

Towards Video Understanding Through Language in Real-life Settings

by

Santiago Castro

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2024

Doctoral Committee:

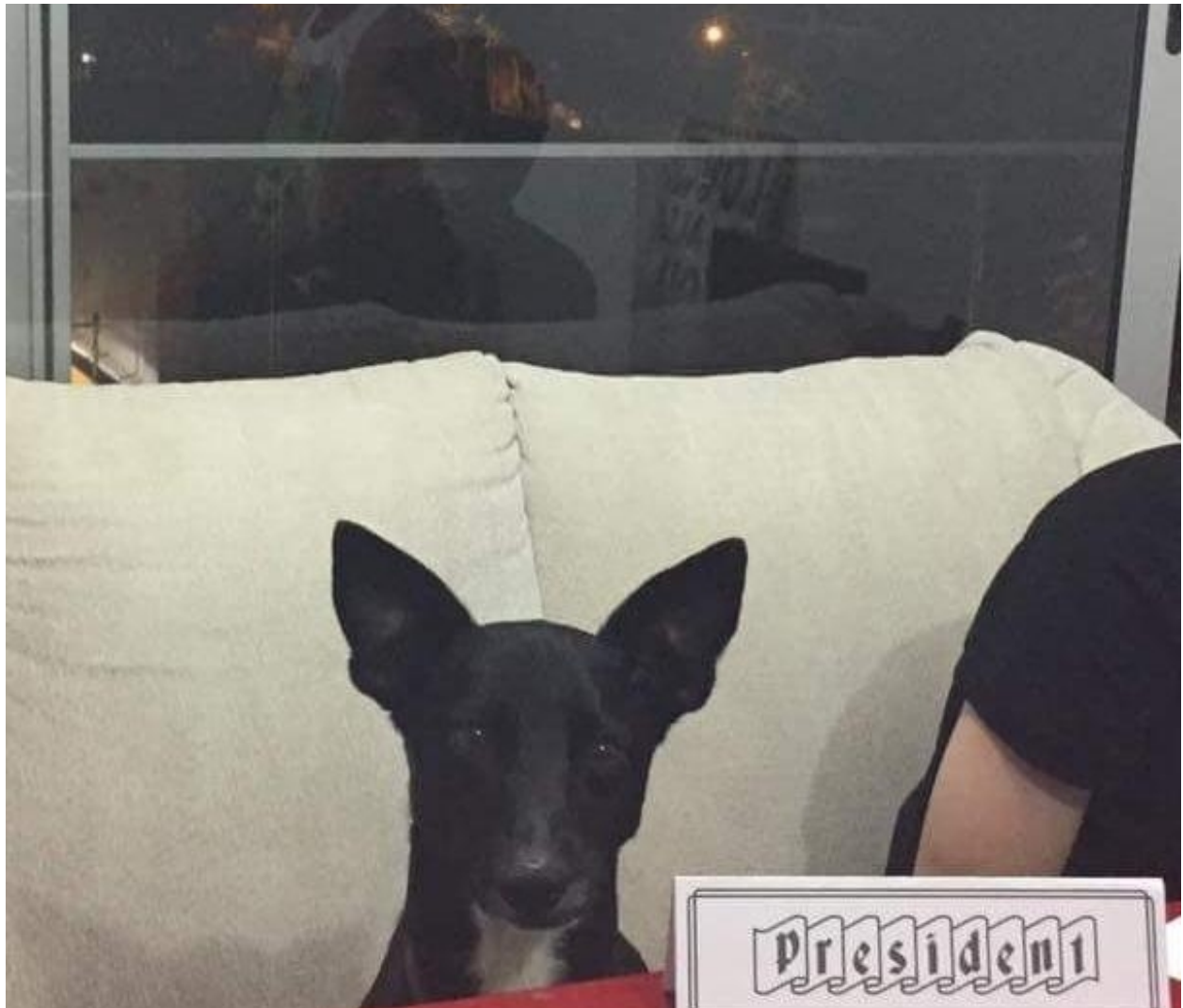
Professor Rada Mihalcea, Chair
Research Scientist Fabian Caba Heilbron, Adobe Research
Professor Joyce Y. Chai
Assistant Professor Justin Johnson
Adjunct Professor Guillermo Moncecchi, Universidad de la
República
Assistant Professor Andrew Owens

Santiago Castro Serra
sacastro@umich.edu
ORCID iD: 0000-0001-8781-9323
© Santiago Castro Serra 2024

DEDICATION

This thesis is dedicated to Zeus Rodríguez, my eleven-year-old dog I left with my mom in Uruguay when I came to the US to pursue a PhD.

