate the need for VC and its influence on downstream results in Table 3.8. We evaluate on two tasks, in-domain VidSitu and zero-shot T2V retrieval on MSRVTT. First, row 1 (R1) shows results for the original CLIP model without adaptation, copied here for ease of comparison. In R2, we directly use CLIP-Video and CLIP-Event losses; while R3 uses VC only for post-pretraining (PPT). In all three cases, downstream tasks (DT) are performed by extracting features directly from the CLIP backbones.

We observe improvements across all metrics as we step from R1 to R2 to R3. For zero-shot T2V on MSRVTT, we see that R2 (direct CLIP PPT) brings a large improvement over R1 (5.1% R@10). This

Table 3.5: Ablation study for how to adapt CLIP. F: Full fine-tuning, P: Partial fine-tuning (freeze first 5 layers), L: LoRA adapter.

| | Backbone | | | VidSitu | |
|---|---|---|---|---|---|
| | IE | TE | HN | Vb@1 | CIDEr |
| 1 | F | - | - | 43.53 | 70.27 |
| 2 | P | P | - | 42.83 | 67.83 |
| 3 | P | - | - | 42.87 | 71.00 |
| 4 | L | L | - | 45.53 | 70.14 |
| 5 | L | - | - | 44.65 | 71.27 |
| 6 | L | - | ✓ | **45.88** | **71.50** |

Table 3.6: Ablation study on impact of loss functions. CE: CLIP-Event, CV: CLIP-Video, VCE: VC-Event, VCV: VC-Video.

| | Losses | | | | | VidSitu | |
|---|---|---|---|---|---|---|---|
| | CE | CV | VCE | VCV | HN | Vb@1 | CIDEr |
| 1 | ✓ | - | - | - | - | 44.77 | 58.37 |
| 2 | ✓ | ✓ | - | - | - | 45.38 | 57.34 |
| 3 | - | - | ✓ | - | - | 43.75 | 71.19 |
| 4 | - | - | ✓ | ✓ | - | 45.12 | 70.02 |
| 5 | ✓ | - | ✓ | ✓ | - | 44.46 | **72.10** |
| 6 | ✓ | ✓ | ✓ | ✓ | - | 44.65 | 71.27 |
| 7 | ✓ | ✓ | ✓ | ✓ | ✓ | **45.88** | 71.50 |

highlights the benefit of using structured SRL for efficient CLIP video adaptation. Nevertheless, including VC results in a further 1.3% improvement on R@10 (R3). On VidSitu, we see steady improvements on CIDEr: 2.2% from R1 to R2, and 2.5% from R2 to R3.

Next, we analyze what happens when using VC in both the post-pretraining and the downstream task (R4). Specifically, we use the $P$ contextualized event representations $\hat{\mathrm{e}}_{ik}$ for VidSitu and the single contextualized video representation $\hat{\mathrm{v}}_i$ for MSRVTT. The VC helps compress and contextualize multiple events in a video, leading to strong performance improvements on VidSitu (SRL CIDEr improves by 11.7% over R3). As VC is trained only on VidSitu, it learns to accommodate the complexity of VidSitu. For example, the overall video representation is trained to match against a dense description of multiple events. However, when VC is used directly on MSRVTT, the performance collapses as the captions in MSRVTT are not as descriptive.

**How many Hard Negatives to use?** We show the impact of varying the number of hard negatives (HNs) in Table 3.9. Here, we see that as $\mathcal{N}_{vr}$ increases, the verb prediction accuracy increases while CIDEr drops by a small amount. This is expected as verb-role HNs tend to improve verb prediction performance. The best performance is considered at the geometric mean between Vb@1 and CIDEr on the VidSitu task.

Table 3.7: Ablation study for how to create text prompts.

| | VidSitu | |
|---|---|---|
| Method | Vb@1 | CIDEr |
| 1 LLaMa2 | 44.88 | 69.66 |
| 2 Template | 44.65 | 71.27 |
| 3 Template+HN | **45.88** | **71.50** |

Table 3.8: Ablation study on when should VC be used. PPT: Post-pretraining, DT: downstream-task.

| | VC | | VidSitu | | MSRVTT | | | |
|---|---|---|---|---|---|---|---|---|
| | PPT | DT | Vb@1 | CIDEr | R@5 | R@10 | Mn.R | Md.R |
| 1 | CLIP | | 44.68 | 55.14 | 53.2 | 63.0 | 41.2 | 4 |
| 2 | - | - | 45.38 | 57.34 | 57.49 | 68.1 | 29.5 | 4 |
| 3 | ✓ | - | 46.65 | 59.83 | **58.2** | **69.4** | **27.8** | **3** |
| 4 | ✓ | ✓ | **45.88** | **71.50** | 15.90 | 24.7 | 120.2 | 45.5 |

**What is the best LoRA configuration?** In Table 3.10, we study the impact of adapting different weights of the SRL-CLIP image encoder with LoRA. We get the best performance when we include LoRA modules only for the attention weights in the Transformer ("q k v" – $W_q$, $W_k$, $W_v$) while keeping everything else frozen. Hence, in our default model, we only adapt the self-attention matrices (parameters) with LoRA.

**More examples of *natural* and *artificial* hard negatives** are shown in Table 3.11. Notice how the naturally occurring hard negatives are good enough to learn strong video representations even without the need for artificial hard negatives. Different from most works that only use text-based negatives, VidSitu also facilitates visual natural negatives for the same text. Note how the hard negative captions are very plausible; in example 1 the action *look* instead of *speak*.

## 3.3 Qualitative Results

We now present qualitative results on 6 datasets. When not mentioned otherwise, we use the default variant of SRL-CLIP.

**MSRVTT.** We show zero-shot text-to-video retrieval on the MSRVTT dataset in Fig. 3.4. We can see that SRL-CLIP performs much better than CLIP when the queries have a compositional nature. The last row shows a failure case. Although SRL-CLIP retrieves a video in which a man is talking, he is not talking about hiking. It is hard to pick the right video just by using the visual modality, as hiking is not

Table 3.9: Impact of varying the number of verb-role hard negatives, $\mathcal{N}_{vr}$.

| | VidSitu | |
|---|---|---|
| $\mathcal{N}_{vr}$ | Vb@1 | CIDEr |
| 0 | 44.65 | 71.27 |
| 1 | 44.05 | **72.10** |
| 2 | 44.15 | 71.98 |
| 3 | 45.74 | 71.47 |
| 4 | **45.88** | 71.50 |

Table 3.10: Impact of adapting different weights using LoRA. q, k, v, and o are the query, key, value, and output projection matrices in the self-attention block. fc and proj are the two MLPs after the self-attention module.

| | VidSitu | |
|---|---|---|
| Weight type | Vb@1 | CIDEr |
| q k v | **45.88** | **71.50** |
| q k v o | 45.39 | 70.27 |
| q k v o fc | 44.58 | 69.52 |
| q k v o fc proj | 42.81 | 69.54 |

very clear by just watching the video. In fact, SRL-CLIP retrieves a video shot outdoors, which may have be associated with hiking rather than the indoor video.

**LSMDC.** We show zero-shot text-to-video retrieval on the LSMDC dataset in Fig. 3.5. LSMDC is a much harder dataset compared to MSRVTT, as it is based on movies that contain more dynamic shot changes. Also, the agent/patient of an action is annotated as "SOMEONE", unlike VidSitu, where they are described according to their characteristics, making it even more challenging. We can see that SRL-CLIP outperforms CLIP here as well.

**VidSitu.** Fig. 3.6 shows the qualitative results on video situation recognition for 5 videos. SRL-CLIP outperforms CLIP, especially when picking attributes like color. SRL-CLIP is also better at predicting the role *manner* (which captures the expression/emotion of the person), which CLIP struggles with. However, both SRL-CLIP and CLIP show similar (good) performance when predicting the *scene*. The last row shows a failure case (note that CLIP also fails to give good noun captions in this case). It is interesting to see that SRL-CLIP correctly identifies the *reacher* as a boy but assigns the wrong attribute to it.

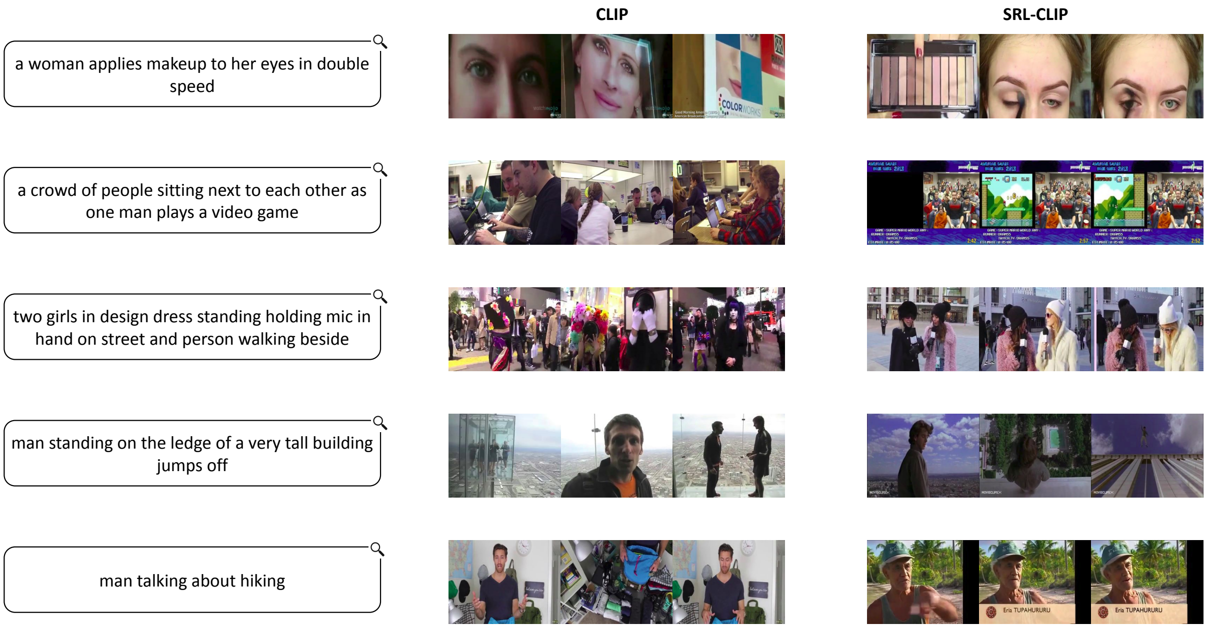



Figure 3.4: Zero-shot text-to-video retrieval on the MSRVTT dataset. We show three frames of the top-1 retrieved video for each query. We can see that SRL-CLIP outperforms CLIP, specially when compositional reasoning is required. The last row shows a failure case. Although SRL-CLIP retrieves a video in which a man is talking, and potentially with more appropriate background, he is not talking about hiking.


Figure 3.5: Zero-shot text-to-video retrieval on the LSMDC dataset. We show three frames of the top-1 retrieved video for each query. We can again notice that SRL-CLIP performs better than CLIP when compositional reasoning is needed. The last row shows a failure case.

Table 3.1: Zero-shot text-to-video retrieval on LSMDC and MSRVTT. Metrics are recall at 5 (R5↑), at 10 (R10↑) and mean and median rank (MnR, MdR↓). Closest to our work, models in the middle section inherit CLIP, and are thus pretrained on 400M samples [1]. They are further post-pretrained: CLIP4Clip on HowTo100M-380k [2], a subset of HowTo100M [3] with 380k videos; ViFi on Kinetics-400 [4] with 300k clips; and our SRL-CLIP on VidSitu's 23k videos [5]. See Section 3.2.1 for details.

| | | | LSMDC | | | | MSRVTT | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | DSize | Params | R5 | R10 | MnR | MdR | R5 | R10 | MnR | MdR |
| *A. Non-CLIP based models* | | | | | | | | | | |
| VideoCLIP [43] | 1M | - | - | - | - | - | 22.2 | 30.0 | - | - |
| Frozen [55] | 5M | 232M | - | - | - | - | 44.6 | 56.6 | - | 7 |
| Clover [112] | 5M | - | 29.2 | 38.2 | - | 24 | 49.5 | 60.0 | - | 6 |
| Singularity [113] | 5M | 209M | - | - | - | - | 50.2 | 59.5 | - | - |
| HiTeA [114] | 5M | 297M | 31.1 | 39.8 | - | - | 54.2 | 62.9 | - | - |
| ALPRO [115] | 5.5M | 231M | - | - | - | - | 44.7 | 55.4 | - | - |
| OmniVL [116] | 18M | - | - | - | - | - | 58.4 | 66.6 | - | - |
| VideoCoCa [117] | 100M | 2.1B | - | - | - | - | 57.8 | 67.0 | - | - |
| VIOLET [118] | 183M | 198M | - | - | - | - | 49.5 | 59.7 | - | - |
| Florence [119] | 900M | 637M | - | - | - | - | 63.8 | 72.6 | - | - |
| *B. CLIP based models with simple post-pretraining* | | | | | | | | | | |
| CLIP ViT B/32 | - | 150M | 28.9 | 35.7 | 130 | 31 | 53.2 | 63.0 | 41.2 | 4 |
| CLIP ViT B/16 | - | 150M | 32.4 | 40.4 | 120 | 21 | 55 | 65.3 | 37.5 | 4 |
| CLIP4Clip [2] | 0.4M | 150M | 28.5 | 36.4 | 117 | 28 | 57.0 | 66.9 | 34 | 4 |
| ViFi [47] ViT B/16 | 0.3M | 150M | 10.6 | 14.8 | 296 | 199 | 26.6 | 33.4 | 171 | 41 |
| ViFi-IFT [47] ViT B/16 | 0.3M | 150M | 32.6 | 39.9 | 125 | 25 | 57.6 | 67.1 | 36.8 | **3** |
| SRL-CLIP ViT B/32 | **23k** | 150M | 31.1 | 39.2 | 103 | 23.5 | 58.2 | 69.4 | 27.8 | **3** |
| SRL-CLIP ViT B/16 | **23k** | 150M | **35.7** | **43.8** | **91.9** | **17** | **59.7** | **71.8** | **26.8** | **3** |
| *C. CLIP based models with sophisticated architecture/modality extensions* | | | | | | | | | | |
| BT-Adapter [107] | 2M | 450M | 35.9 | 45.0 | - | - | 64.7 | 73.5 | - | - |
| ImageBind [120] | 3M | 1B | - | - | - | - | 61.8 | 70.0 | - | - |
| UMT-L [121] | 5M | 304M | **37.2** | 43.7 | - | - | 58.1 | 66.7 | - | - |
| TVTSv2 [122] | 8.5M | 1B | 32.5 | 41.4 | - | 20 | 62.4 | 73.2 | - | 3 |
| VAST [123] | 154M | 1.3B | - | - | - | - | **68.3** | **73.9** | - | - |

Table 3.2: ZS T2V retrieval performance on MSRVTT. (i) CLIP model sizes ViT B/32 *vs*. ViT L/14. (ii) Adaptation dataset: VidSitu (from the main paper) and synthetic dense SRL annotations from Kinetics (K-SRL).

| # | Method | DSize | VidSitu | K-SRL | R@5 | R@10 | MnR | MdR |
|---|---|---|---|---|---|---|---|---|
| 1 | CLIP ViT B/32 | - | - | - | 53.2 | 63.0 | 41.2 | 4 |
| 2 | SRL-CLIP ViT B/32 | 23k | ✓ | - | **58.2** | 69.4 | 27.9 | 3 |
| 3 | SRL-CLIP ViT B/32 | 40k | - | ✓ | 56.4 | 66.4 | 30.7 | 3 |
| 4 | SRL-CLIP ViT B/32 | 63k | ✓ | ✓ | 58.0 | **70.3** | **27.2** | 3 |
| 5 | CLIP ViT L/14 | - | - | - | 59.0 | 69.6 | 33.6 | 3 |
| 6 | SRL-CLIP ViT L/14 | 23k | ✓ | - | 64.7 | 72.8 | 23.9 | 3 |
| 7 | SRL-CLIP ViT L/14 | 40k | - | ✓ | 64.1 | 73.8 | 26.6 | 3 |
| 8 | SRL-CLIP ViT L/14 | 63k | ✓ | ✓ | **65.0** | **74.0** | **22.5** | 3 |

Table 3.3: Performance on VidSitu [5]. Action recognition measured as top-1 and top-5 verb accuracy (Vb@1, Vb@5). Captioning performance measured through CIDEr. We see a large performance improvement over base CLIP, while also achieving a new SoTA.