You can find him in Google Street View, at the intersection of Gonzalo Ramírez and Santiago de Chile streets, Montevideo, Uruguay: <https://www.google.com/maps/@-34.9124528,-56.1847537,3a,15.5y,132.7h,73.07t/data=!3m6!1e1!3m4!1syUCxQr6aSYJ0RuXBYyBuEQ!2e0!7i13312!8i6656?entry=ttu>.



ACKNOWLEDGMENTS

I guess I should thank Rada first. Thanks for giving me this PhD opportunity, which has led to so many things! My life has been lucky in many ways, but I feel I have been particularly lucky with the advisor lottery since you only truly know an advisor once you work with them for a while. I have found one who is supportive and tolerant. A person with patience even when we move really slowly with projects. If I want to do something different, either in terms of internship, out-of-the-ordinary research ideas, tasks, paper writing, personal decisions, or whatever, she will hardly say no but give her stance and still encourage me to do what we feel like. I admire how you manage the lab, balancing productivity and happiness. I also admire how you really care about diversity. Dear reader, do you want to know if your advisor/mentor/boss/leader is cool? Ask yourself: did she ever give you a birthday cake? Oh, and thanks for the free food every Tuesday.

Thanks to my thesis committee: Rada Mihalcea, Andrew Owens, Joyce Chai, Guillermo Moncecchi, Justin Johnson, and Fabian Caba Heilbron. Their support and suggestions have been valuable for this dissertation.

Thanks to my parents for raising me and giving me opportunities, even when they are both crazy people (whose parents are not?). Thanks for teaching me independence and a great sense of curiosity.

Thanks to Oana for her constant support. Life in the US by her side has been amazing. We have shared (and will continue sharing!) lots of experiences and adventures together. I feel she made me step up my growth, both professionally and in life.

Thanks to all my family and friends. Life would be pointless without them.

Thanks to the people from my internships who gave me opportunities and believed in me. Thanks to Vinith Misra, Boris Chen, Amir Ziai, Avneesh Saluja, Fabian Caba, Ruben Villegas, Mohammad Babaeizadeh, and Zhuoning Yuan. The experiences in each of them were unique, even when they were remote most of the time.

Thanks to the NLP Group in Uruguay (Grupo PLN). They opened many doors for me when I was new to this field. It all started when I took an NLP class in 2013, and I loved it. It inspired me to do my undergraduate thesis on recognizing humor, and they received my idea with open arms. They welcomed me to the group and gave me countless opportunities

to work on research projects in multiple roles. They strove to secure funding to attend multiple venues, especially when I got papers in. I still remember using five or six different funds from them to attend ACL 2018 in Melbourne, and it was worth it! Since I started my PhD degree in 2018, they have kept me in the loop for projects and opportunities. I hope to continue still having an active role now that I am finishing my PhD.

Thanks to the people at Xmartlabs, a place full of highly motivated and talented folks. I feel they believed in me from day one and empowered me to explore many directions. Some ideas turned out to be great, others not so much. But they always listened to me when steering the direction. I learned a lot during my time there! It is hard to put into words all the great values that live in Xmartlabs daily. Values that I feel are hard to find elsewhere. That is what makes this place unique. I have been in contact with them since I finished my formal labor relationship, and I hope to continue in contact since I have good friends there.

Thanks to all my paper collaborators. Thanks to Aurelia Bunescu, Daniel D’Souza, Penghao He, Shubham Dash, and Yu-Wei Chao for helping to collect and annotate the LifeQA dataset. Thanks to William McNamee for the help with the WILDQA video collection process and all the annotators for their hard work. Thanks to Gautam Naik for his help in curating part of the MUsTARD dataset from online resources. Thanks to Laura Biester for helping with data quality assurance for FIBER. Thanks to Muhammad Khalifa, Oana Ignat, Andrew Lee, Artem Abzaliev, Mohamed El Banani, Karan Desai, Fabian Caba Heilbron, Ruben Villegas, Michalis Papakostas, Honglak Lee, Yiqun Yao, and the whole LIT Lab for many productive discussions. Thanks to Pablo Delgado and Netflix’s training platform team for their invaluable help with using Netflix’s computational resources. Thanks to Christine Feak, Oana Ignat, Artem Abzaliev, Do June Min, Victoria Florence, Zhijing Jin, and Max Krogius for proofreading and suggestions for many of the papers related to this document. Thanks to the anonymous reviewers for their constructive feedback on the papers related to this thesis (but not you, Reviewer 2).