| Method | Vb@1 ↑ | Vb@5 ↑ | CIDEr ↑ |
|---|---|---|---|
| VidSitu [5] | 46.79 | 75.90 | 46.01 |
| Slow-D+TxE+TxD [126] | - | - | 60.34 |
| VideoWhisperer [127] | 45.06 | 75.59 | 68.54 |
| CLIP (ViT B/32) | 44.68 | 79.79 | 55.14 |
| SRL-CLIP (ViT B/32) | 45.88 | 80.66 | 71.50 |
| CLIP (ViT B/16) | 45.83 | 80.12 | 54.25 |
| SRL-CLIP (ViT B/16) | **48.78** | **81.95** | **72.11** |

Table 3.4: Dense video captioning on ActivityNet [6]. CIDEr and METEOR estimate captioning quality. Grounding precision and recall evaluate localization.

| Method | CIDEr ↑ | METEOR ↑ | G Rec. ↑ | G Prec. ↑ |
|---|---|---|---|---|
| CLIP | **29.50** | 79.07 | 50.04 | 54.37 |
| SRL-CLIP | 28.27 | **84.09** | **51.09** | **57.01** |

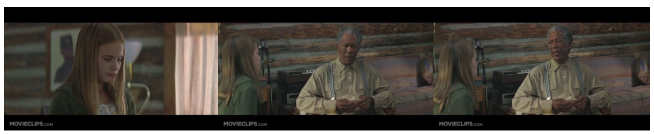
| Positive Prompt | Natural Hard Negatives | Verb-role Hard Negatives |
|---|---|---|
| In this photo, the *action* is speak where, the *talker* is man standing in yellow sweatshirt, the *hearer* is woman with scarf, the *manner* is standing in the middle of a full airplane, the *scene* of the event is an airplane. | In this photo, the *action* is turn where, the *the turner* is man standing in yellow sweatshirt, the *the thing turning* is his body, the *direction* is towards woman with scarf, the *scene* of the event is an airplane. | In this photo, the *action* is look where, the *looker* is man standing in yellow sweatshirt, the *thing looked at* is woman with scarf, the *direction* is *is to his back*, *manner* is standing in the middle of a full airplane, the *scene* of the event is an airplane. |
| In this photo, the *action* is open where, the *opener* is man in brown jacket and man in gray suit, the *the thing opening* is trunk of taxi, the *manner* is annoyed, the *scene* of the event is near a taxi. | In this photo, the *action* is hoist where, the *lifter* is man in brown jacket and man in gray suit, the *thing going up* is dead body, the *direction* is up into trunk of taxi, the *scene* of the event is near a taxi. | In this photo, the *action* is respond where, the *replier* is man in brown jacket and man in gray suit, the *scene* of the event is near a taxi. |
| In this photo, the *action* is bow where, the *bower* is the woman in glasses, the *bowed to* is man wearing black, the *manner* is on her knees, the *scene* of the event is in a well lit room. | In this photo, the *action* is photograph, take a picture where, the *photographer* is the man in black suit, the *subject* is woman in glasses, *scene* of the event is in a well lit room. | In this photo, the *action* is smash where, the *smasher* is the woman in glasses, the *smashed* is man wearing black, the *direction* is on patients face, the *scene* of the event is in a well lit room. |

Table 3.11: We show the naturally occurring hard negatives in a batch as well as the process of converting a standard positive prompt into hard negatives by swapping verb-role information. The template is shown in gray, *e.g.* In this photo,. The action and roles are shown in *italics*, *e.g. action, talker, hearer*. The correct prompt values (verbs or nouns) are in cobalt blue, *e.g. speak, man standing in yellow sweatshirt*; and the replaced verbs, roles, or nouns are in deep red. We swap the verb and roles in verb-role hard negatives while keeping the same nouns and performing some mapping between previous and new roles.

| | verb | driver | vehicle | direction | manner | scene |
|---|---|---|---|---|---|---|
| **CLIP** | drive | man in hat | car | down the street | slowly | in a car |
| **SRL-CLIP** | drive | man in black jacket | car | down the road | with a serious look on his face | in a car |
| **GT** | drive | man in a black police uniform | car | forward | intently | car |

| | verb | entity entering | thing entered | manner | scene |
|---|---|---|---|---|---|
| **CLIP** | walk | man in blue shirt | the house | with his right hand | outside a house |
| **SRL-CLIP** | enter | boy in red shirt | the door | slowly | outside a building |
| **GT** | enter | kid in red | door | casually | in doorway |

| | verb | looker | looked at | direction | manner | scene |
|---|---|---|---|---|---|---|
| **CLIP** | stare | woman with dark hair | man in green shirt | down | with a slight smile | in a room |
| **SRL-CLIP** | stare | woman with blonde hair | man in brown jacket | down | with a sad expression | in a room |
| **GT** | look | girl with blonde hair | a man in front of her | forward | sadly | in a room |



| | verb | talker | hearer | manner | scene |
|---|---|---|---|---|---|
| **CLIP** | look | girl with dark hair | man in blue shirt | while seated next to each other | in a room |
| **SRL-CLIP** | speak | woman in a green shirt | man in brown shirt | while face to face | in a kitchen |
| **GT** | speak | blonde girl | old man | while sitting down | cabin |

| | verb | reacher | body part | goal | direction | purpose | scene |
|---|---|---|---|---|---|---|---|
| **CLIP** | kneel | woman in blue coat | hand | to grab something | down | to get something | in a room |
| **SRL-CLIP** | kneel | boy in orange shirt | his body | to grab something | down | to pick up a plate | in a room |
| **GT** | grab | boy | his hand | booklet | towards the man | to take booklet from man | lab |

Figure 3.6: Video Situation Recognition on 5 videos. SRL-CLIP performs much better than CLIP in picking the right attributes of an entity. The last row shows a failure case where the semantic role labels predicted by SRL-CLIP deviates from the ground-truth (GT).

*Chapter 4*

# The Multimodal Challenge: Fine-Grained Understanding, Discrimination, and Compositional Reasoning

In machine learning, generative and contrastive training represent two fundamental paradigms, each with certain strengths and weaknesses. In generative training, we aim to model the underlying distribution directly, so that we can generate new samples that are similar to the training data. A common method of achieving this is to maximize the likelihood of the observed data. Contrastive training, on the other hand, involves contrasting similar and dissimilar examples in order to learn representations. Instead of explicitly modeling the data distribution, it uses relative similarity to distinguish between instances. As a result, discriminative representations are often formed, which is beneficial for tasks such as classification and retrieval.

In this chapter, we explore the nuances of one such effective instantiation of contrastive learning called *self-retrieval*. In self-retrieval, a model generates a caption for an image (the anchor) and then uses that caption to retrieve the original image from a collection of distractors (negative examples). This framework compels the model to generate discriminative captions, ensuring that they capture the unique and essential features of the anchor image. Moving beyond image captioning, we will then explore the evaluation of compositionality in video language models, examining how both contrastive and generative approaches fare in understanding the intricate relationships between entities, actions, and temporal sequences within the video. To assess this ability, we utilize the *VELOCITI* benchmark, which probes models' understanding of complex video narratives by evaluating their ability to identify key entities, actions, and temporal relationships.

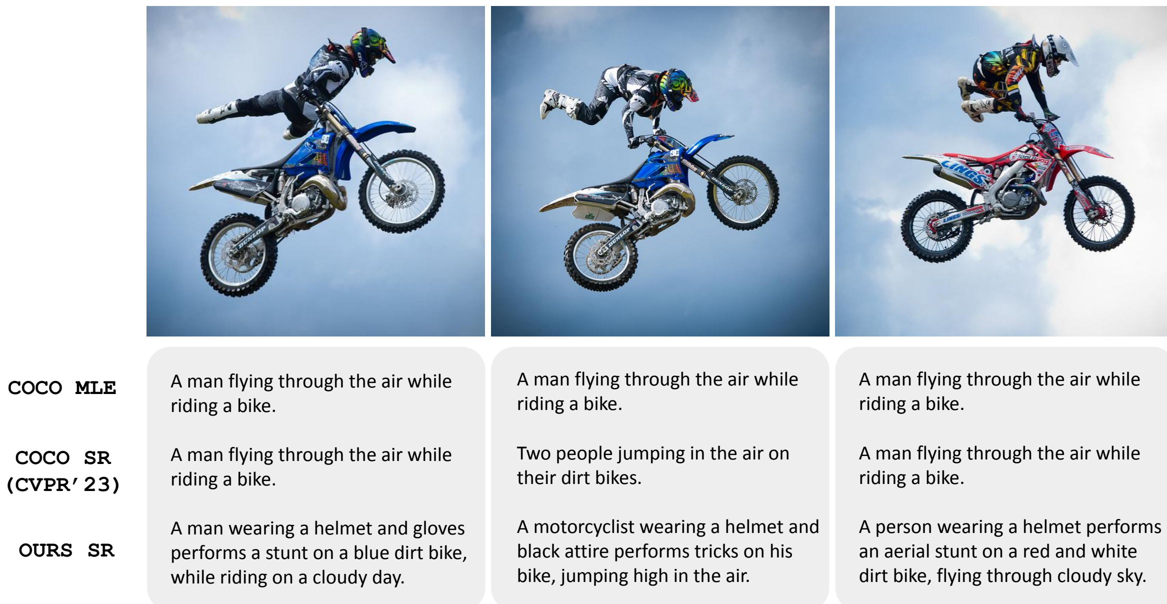



Figure 4.1: For a set of similar images, captioning systems struggle to generate meaningful captions that uniquely describe each image. **COCO MLE**: A model trained on COCO with MLE generates the same description for all images. **COCO SR** [7]: While the SR objective may help, the COCO captions are not rich enough and lead to hallucinations such as "two people" (middle). **OURS SR**: Our improved visual captions and SR fine-tuning with hard negatives results in discriminative captions.

## 4.1 No Detail Left Behind: Revisiting Self-Retrieval for Fine-Grained Image Captioning

Image captioning, or generating natural language image descriptions, has witnessed remarkable progress over the last decade. Today's captioning systems are composed of sophisticated deep learning architectures [132, 133, 105, 134] trained on vast datasets [135, 136, 137, 138, 139]. However, even with these advances, approaches often generate generic captions that are unable to differentiate similar images (see Figure 4.1), violating the fundamental purpose of a caption: *to facilitate accurate and efficient communication of visual content* [140, 141, 7]. We attribute the shortcomings of image captioning systems to three key factors: (i) the nature of their training data, (ii) captioning evaluation metrics, and (iii) the maximum likelihood estimation (MLE) training approach.

In this work, we circumvent this bottleneck by improving the MLE initialization of the captioning system and designing a curriculum for the SR fine-tuning process. To this extent, we present (1) Visual Caption Boosting, a novel framework to instill fine-grainedness in generic image captioning datasets while remaining anchored in human annotations; and (2) BagCurri, a carefully designed training curriculum that more optimally leverages the contrastive nature of the self-retrieval reward. Jointly, they enable the captioner to describe fine-grained aspects in the image while preserving faithfulness to ground-truth captions. Our approach outperforms previous work by +8.9% on SR against 99 random distractors (RD100) [7]; and +7.6% on ImageCoDe.

Additionally, existing metrics to evaluate captioning systems fail to reward diversity or evaluate a model's fine-grained understanding ability. Our third contribution addresses this by proposing self-retrieval from the lens of evaluation. We introduce TrueMatch, a benchmark comprising bags of highly similar images that uses SR to assess the captioner's ability to capture subtle visual distinctions. We evaluate and compare several state-of-the-art open-source MLLMs on TrueMatch, and find that our SR approach outperforms them all by a significant margin (*e.g.* +4.8% - 7.1% over Cambrian) while having 1-2 orders of magnitude fewer parameters. We also outperform vanilla SR by +14.4% to +19.5%.

## 4.2 Detect, Describe, Discriminate: Moving Beyond VQA for MLLM Evaluation

Visual Question Answering (VQA) with multiple choice questions enables a vision-centric evaluation of Multimodal Large Language Models (MLLMs). Although it reliably checks the existence of specific visual abilities, it is easier for the model to select an answer from multiple choices (VQA evaluation) than to generate the answer itself (see Figure 4.2). In this work, we offer a novel perspective: we evaluate how well an MLLM understands a specific visual concept by its ability to uniquely describe two extremely similar images that differ only in the targeted visual concept. Specifically, we assess the ability of MLLMs to capture specific points of visual differences using self-retrieval [142], *i.e.* by retrieving the target image using its generated caption against the other image in the pair serving as the distractor. We curate 247 highly similar image pairs as part of the $\mathbf{D_3}$ benchmark. For each image pair, the model is prompted to: (1) **Detect** a specific visual difference, and (2) **Describe** the target image uniquely such that it (3) **Discriminate**s the target image from the distractor. Self-retrieval within $D_3$ enables whitebox evaluation across six different visual patterns, revealing that current models struggle to independently discern fine-grained visual differences, with open-source models failing to outperform random guess.

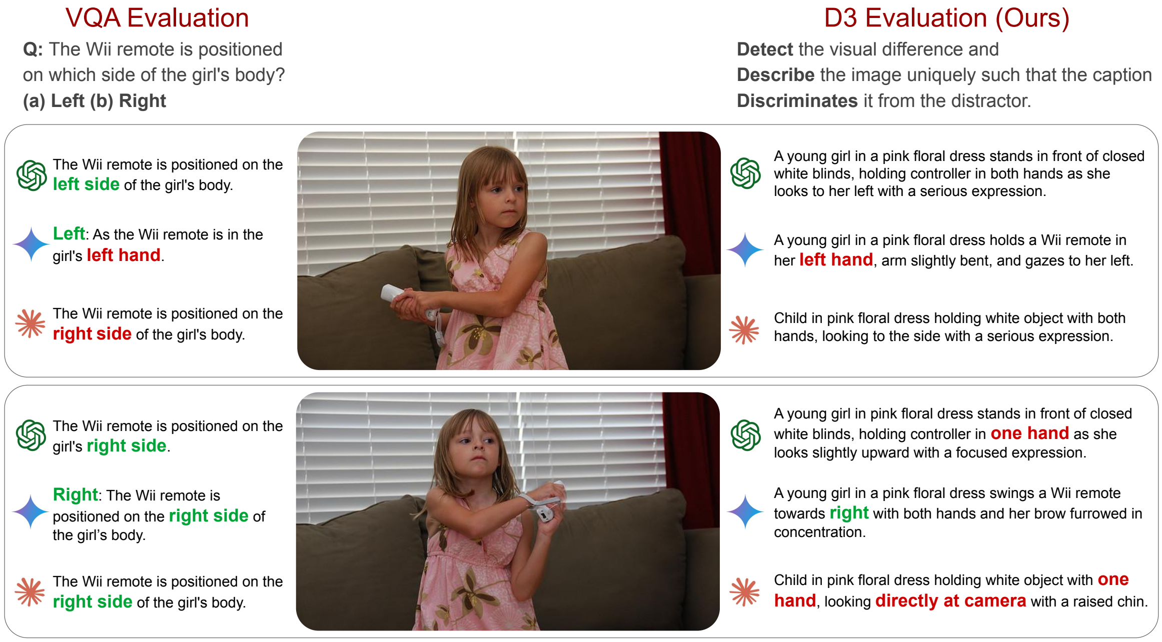



Figure 4.2: When prompted with a question and/or multiple choices (*VQA evaluation*), MLLMs show middling performance on identifying fine-grained differences between an image pair. Harder still, is when MLLMs need to independently detect and describe such differences (*our evaluation*). Our work finds that state-of-the-art MLLMs struggle to discern fine-grained difference with our detect-describe-discriminate evaluation framework, with open-source MLLMs failing to outperform random guess. The text highlighted in **green** represents the fine-grained differences captured by the MLLMs, while that marked in **red** represents erroneous descriptions (hallucinations). Results are presented for GPT-4o, Gemini-1.5-Pro, and Claude-Sonnet-3.5.

## 4.3 VELOCITI: Can Video-Language Models Bind Semantic Concepts through Time?

Comprehensive video-language understanding requires seamless alignment between vision and language modalities. Although progress has been made in several benchmarks [143, 144, 145], current state-of-the-art models struggle to discriminate between similar videos/descriptions such as *A girl wearing a hat is holding a dress* and *A girl wearing a dress is holding a hat*, something trivial for humans. We hypothesize that this failure stems from the lack of compositionality in the model's representation. While several works (*e.g.* Winoground [146], SugarCrepe [147], Cola [148]) have studied this for image-based models, this is a serious challenge for videos as they contain multiple persons, objects, and scenes that dynamically interact and change over *time*.