If you are reading this and gave me opportunities or believed in me and do not fall into any of the above categories, then I forgot to mention you, and I am sorry. Thank you!

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGMENTS	iii
LIST OF FIGURES	viii
LIST OF TABLES	x
LIST OF APPENDICES	xii
ABSTRACT	xiii
CHAPTER	
1 Introduction	1
1.1 Funding Acknowledgments	4
2 Understanding Daily Life Situations in Videos	6
2.1 Introduction	6
2.2 Related Work	7
2.2.1 Text-based Question Answering	7
2.2.2 Multimodal Question Answering	8
2.2.3 In-the-Wild Datasets	8
2.3 LifeQA Dataset	9
2.3.1 Dataset Collection	9
2.3.2 Dataset Analysis	10
2.4 Experiments	13
2.4.1 Baselines	13
2.5 Results	15
2.6 Conclusion	17
3 Video Understanding in In-The-Wild Scenarios	18
3.1 Introduction	18
3.2 Related Work	20
3.3 WildQA Dataset	21
3.4 Video Question Answering	23
3.4.1 Baselines	24

3.4.2	Results	27
3.5	Video Evidence Selection	27
3.5.1	Baselines	27
3.5.2	Results	28
3.5.3	Analysis and Discussion	29
3.6	Conclusion	32
4	Addressing Real-Life Human Behavior	34
4.1	Introduction	34
4.2	Dataset	35
4.2.1	Annotation Process	35
4.2.2	Transcriptions	37
4.3	Multimodal Feature Extraction	37
4.3.1	Text Features	37
4.3.2	Video Features	37
4.4	Experiments	38
4.4.1	Experimental Setup	38
4.4.2	Baselines	38
4.5	Multimodal Sarcasm Classification	38
4.5.1	The Role of Context and Speaker Information	40
4.6	Conclusions	41
5	Realistic and Robust Video Understanding Evaluation	43
5.1	Introduction	43
5.2	Related Work	44
5.3	Video Fill-in-the-Blanks Dataset	46
5.3.1	Data Generation	46
5.3.2	Data Annotation	47
5.3.3	Data Analysis	49
5.3.4	Human Agreement	50
5.3.5	Limitations	52
5.4	Multimodal Method for Video Fill-in-the-Blanks	53
5.4.1	Implementation Details	55
5.4.2	Baselines	55
5.5	Experiments and Results	56
5.5.1	Evaluation Metrics	56
5.5.2	Results	56
5.5.3	Error Analysis	57
5.6	Conclusions	59
6	Practical and Scalable Video Understanding	60
6.1	Introduction	60
6.2	Related Work	61
6.3	Method: FitCLIP	63
6.3.1	Teacher-Student Fine-tuning	63

6.3.2	Fusing Teacher-Student Knowledge	65
6.3.3	FitCLIP’s Implementation Details	66
6.4	Zero-shot Video Understanding Benchmark	66
6.4.1	Baselines	66
6.4.2	Zero-shot Tasks and Datasets	67
6.5	Experimental Results	68
6.5.1	Zero-shot Action Recognition Results	69
6.5.2	Zero-shot Text-to-video Retrieval	70
6.5.3	Diagnostic Analysis	71
6.6	Conclusions	72
7	Compositional Generalization with Image-Text Models	73
7.1	Introduction	73
7.2	Related Work	75
7.3	Understanding the Limitations of Vision-Language Models: a Case Study on CLIP	77
7.3.1	Methodology	78
7.3.2	Results	83
7.4	CLOVE: A Framework to Increase Compositionality in Contrastive VLMs	92
7.4.1	Synthetic Captions	93
7.4.2	Hard Negatives	93
7.4.3	Model Patching	94
7.5	Case Study on CLIP	94
7.5.1	Experimental Setup	95
7.5.2	Using CLOVE to Bring Compositionality into CLIP	96
7.5.3	Generalization to Unseen Verb-Object Compositions	98
7.5.4	Ablation Studies	99
7.6	Conclusions	102
8	Conclusions	104
8.1	Research Questions Revisited	104
8.2	Future Directions	108
8.2.1	Other Common Realistic Human Behavior	108
8.2.2	Understanding Novel Actions Compositions Through Motion	108
8.2.3	Extending the Fill-in-the-Blank Framework to Other Tasks	109
8.2.4	Fixing Compositionality Generalization Root Cause with Inductive Biases	109
	APPENDICES	110
	BIBLIOGRAPHY	151

LIST OF FIGURES

FIGURE

2.1	An example from LifeQA.	7
2.2	Distribution of the LifeQA questions’ tokens.	11
2.3	Venn diagram at scale showing the number of questions by answer type.	11
2.4	Distribution of the LifeQA questions by type.	12
2.5	Additional instances from LifeQA.	13
3.1	An example from our WildQA dataset.	19
3.2	Examples from MovieQA, TVQA, and our WildQA dataset.	19
3.3	The two phases of data annotation.	22
3.4	Percentage distribution of question types.	24
3.5	Examples of questions and answers from WildQA.	25
3.6	Multi _{T+V,SE} performance for Video QA when tuned on a single domain and tested against each domain.	31
3.7	Multi _{T+V,SE} performance for Video Evidence Selection when tuned on a single domain and tested against each domain.	32
4.1	Graphical user interface used by the annotators to label the videos in our dataset.	36
4.2	Correlation in speaker-specific sarcastic tendencies of the top-7 speakers.	42
5.1	Annotation interface.	48
5.2	The 2D t-SNE representation of the clustering of the top 100 most frequent answers provided for the blanks.	51
5.3	Answer agreement per caption grouped by the number of annotations of its caption.	52
5.4	Multimodal models for video fill-in-the-blanks.	54
6.1	FitCLIP refinement strategy and model.	64
7.1	Example of compositions recognized by CLOVE but not by CLIP.	75
7.2	ImageNet vs. SugarCrepe performance for CLOVE and baselines.	76
7.3	CLIP analysis framework.	79
7.4	CLIP score distribution for positives and negatives.	84
7.5	Average word concreteness vs. CLIP score.	85
7.6	Number of words in the sentence vs. the CLIP score (line plot).	86
7.7	Number of words in the sentence vs. the CLIP score (box plot).	86
7.8	Average word frequency in the sentence vs. CLIP score.	87
7.9	Average word synset count vs. CLIP score.	88

7.10	The CLIP score distribution of the similarity between the image and positive vs. the negative text captions.	89
7.11	The CLIP score distribution of the similarity between the image and the replaced vs. the new word.	90
7.12	CLOVE framework diagram.	92
7.13	The effect of model patching.	101
A.1	Interface for annotation Phase 1.	117
A.2	Interface for annotation Phase 2.	118
A.3	Distribution of questions by the first four tokens.	122
A.4	Venn diagrams showing whether the question depends on visual or audio from the original video.	123
A.5	Multi-Task ROUGE-2 scores for Video QA when tuned on a single domain and tested against each domain.	126
A.6	Multi-Task ROUGE-L scores for Video QA when tuned on a single domain and tested against each domain.	127
A.7	Multi _{T+V,SE} performance on different question types for Video QA.	128
B.1	Top 20 nouns for the originally blanked phrases and the annotations in the validation and test data.	130
B.2	Relative frequency of part-of-speech tags in the originally blanked phrases and the annotated answers.	131
B.3	Relative frequency of POS tag sequences in the originally blanked phrases and the annotated answers.	132
B.4	Relative frequency of dependency types for the root token of the original blanked phrases.	133
B.5	Average number of unique answers per caption, grouped by the dependency type of the root word of the originally blanked phrases.	134
B.6	Heat map showing how frequently the blanked entity appears within a given location of the video.	136
B.7	Frequency that the blanked entity appears at each one-second interval in a given video.	136
B.8	Distribution of the total time that each blanked entity is seen within its video.	137
C.1	FitCLIP vs. Teacher per-class improvements.	141
C.2	FitCLIP vs. CLIP distribution of Text-to-Video Retrieval rankings.	142
C.3	Impact of changing the value of weight-ensembling α value when fusing the teacher and the student.	146
C.4	Text-to-Video top-1 recall on WebVid-2.5M of different training subset sizes when fine-tuning CLIP and then applying weight-space ensembling.	147
C.5	The effect on the zero-shot performance of the share of the pseudo-labeled and labeled losses in FitCLIP.	148