Compositionality involves the **perception** (identification) of atomic entities like persons, actions, objects, and scenes and **binding** these entities with each other through the right relationships. Existing evaluation benchmarks mainly focus on the former along with common-sense reasoning. Typical questions from SEED-Bench [143] address high level semantics: *Where is the dog located in the living room?*, or *What is the material of the table?*. Typical examples from MVBench [144] include: *What is the pose performed by the person?*, or *What will the person do after reading a book?* Such questions can often be answered by analyzing action or object cues independently, without requiring understanding the underlying compositional spatio-temporal scene dynamics. Examples from a recent work, Video-Con [145], contrast: *A person riding a brown horse. vs. A person riding a green horse.* Such examples may not even require visual understanding as green horses are unlikely.

We propose *VELOCITI* (Video et Language Compositionality through Time), a new benchmark to objectively evaluate compositionality in video language models (both contrastive and LLM-based). The tests in VELOCITI study perception and binding by quantifying the ability to discriminate between the correct(positive) and incorrect(negative) captions when given a video or vice-versa (see Figure 4.3). Perception tests require models to identify the presence of entities in the positive caption and the absence of distractor entities in the negative caption. A bag-of-words style representation is sufficient to solve these tests. Binding tests require models to verify the associations between two or more entities. While all entities from both captions appear in the video, the associations are incorrect for the negative caption. A "lady in blue shirt" is "covering her ears" while a "person in white shirt" looks at her. The positive caption is "lady in blue is covering her ears." For perception tests, the negative caption is "man in hat is covering his ears". A model with bag-of-words capabilities that can identify the presence of "woman in blue" and the absence of "man in hat" can solve this task. For binding tests, the negative caption is "person in white shirt is covering her ears". As the person in white also appears in the video, a bag-of-words model cannot disambiguate the positive and negative caption. To solve this test, the model must identify associations such as *covering(lady in blue, ears)* and *looking(person in white, lady in blue)* and infer *NOT covering(person in white, ears)*.

**Perception tests** comprise the following: Control Tests check whether models can differentiate between easy video-caption negatives with very different entities, establishing a baseline ability. Extending this, Intra-Video Association Testevaluates models' ability to contrast between two events from the same video that have similar entities. We find that many VLMs struggle at this. The next set of perception tests Agent Identification, Action Adversarial, and Action Modifier tests evaluate understanding of individual concepts one at a time, i.e. agent, action, and action modifiers respectively.

**Binding tests** include tests for studying various aspects of agent-action and event-time associations. Agent binding and Action binding tests require discrimination of agents and actions from the same video based on the associations in the captions. Going a step further, Agent co-reference test studies agent-action associations across events in a video. This ability is crucial to understanding continuity in a video and seems to be challenging even for the best commercial models (*e.g*. Gemini). Finally, Chronology test probes models' ability to bind events within a video to their correct temporal ordering concepts such as before, after, etc (shown to be poor in [149]).

To construct VELOCITI, we adopt VidSitu [5], a video situation recognition dataset that consists of dense annotations with *semantic role labels* (SRLs). Sourced from movie clips, VidSitu videos present challenging scenarios with frequent shot changes, fast action sequences, multi-event complex situations, role switching between the agent (doer) and patient (receiver), and co-referencing of entities, all entangled through time. To parse these complex videos, dense SRL annotations capture multiple and changing aspects of the events in the video, including the *action*, *doer*, *receiver*, *instrument*, *scene*, and *manner* in a structured way [5]. We build on these rich annotations to create a challenging and high-quality benchmark test suite for evaluating compositionality.

In summary: (i) We propose to use rich video annotations in the form of *semantic role labels* (SRLs) together with visually complex movie clips to create VELOCITI, a manually verified and rigorous benchmark for evaluating compositionality in video language models. (ii) We study *perception* and *binding* capabilities using tests within the same benchmark, enabling a comprehensive evaluation of compositionality and novel takeaways. (iii) We demonstrate that contrastive models and Video-LLMs struggle with performance close to or slightly above random, especially on binding tasks. Powerful commercial models like Gemini 1.5 Flash, while significantly better, are also found to be lacking as compared to humans, who have innate expertise at such tasks, with accuracy above $90\%$.

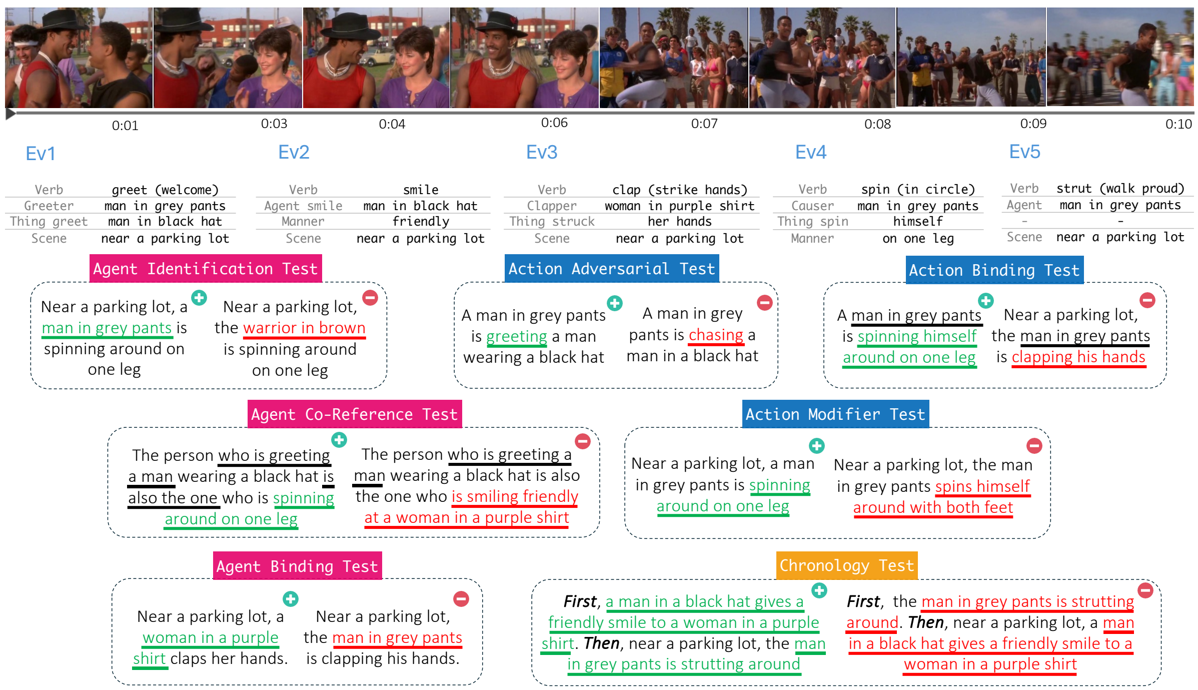



Figure 4.3: The VELOCITI benchmark features complex movie videos with rich semantic role label (SRL) annotations from the VidSitu dataset [5] based on which we create multiple benchmark tests. These require models to perform fine-grained and compositional reasoning with semantic concept binding across agents, actions, and time.

*Chapter 5*

# Conclusion

This work has presented several significant contributions to the field of video representation learning and related multimodal tasks. **AVLectures** introduced a valuable new resource for the educational domain: a large-scale dataset of STEM lectures accompanied by a novel unsupervised lecture segmentation task.

**SRL-CLIP** tackled the challenge of efficiently adapting pre-trained vision-language models like CLIP to video understanding. By utilizing the richer information provided by Semantic Role Labels (SRLs), SRL-CLIP achieved superior performance on zero-shot text-to-video retrieval and downstream video understanding tasks, demonstrating the effectiveness of dense semantic supervision and significantly reducing the reliance on massive video datasets for post-pretraining.

Addressing the crucial aspect of compositional understanding, **VELOCITI** introduced a novel benchmark focusing on the ability of video-language models to bind semantic concepts and suppress distractors. The benchmark revealed significant limitations in current models, particularly in handling complex temporal relationships and co-references.

Furthermore, **NDLB** addressed challenges in fine-grained image captioning and evaluation. NDLB proposed a novel framework (VCB) to enhance datasets with fine-grained information, a new benchmark for evaluating captioning models based on self-retrieval, and a curriculum learning strategy (BagCurri) for effective self-retrieval fine-tuning.