LIST OF TABLES

TABLE

2.1	Statistics of the LifeQA dataset.	10
2.2	Video and Dialog QA datasets comparison.	12
2.3	Baselines on the LifeQA dataset.	16
3.1	Video and question count for each domain.	22
3.2	Dataset statistics for WildQA.	23
3.3	Comparison between our WildQA and other existing datasets.	23
3.4	ROUGE scores for the task of Video Question Answering.	26
3.5	IOU-F1 scores for Video Evidence Selection.	28
3.6	Multi _{T+V,SE} performance on different question types for Video QA and for Video Evidence Selection.	29
3.7	ROUGE scores for the task of Video Question Answering for few-shot learning setting.	30
4.1	Speaker-dependent setup.	39
4.2	Speaker-independent setup.	40
4.3	Role of context and utterance’s speaker.	41
5.1	Three examples from the FIBER dataset.	44
5.2	Summary statistics for the originally blanked phrases and the annotated answers.	49
5.3	Agreement statistics for the answers.	50
5.4	Results on the validation set.	57
5.5	F1 scores on the validation set for blanks with different semantic categories.	58
6.1	Zero-shot action recognition results.	69
6.2	Zero-shot text-to-video retrieval results.	70
6.3	Impact of fusing teacher-student knowledge.	72
7.1	CLIP relative performance analysis on a subset of binary features.	91
7.2	Zero-shot compositional evaluation results.	95
7.3	Zero-shot classification results.	95
7.4	Zero-shot retrieval results.	97
7.5	Results on RareAct.	98
7.6	Compositionality performance changes when fine-tuning CLIP with different datasets.	99
7.7	Compositional performance when employing negatives.	100

A.1	Average scores of the pilot study for Phase 1.	119
A.2	Examples in pilot study for Phase 1.	119
A.3	ROUGE and IOU-F1 scores for the pilot study in Phase 2.	120
A.4	Information about the expert annotators who annotate the questions.	121
A.5	Three most common words for each domain after removing stop-words.	122
A.6	Annotation statistics for Phase 1.	124
A.7	Annotation statistics for Phase 2.	124
A.8	Multi-task parameter selection results for the Evidence Selection SE method. . .	125
A.9	Multi-task parameter selection results for the Evidence Selection IO method. . .	125
A.10	Ablation study on the Video Evidence Selection.	126
B.1	F1 scores on the validation set for the beam sizes 1 (greedy search), 2, 4, and 8. .	138
B.2	Results on the validation set for different model sizes of the T5 text-only zero-shot model.	138
B.3	Examples of instances correctly predicted by the best multimodal method but incorrectly predicted by the best text-only method.	139
C.1	Pretraining and zero-shot datasets.	141
C.2	Zero-shot action recognition results of Frozen in Time pre-trained on different datasets.	143
C.3	Zero-shot text-to-video retrieval results of Frozen in Time pre-trained on different datasets.	144
C.4	Impact of fusing teacher-student knowledge on zero-shot action recognition. . .	144
C.5	Impact of fusing teacher-student knowledge on zero-shot text-to-video retrieval. .	145
C.6	Importance of the Pseudo-Labels.	148
D.1	Results on SugarCrepe.	149
D.2	Zero-shot classification results without employing text prompts.	150
D.3	Retrieval results for Flickr30k and COCO Captions.	150