Finally, $\mathbf{D_3}$ introduced a benchmark designed to evaluate the fine-grained visual discrimination capabilities of MLLMs. Using a self-retrieval approach, $D_3$ provided a valuable tool for understanding the strengths and weaknesses of current MLLMs in discerning subtle visual differences.

Collectively, these contributions provide valuable datasets, methods, benchmarks, and insights that advance the field of video representation learning and related multimodal tasks, paving the way for more robust, efficient, and semantically rich visual understanding systems.

# Bibliography


[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," in *ICML*. PMLR, 2021, pp. 8748–8763.

[2] H. Luo, L. Ji, M. Zhong, Y. Chen, W. Lei, N. Duan, and T. Li, "CLIP4Clip: An Empirical Study of CLIP for End to End Video Clip Retrieval," *arXiv preprint arXiv:2104.08860*, 2021.

[3] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips," in *CVPR*, 2019, pp. 2630–2640.

[4] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[5] A. Sadhu, T. Gupta, M. Yatskar, R. Nevatia, and A. Kembhavi, "Visual Semantic Role Labeling for Video Understanding," in *CVPR*, 2021, pp. 5589–5600.

[6] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding," in *CVPR*, 2015, pp. 961–970.

[7] R. Dessì, M. Bevilacqua, E. Gualdoni, N. C. Rakotonirina, F. Franzon, and M. Baroni, "Cross-Domain Image Captioning With Discriminative Finetuning," in *CVPR*, 2023, pp. 6935–6944.

[8] D. Singh S, A. Gupta, C. V. Jawahar, and M. Tapaswi, "Unsupervised audio-visual lecture segmentation," in *Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 5232–5241.

[9] D. Singh S, Z. Khan, and M. Tapaswi, "Seeing Beyond Captions: Efficient CLIP Video Adaptation via Structured Semantic Role Labels," 2024, under review.

[10] D. Saravanan, D. Singh S, V. Gupta, Z. Khan, V. Gandhi, and M. Tapaswi, "VELOCITI: Can Video-Language Models Bind Semantic Concepts through Time?" *arXiv preprint arXiv:2406.10889*, 2024.

[11] M. Gaur, D. Singh S, and M. Tapaswi, "No Detail Left Behind: Revisiting Self-Retrieval for Fine-Grained Image Captioning," *arXiv preprint arXiv:2409.03025*, 2024.

[12] M. Gaur, D. Singh S, and M. Tapaswi, "Detect, Describe, Discriminate: Moving Beyond VQA for MLLM Evaluation," *arXiv preprint arXiv:2409.15125*, 2024.

[13] P. Papalampidi, F. Keller, and M. Lapata, “Movie summarization via sparse graph construction,” in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021.

[14] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele, “Movie description,” *International Conference on Computer Vision (ICCV)*, 2017.

[15] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, “Movieqa: Understanding stories in movies through question-answering,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien, “Unsupervised learning from narrated instruction videos,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, “Howto100m: Learning a text-video embedding by watching hundred million narrated video clips,” in *International Conference on Computer Vision (ICCV)*, 2019.

[18] A. Rouditchenko, A. Boggust, D. Harwath, B. Chen, D. Joshi, S. Thomas, K. Audhkhasi, H. Kuehne, R. Panda, R. Feris, B. Kingsbury, M. Picheny, A. Torralba, and J. Glass, “AVLnet: Learning Audio-Visual Language Representations from Instructional Videos,” in *Proc. Interspeech 2021*, 2021, pp. 1584–1588.

[19] P. Das, C. Xu, R. F. Doell, and J. J. Corso, “A Thousand Frames in Just a Few Words: Lingual Description of Videos through Latent Topics and Sparse Object Stitching,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[20] L. Zhou, N. Louis, and J. J. Corso, “Weakly-supervised video object grounding from text by loss weighting and object interaction,” in *BMVC*, 2018.

[21] S. Bulathwela, M. Perez-Ortiz, E. Novak, E. Yilmaz, and J. Shawe-Taylor, “Peek: A large dataset of learner engagement with educational videos,” *arXiv preprint arXiv:2109.03154*, 2021.

[22] S. Bulathwela, M. Perez-Ortiz, E. Yilmaz, and J. Shawe-Taylor, “VLEngagement: A Dataset of Scientific Video Lectures for Evaluating Population-based Engagement,” *arXiv preprint arXiv:2011.02273*, 2020.

[23] K. Dutta, M. Mathew, P. Krishnan, and C. Jawahar, “Localizing and recognizing text in lecture videos,” in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018.

[24] A. Gandhi, A. Biswas, and O. Deshmukh, “Topic transition in educational videos using visually salient words.” *International Educational Data Mining Society*, 2015.

[25] D. Mahapatra, R. Mariappan, and V. Rajan, “Automatic hierarchical table of contents generation for educational videos,” in *Companion Proceedings of The Web Conference*, 2018.

[26] D. Mahapatra, R. Mariappan, V. Rajan, K. Yadav, and S. Roy, "Videoken: Automatic video summarization and course curation to support learning," in *Companion Proceedings of The Web Conference*, 2018.

[27] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-End Learning of Visual Representations from Uncurated Instructional Videos," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[28] T.-J. Fu, L. Li, Z. Gan, K. Lin, W. Y. Wang, L. Wang, and Z. Liu, "Violet: End-to-end video-language transformers with masked visual-token modeling," *arXiv preprint arXiv:2111.12681*, 2021.

[29] J. Lei, L. Li, L. Zhou, Z. Gan, T. L. Berg, M. Bansal, and J. Liu, "Less is more: Clipbert for video-and-language learning via sparse sampling," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[30] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *International Conference on Computer Vision (ICCV)*, 2015.

[31] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid, "Just Ask: Learning to Answer Questions from Millions of Narrated Videos," in *International Conference on Computer Vision (ICCV)*, 2021.

[32] V. Iashin and E. Rahtu, "Multi-modal dense video captioning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[33] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[34] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] M. Narasimhan, A. Rohrbach, and T. Darrell, "CLIP-It! language-guided video summarization," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[36] K. Desai and J. Johnson, "Virtex: Learning visual representations from textual annotations," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[37] L. Ding and C. Xu, "Tricornet: A hybrid temporal convolutional and recurrent network for video action segmentation," *arXiv preprint arXiv:1705.07818*, 2017.

[38] J. Li, P. Lei, and S. Todorovic, "Weakly supervised energy-based learning for action segmentation," in *International Conference on Computer Vision (ICCV)*, 2019.

[39] Y. Souri, M. Fayyaz, and J. Gall, "Weakly supervised action segmentation using mutual consistency," *arXiv preprint arXiv:1904.03116*, 2019.

[40] S. N. Aakur and S. Sarkar, "A perceptual prediction framework for self supervised event segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[41] A. Kukleva, H. Kuehne, F. Sener, and J. Gall, "Unsupervised learning of action classes with continuous temporal embedding," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[42] S. Sarfraz, N. Murray, V. Sharma, A. Diba, L. Van Gool, and R. Stiefelhagen, "Temporally-weighted hierarchical clustering for unsupervised action segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[43] H. Xu, G. Ghosh, P.-Y. Huang, D. Okhonko, A. Aghajanyan, F. Metze, L. Zettlemoyer, and C. Feichtenhofer, "VideoCLIP: Contrastive Pre-training for Zero-shot Video-Text Understanding," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2021, pp. 6787–6800. [Online]. Available: https://aclanthology.org/2021.emnlp-main.544

[44] M. Wang, J. Xing, and Y. Liu, "ActionCLIP: A New Paradigm for Video Action Recognition," *arXiv preprint arXiv:2109.08472*, 2021.

[45] Y. Ma, G. Xu, X. Sun, M. Yan, J. Zhang, and R. Ji, "X-clip: End-to-end multi-grained contrastive learning for video-text retrieval," in *ACM MM*, 2022, pp. 638–647.

[46] H. Xue, Y. Sun, B. Liu, J. Fu, R. Song, H. Li, and J. Luo, "CLIP-ViP: Adapting Pre-trained Image-Text Model to Video-Language Alignment," in *ICLR*, 2023. [Online]. Available: https://openreview.net/forum?id=GNjzMAgawq

[47] H. Rasheed, M. U. khattak, M. Maaz, S. Khan, and F. S. Khan, "Finetuned CLIP models are efficient video learners," in *CVPR*, 2023.