LIST OF APPENDICES

A Video Understanding in In-The-Wild Scenarios: Supplementary Material	110
Annotation Details	110
Annotation Statistics	124
Details of Multi-task Learning	124
Experiment Results	124
B Realistic and Robust Video Understanding Evaluation: Supplementary Material	129
Dataset	129
Experiments and Results	135
C Practical and Scalable Video Understanding with a Single Model: Supplementary Material	140
Pretraining Datasets	140
FitCLIP vs. CLIP per-class performance	140
FitCLIP vs. CLIP ranking distributions	142
Frozen in Time Variants	142
Impact of Fusing the Teacher-Student Knowledge	143
Alpha Value	143
Impact of the Labeled Data Size	146
Share of Pseudo-Labels/Labels	147
D Compositional Generalization with Image-Text Models: Supplementary Material	149
SugarCrepe Fine-Grained Performance	149
Classification without Prompts	149
Performance in Flickr and COCO Retrieval Tasks	150

ABSTRACT

Videos have become an integral part of our daily lives, with a rapidly growing number on YouTube, Netflix, and TikTok serving as testimony to their widespread popularity. Behind the simplicity of their interfaces and user experiences, the systems that power these products employ numerous video-understanding techniques, even for straightforward use cases such as finding a video on how to cook salmon. Despite the significant progress achieved in this area, there remains a gap between lab-setting capabilities and reality, as multiple phenomena are not adequately designed for realistic settings, causing various issues such as domain mismatches and the diverse way people interact in videos (e.g., sarcastically). My work aims to bridge this gap by enabling the understanding of video content in realistic settings.

The issues that make current video understanding research unsuitable for real life can be classified into data, methods, and evaluation. The data aspect is crucial since current research has predominantly overlooked real-life settings. I present new datasets and benchmarks for such domains: daily situations and in-the-wild scenarios. These benchmarks measure the effectiveness of new methods in these more realistic settings. Likewise, I introduce a novel framework that accounts for a typical yet understudied human behavior: sarcasm. Sarcasm is particularly suited to be studied in video since I show that leveraging what we see and hear (as people commonly do) allows one to understand it better. For the methods aspect, I consider a fundamental issue, which is the impracticality and lack of scalability of the traditional in-the-lab setting, tuning one model for each newly addressed task and domain. I propose a robust method that allows practitioners to employ a single model for novel tasks and domains with satisfactory performance. Additionally, I present a technique to improve the compositional generalization of existing models. Finally, I focus on current practices for evaluation and propose a framework better suited to realistic settings. Current benchmarks for short video understanding have drawbacks, such as employing easy-to-detect distractor answers, not accounting for diversity when depicting the same situation, and not considering realistic settings. I present a novel evaluation format that tackles all these issues and a benchmark that leverages it. The benchmark shows a gap between the performance of several methods and humans.

CHAPTER 1

Introduction

Over the last decade, we have witnessed an increasing demand for multimedia content worldwide. We watch movies on Netflix, learn from YouTube videos, and surface diverse content on TikTok and Instagram. We use devices at home that can listen or even look at what happens at home.

Hidden from plain sight, plenty of video understanding methods help us achieve our goals in this context, such as when we want to find a video to learn how to cook a salmon [158, 280, 157]. To make things even more complicated, 500 hours of video are uploaded to YouTube every minute.¹ New possibilities are routinely achieved by researchers and engineers, such as making video editors avoid wasting several hours finding the video footage they need. Likewise, we expect home devices and soon-to-arrive assistant robots to provide quick answers based on what they perceive.

Similarly to how humans generally combine information from multiple sources (e.g., we look and listen) – humans are not unimodal; the fields of Natural Language Processing and Computer Vision have been flooded by works that borrow ideas from each other, which consider both vision and language (more broadly, that are multimodal) [60, 235, 223, 33, 46, 183]. Several methods have been proposed in both fields to tackle different video understanding problems [158, 157, 239, 149]. However, state-of-the-art video understanding methods are typically effective only under ideal in-the-lab conditions and fail in realistic use cases. There are many aspects along the video understanding pipeline that fail when faced with real-life settings. These aspects can be classified into Data, Methods, and Evaluation.

Data. For this aspect, the domain of the data is critical. If we want to deploy robots that assist people at home and school, we need systems that understand the situations they “see”. However, current datasets and systems focus on domains such as movies [225, 112, 127, 128, 91] and arbitrary user-generated content [251, 79, 2], or consider only brief activities such as recognizing when somebody is drinking coffee [79, 108]. Likewise, we lack systems that can

¹<https://statista.com/statistics/259477/>

be deployed to study natural environments, as nature has been an understudied domain. Another overlooked part of the data is how people act. We naturally express ourselves in multiple ways, such as being deceptive, humorous, or sarcastic. If a person answers a system at home sarcastically with “I understood everything”, the system should not act upon it as if it were literal.

Methods. The methods researchers devise in laboratory settings are typically assumed to perform on real-world applications as they do on standard benchmarks. Drops in performance are expected if such methods are not tuned to the new task and domain at hand, even when these methods take advantage of previously learned information [109]. In addition to this reason to tune the existing methods to new tasks, practitioners regularly find that the methods they rely on were initially adapted to a particular task format that they now need to change, such as a method capable of recognizing an action out of a fixed set [120, 211, 79, 108]. These issues build up an expectation to produce different models for every task and domain. While this approach may provide excellent performance in some cases, it is not practical and does not scale. A better approach would be to leverage the similarities between the different domains and tasks to build more general systems.

Evaluation. How can we evaluate such systems? Suppose we ask “where did I leave my keys?” A multiple-choice approach [225, 127, 99, 128] would be inappropriate as users would not want to provide options. Evaluating a free-form text-based answer, while ideal in the format, is still an open research question when accounting for the diversity a correct answer could have. In particular, Video Captioning benchmarks typically show a low human agreement, even when employing several annotators [251, 241].

The research community should work more on multiple aspects of video understanding in realistic settings to build systems that can work for videos in real-life use cases. This dissertation aims to produce datasets, benchmarks, and methods that leverage language to achieve a more realistic understanding of videos. Concretely, I look to answer the following research questions:

1. Can we build a language-based video understanding benchmark for overlooked real-life domains, such as daily situations and in-the-wild scenarios?

Previous work has worked on professionally-edited videos with crisp audio from movies and TV series [225, 127, 112, 91], considered only atomic actions [120, 211, 79, 108], or focused only on people cooking [280, 42]. The video understanding literature has greatly overlooked real-life settings, which typically involve multiple interactions and understanding a context and what happens in nature, such as during natural disasters. Chapters 2 and 3 presents benchmarks that consider these situations for long videos.

This work contributes to the aspects of **data**, **methods**, and **evaluation**.

2. Does combining vision and language help better recognize naturally occurring human behaviors in videos?

People’s daily interactions involve phenomena consistently omitted in the video understanding literature, such as deception, humor, and sarcasm. Sarcasm, in turn, has almost uniquely been studied from an unimodal perspective in Computer Science, especially only in text. This issue worsens when considering that text-only sarcasm is hard to grasp even for humans [185]. I encourage the reader to think about how often they observed somebody not getting the sarcasm on Twitter. In Chapter 4, I explore the study of sarcasm by combining vision and speech (with both textual and non-textual features). This work brings value to **data**, **methods**, and **evaluation**.

3. Can language be leveraged to build an automatic evaluation framework for video understanding that better reflects real-life situations?

Many video understanding evaluation frameworks are based on multiple-choice answers, making models select the single correct answer [225, 127, 99]. However, the multiple-choice format suffers from models only learning to identify the distractors [96] and is unrealistic – we cannot pretend people will provide AI systems such as Alexa or Siri with answer choices. Video captioning, while flexible because it considers free-form textual answers, inaccurately represents the diverse way a person can describe the content of a video, mainly because of the available noisy metrics (e.g., BLEU [170] and ROUGE [140]) to compare a predicted answer with a set of reference answers. In Chapter 5, I present a novel evaluation framework, which accounts for the diverse ways a person can describe the content of a ten-second video while still being challenging and making models generate answers (as opposed to making choices). Here, my contributions are to **data**, **methods**, and **evaluation**.

4. Can large pre-trained image-text alignment models be used for robust zero-shot video understanding?

Having a single model for an arbitrary number of tasks is appealing to many production use cases instead of having to train and deploy a model for each task and domain. Language is a great way to use a single model for tasks that support different formats [46, 183, 100], by employing what is known as zero-shot learning (training a model on a specific task and domain, then evaluating it on any other ones). For zero-shot image tasks, large pre-trained image-text models have shown remarkable performance [183, 100]. Practitioners could replicate these same ideas for video tasks to achieve

similar results. However, gathering a (weakly) labeled video dataset in the order of the million videos is still an open research problem [164], and training models on such a dataset would only be possible for the big industry players. In Chapter 6, I show a method to leverage these large pre-trained image-text models for zero-shot video tasks by leaning on language. This study contributes mainly to **methods**.