[48] Z. Qing, S. Zhang, Z. Huang, Y. Zhang, C. Gao, D. Zhao, and N. Sang, "Disentangling Spatial and Temporal Learning for Efficient Image-to-Video Transfer Learning," in *ICCV*, October 2023, pp. 13 934–13 944.

[49] S. Buch, C. Eyzaguirre, A. Gaidon, J. Wu, L. Fei-Fei, and J. C. Niebles, "Revisiting the "Video" in Video-Language Understanding," in *CVPR*, June 2022, pp. 2917–2927.

[50] H. Fang, P. Xiong, L. Xu, and Y. Chen, "CLIP2Video: Mastering Video-Text Retrieval via Image CLIP," *arXiv preprint arXiv:2106.11097*, 2021.

[51] Z. Wang, Y.-L. Sung, F. Cheng, G. Bertasius, and M. Bansal, "Unified Coarse-to-Fine Alignment for Video-Text Retrieval," *arXiv preprint arXiv:2309.10091*, 2023.

[52] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019.

[53] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.

[54] H. Xue, T. Hang, Y. Zeng, Y. Sun, B. Liu, H. Yang, J. Fu, and B. Guo, “Advancing high-resolution video-language representation with large-scale video transcriptions,” in *CVPR*, 2022, pp. 5036–5045.

[55] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, “Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval,” in *ICCV*, 2021, pp. 1728–1738.

[56] M. Patrick, P.-Y. Huang, Y. Asano, F. Metze, A. G. Hauptmann, J. F. Henriques, and A. Vedaldi, “Support-set bottlenecks for video-text representation learning,” in *ICLR*, 2021.

[57] V. Gabeur, C. Sun, K. Alahari, and C. Schmid, “Multi-modal Transformer for Video Retrieval,” in *ECCV*, 2020, pp. 214–229.

[58] J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,” in *CVPR*, July 2017.

[59] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, *et al.*, “The” something something” video database for learning and evaluating visual common sense,” in *ICCV*, 2017, pp. 5842–5850.

[60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is All you Need,” in *NeurIPS*, 2017.

[61] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, “A CLIP-Hitchhiker’s Guide to Long Video Retrieval,” *arXiv preprint arXiv:2205.08508*, 2022.

[62] S. K. Gorti, N. Vouitsis, J. Ma, K. Golestan, M. Volkovs, A. Garg, and G. Yu, “X-Pool: Cross-Modal Language-Video Attention for Text-Video Retrieval,” in *CVPR*, 2022, pp. 5006–5015.

[63] Z. Qing, S. Zhang, Z. Huang, Y. Zhang, C. Gao, D. Zhao, and N. Sang, “Disentangling Spatial and Temporal Learning for Efficient Image-to-Video Transfer Learning,” in *ICCV*, 2023, pp. 13 934–13 944.

[64] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-Efficient Transfer Learning for NLP,” in *ICML*, 2019, pp. 2790–2799.

[65] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-Rank Adaptation of Large Language Models,” in *ICLR*, 2022.

[66] Y.-L. Sung, J. Cho, and M. Bansal, “VL-Adapter: Parameter-Efficient Transfer Learning for Vision-and-Language Tasks,” in *CVPR*, 2022, pp. 5227–5237.

[67] O. Pantazis, G. Brostow, K. Jones, and O. Mac Aodha, “SVL-Adapter: Self-Supervised Adapter for Vision-Language Pretrained Models,” in *BMVC*, 2022.

[68] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao, "Clip-adapter: Better vision-language models with feature adapters," *International Journal of Computer Vision*, pp. 1–15, 2023.

[69] J. Pan, Z. Lin, X. Zhu, J. Shao, and H. Li, "ST-Adapter: Parameter-Efficient Image-to-Video Transfer Learning," in *NeurIPS*, 2022, pp. 26 462–26 477.

[70] H. Jiang, J. Zhang, R. Huang, C. Ge, Z. Ni, J. Lu, J. Zhou, S. Song, and G. Huang, "Cross-modal adapter for text-video retrieval," *arXiv preprint arXiv:2211.09623*, 2022.

[71] S. Hayou, N. Ghosh, and B. Yu, "LoRA+: Efficient Low Rank Adaptation of Large Models," *arXiv preprint arXiv:2402.12354*, 2024.

[72] S.-Y. Liu, C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, and M.-H. Chen, "DoRA: Weight-Decomposed Low-Rank Adaptation," *arXiv preprint arXiv:2402.09353*, 2024.

[73] "Benefits of Using OER," https://oer.psu.edu/benefits-of-using-oer/, 2022.

[74] A. Rao, L. Xu, Y. Xiong, G. Xu, Q. Huang, B. Zhou, and D. Lin, "A Local-to-Global Approach to Multi-modal Movie Scene Segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[75] S. Sarfraz, V. Sharma, and R. Stiefelhagen, "Efficient parameter-free clustering using first neighbor relations," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[76] P. J. Guo and K. Reinecke, "Demographic differences in how students navigate through MOOCs," in *Proceedings of the ACM Conference on Learning @ Scale Conference*, 2014.

[77] G. Verma, T. Nalamada, K. Harpavat, P. Goel, A. Mishra, and B. V. Srinivasan, "Non-linear consumption of videos using a sequence of personalized multimodal fragments," 2021.

[78] S. Bulathwela, S. Kreitmayer, and M. Pérez-Ortiz, "What's in it for me? augmenting recommended learning resources with navigable annotations," 2020.

[79] J. Chen, X. Chen, L. Ma, Z. Jie, and T.-S. Chua, "Temporally grounding natural sentence in video," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[80] M. Perez-Ortiz, C. Dormann, Y. Rogers, S. Bulathwela, S. Kreitmayer, E. Yilmaz, R. Noss, and J. Shawe-Taylor, "X5learn: A personalised learning companion at the intersection of AI and HCI," 2021.

[81] "Massachusetts institute of technology: Mit opencouseware," https://ocw.mit.edu/, 2022, license: Creative Commons BY-NC-SA.

[82] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mpnet: Masked and permuted pre-training for language understanding," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[83] T. Wolf, L. Debut, V. Sanh, J. Chaumond, *et al.*, "Transformers: State-of-the-art natural language processing," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[84] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[85] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[86] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[87] A. Miech, I. Laptev, and J. Sivic, "Learning a text-video embedding from incomplete and heterogeneous data," *arXiv preprint arXiv:1804.02516*, 2018.

[88] A. Karpathy, A. Joulin, and L. F. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[89] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[90] L. Anne Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, "Localizing moments in video with natural language," in *International Conference on Computer Vision (ICCV)*, 2017.

[91] "PySceneDetect," http://scenedetect.com/en/latest/reference/detection-methods/, 2021.

[92] "Natural Language Toolkit: TextTiling," https://www.nltk.org/_modules/nltk/tokenize/texttiling.html, 2022.

[93] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning research*, vol. 3, no. Jan, 2003.

[94] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval (chapter 16)*. Cambridge University Press, 2008.

[95] "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu, 2002.

[96] F. Bianchi, S. Terragni, and D. Hovy, "Pre-training is a hot topic: Contextualized document embeddings improve topic coherence," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021, pp. 759–766.

[97] S. Terragni, E. Fersini, B. G. Galuzzi, P. Tropeano, and A. Candelieri, "Octis: comparing and optimizing topic models is simple!" in *European Chapter of the Association for Computational Linguistics: System Demonstrations*, 2021.

[98] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[99] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation," in *ICML*. PMLR, 2022, pp. 12 888–12 900.

[100] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," in *ICML*, 2023.

[101] J. Li, R. R. Selvaraju, A. D. Gotmare, S. Joty, C. Xiong, and S. Hoi, "Align before Fuse: Vision and Language Representation Learning with Momentum Distillation," in *NeurIPS*, 2021.

[102] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, "Flamingo: a Visual Language Model for Few-Shot Learning," *NeurIPS*, vol. 35, pp. 23 716–23 736, 2022.

[103] J. Xu, X. Zhou, S. Yan, X. Gu, A. Arnab, C. Sun, X. Wang, and C. Schmid, "Pixel Aligned Language Models," *arXiv preprint arXiv: 2312.09237*, 2023.

[104] W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song, J. Xu, B. Xu, J. Li, Y. Dong, M. Ding, and J. Tang, "CogVLM: Visual Expert for Pretrained Language Models," 2023.

[105] Wenliang Dai and Junnan Li and Dongxu Li and Anthony Meng Huat Tiong and Junqi Zhao and Weisheng Wang and Boyang Li and Pascale Fung and Steven Hoi, "Instructblip: Towards general-purpose vision-language models with instruction tuning," 2023.

[106] L. Momeni, M. Caron, A. Nagrani, A. Zisserman, and C. Schmid, "Verbs in action: Improving verb understanding in video-language models," in *ICCV*, 2023, pp. 15 579–15 591.

[107] R. Liu, C. Li, Y. Ge, Y. Shan, T. H. Li, and G. Li, "One for all: Video conversation is feasible without video instruction tuning," *arXiv preprint arXiv:2309.15785*, 2023.

[108] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," *arXiv preprint arXiv:1807.03748*, 2018.

[109] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[110] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[111] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *ICLR*, 2019.

[112] J. Huang, Y. Li, J. Feng, X. Wu, X. Sun, and R. Ji, "Clover: Towards a Unified Video-Language Alignment and Fusion Model," in *CVPR*, 2023, pp. 14 856–14 866.

[113] J. Lei, T. L. Berg, and M. Bansal, "Revealing Single Frame Bias for Video-and-Language Learning," *arXiv preprint arXiv:2206.03428*, 2022.

[114] Q. Ye, G. Xu, M. Yan, H. Xu, Q. Qian, J. Zhang, and F. Huang, "HiTeA: Hierarchical Temporal-Aware Video-Language Pre-training," in *ICCV*, 2023, pp. 15 405–15 416.

[115] D. Li, J. Li, H. Li, J. C. Niebles, and S. C. Hoi, "Align and Prompt: Video-and-Language Pre-Training With Entity Prompts," in *CVPR*, 2022, pp. 4953–4963.

[116] J. Wang, D. Chen, Z. Wu, C. Luo, L. Zhou, Y. Zhao, Y. Xie, C. Liu, Y.-G. Jiang, and L. Yuan, "OmniVL: One Foundation Model for Image-Language and Video-Language Tasks," *NeurIPS*, vol. 35, pp. 5696–5710, 2022.

[117] S. Yan, T. Zhu, Z. Wang, Y. Cao, M. Zhang, S. Ghosh, Y. Wu, and J. Yu, "VideoCoCa: Video-Text Modeling with Zero-Shot Transfer from Contrastive Captioners," *arXiv preprint arXiv:2212.04979*, 2022.

[118] T.-J. Fu, L. Li, Z. Gan, K. Lin, W. Y. Wang, L. Wang, and Z. Liu, "VIOLET: End-to-End Video-Language Transformers with Masked Visual-token Modeling," *arXiv preprint arXiv:2111.12681*, 2021.

[119] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, *et al.*, "Florence: A new foundation model for computer vision," *arXiv preprint arXiv:2111.11432*, 2021.

[120] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra, "Imagebind: One embedding space to bind them all," in *CVPR*, 2023, pp. 15 180–15 190.

[121] K. Li, Y. Wang, Y. Li, Y. Wang, Y. He, L. Wang, and Y. Qiao, "Unmasked teacher: Towards training-efficient video foundation models," *arXiv preprint arXiv:2303.16058*, 2023.

[122] Z. Zeng, Y. Ge, Z. Tong, X. Liu, S.-T. Xia, and Y. Shan, "TVTSv2: Learning Out-of-the-box Spatiotemporal Visual Representations at Scale," *arXiv preprint arXiv:2305.14173*, 2023.

[123] S. Chen, H. Li, Q. Wang, Z. Zhao, M. Sun, X. Zhu, and J. Liu, "Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset," in *NeurIPS*, vol. 36, 2023.

[124] J. Xu, T. Mei, T. Yao, and Y. Rui, "MSR-VTT: A Large Video Description Dataset for Bridging Video and Language," in *CVPR*, 2016, pp. 5288–5296.

[125] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele, "Movie description," *IJCV*, vol. 123, pp. 94–120, 2017.

[126] F. Xiao, K. Kundu, J. Tighe, and D. Modolo, "Hierarchical self-supervised representation learning for movie understanding," in *CVPR*, 2022, pp. 9727–9736.

[127] Z. Khan, C. Jawahar, and M. Tapaswi, "Grounded Video Situation Recognition," *NeurIPS*, vol. 35, pp. 8199–8210, 2022.

[128] Y. Yamada, Y. Tang, and I. Yildirim, “When are Lemons Purple? The Concept Association Bias of CLIP,” *arXiv preprint arXiv:2212.12043*, 2022.

[129] T. Wang, R. Zhang, Z. Lu, F. Zheng, R. Cheng, and P. Luo, “End-to-End Dense Video Captioning with Parallel Decoding,” in *ICCV*, 2021, pp. 6847–6857.

[130] T. Wang, H. Zheng, M. Yu, Q. Tian, and H. Hu, “Event-Centric Hierarchical Representation for Dense Video Captioning,” *IEEE TCSVT*, vol. 31, no. 5, pp. 1890–1900, 2021.

[131] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers,” in *ECCV*. Springer, 2020, pp. 213–229.

[132] R. Mokady, A. Hertz, and A. H. Bermano, “ClipCap: CLIP Prefix for Image Captioning,” *arXiv preprint arXiv:2111.09734*, 2021.

[133] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni, and R. Cucchiara, “From Show to Tell: A Survey on Deep Learning-Based Image Captioning,” *IEEE TPAMI*, vol. 45, no. 1, pp. 539–559, 2022.

[134] H. Liu, C. Li, Y. Li, and Y. J. Lee, “Improved baselines with visual instruction tuning,” in *CVPR*, 2023.

[135] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” in *ECCV*. Springer, 2014, pp. 740–755.

[136] P. Sharma, N. Ding, S. Goodman, and R. Soricut, “Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning,” in *Association of Computational Linguistics (ACL)*, 2018, pp. 2556–2565.

[137] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, “Yfcc100m: The new data in multimedia research,” *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[138] K. Desai, G. Kaul, Z. T. Aysola, and J. Johnson, “RedCaps: Web-curated image-text data created by the people, for the people,” in *NeurIPS*, 2021.

[139] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, *et al.*, “LAION-5B: An open large-scale dataset for training next generation image-text models,” *NeurIPS*, vol. 35, pp. 25 278–25 294, 2022.

[140] A. Fisch, K. Lee, M.-W. Chang, J. H. Clark, and R. Barzilay, “CapWAP: Image Captioning with a Purpose,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 8755–8768.

[141] E. Kreiss, F. Fang, N. Goodman, and C. Potts, “Concadia: Towards Image-Based Text Generation with a Purpose,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2022, pp. 4667–4684.

[142] X. Liu, H. Li, J. Shao, D. Chen, and X. Wang, “Show, Tell and Discriminate: Image Captioning by Self-Retrieval with Partially Labeled Data,” in *ECCV*, 2018.

[143] B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, and Y. Shan, "SEED-Bench: Benchmarking Multimodal LLMs with Generative Comprehension," *arXiv preprint arXiv:2307.16125*, 2023.

[144] K. Li, Y. Wang, Y. He, Y. Li, Y. Wang, Y. Liu, Z. Wang, J. Xu, G. Chen, P. Luo, *et al.*, "MVBench: A Comprehensive Multi-modal Video Understanding Benchmark," *arXiv preprint arXiv:2311.17005*, 2023.

[145] H. Bansal, Y. Bitton, I. Szpektor, K.-W. Chang, and A. Grover, "VideoCon: Robust Video-Language Alignment via Contrast Captions," in *CVPR*, 2023.

[146] T. Thrush, R. Jiang, M. Bartolo, A. Singh, A. Williams, D. Kiela, and C. Ross, "Winoground: Probing vision and language models for visio-linguistic compositionality," in *CVPR*, 2022, pp. 5238–5248.

[147] C.-Y. Hsieh, J. Zhang, Z. Ma, A. Kembhavi, and R. Krishna, "SugarCrepe: Fixing Hackable Benchmarks for Vision-Language Compositionality," in *NeurIPS*, 2023.

[148] A. Ray, F. Radenovic, A. Dubey, B. A. Plummer, R. Krishna, and K. Saenko, "Cola: A Benchmark for Compositional Text-to-image Retrieval," in *Advances in Neural Information Processing Systems (NeurIPS): Track on Datasets and Benchmarks*, 2023.

[149] P. Bagad, M. Tapaswi, and C. G. M. Snoek, "Test of Time: Instilling Video-Language Models with a Sense of Time," in *CVPR*, 2023.