5. Can we align vision and language models so that they better generalize to unseen verb-object compositions?

Large pre-trained image-text models are practical because a single model can be used for multiple tasks without further training. However, evidence shows that such models cannot generalize well to unseen compositions, such as people playing basketball on the grass with ordinary clothes on [228, 82, 28]. In Chapter 7, I propose a method to improve the zero-shot performance of such models on unseen verb-object compositions while maintaining the general visual-text alignment on standard classification tasks. With this research question, I bring contributions to **methods**.

1.1 Funding Acknowledgments

The following organizations have partially supported my research during my academic years:

- The Michigan Institute for Data Science.
- The National Science Foundation (grant #1815291).
- The John Templeton Foundation (grant #61156).
- DARPA (grant #HR001117S0026-AIDA-FP-045).
- Toyota Research Institute (TRI).
- Automotive Research Center (ARC).
- Adobe Research.
- Netflix.

The following organizations have supported my research during internships:

- Netflix
- Adobe Research

- Google Brain

Any opinions, findings, conclusions, or recommendations expressed in this material are mine and do not necessarily reflect the views of the previous entities or any related ones.

CHAPTER 2

Understanding Daily Life Situations in Videos

2.1 Introduction

Video Question Answering (Video QA) is one of artificial intelligence’s most challenging and crucial problems. In this task, we are given a video and must answer natural language questions about its content, such as “What game is the little girl playing?”. Answering these questions requires a rich understanding of the video’s visual and auditory content and the ability to relate this content to natural language concepts. Like many challenging tasks, much of the recent progress on Video QA is due to the introduction of several large-scale datasets, which consist primarily of movies and TV shows [225, 195, 127]. Movies and TV shows provide countless hours of clean, crisply-edited video and accurately captioned audio and are, therefore, easily adapted into datasets. However, these same features mean that movies and TV are not representative of day-to-day life. Thus, these datasets cannot be used to evaluate how well models perform when applied to realistic videos of day-to-day life.

To address this issue, we introduce **Life Question Answering (LifeQA)**, a Video QA benchmark dataset that consists of videos and questions about day-to-day life. LifeQA is drawn from hand-picked YouTube videos, which depict scenarios such as children playing, a family having a meal together, or a snapshot from a daycare. These videos are not professionally shot, edited, or scripted, making them much more representative of daily life than prior datasets. They also benefit from increased diversity regarding the number of people and scenes that appear since they are not drawn for a small set of shows or films. In addition, the questions include few proper names or references to known locations, which are commonly referenced in TV datasets that feature well-known characters (such as “Sheldon”, or “Monica’s apartment”), and therefore, the questions have to be answered without prior knowledge about the scene. Moreover, the questions are challenging as they cover visual grounding (“What color is the blanket?”), intent (“What does the father want to do with the box?”), and commonsense reasoning (“What is in the bottle?”), all hallmarks of a comprehensive QA

Caitlin, are you gonna be a little helper on this challenge?

What is the name of the younger girl?	How many people are playing?
A. Caitlin	A. 2
B. Lucy	B. 4
C. Jane	C. 3
D. Cindy	D. 1

Figure 2.1: An example from LifeQA. The image shows a frame from the video, part of the transcriptions, two questions, the candidate answers, and the correct answers in bold.

dataset.

LifeQA consists of 275 videos and 2,326 multiple-choice questions, making it a suitable complement for existing datasets and a challenging benchmark for existing Video QA systems. To enable future research, we are making LifeQA publicly available, along with automatically and manually generated transcriptions (from the speech in the audio channel) and pre-computed features for every video. In this chapter, we describe the LifeQA dataset, present several analyses, and evaluate the performance of several baselines highlighting the task’s difficulty.

2.2 Related Work

2.2.1 Text-based Question Answering

Question answering based on text has been extensively explored [194, 84, 243]. Early question-answering systems were developed for restricted domains, relied on manually crafted features, and had limited capabilities [107, 212, 13]. Recently, the rise of deep learning methods motivated the need for large question-answering datasets to leverage the capabilities of such models. With that goal in mind, several large-scale reading comprehension datasets were introduced [187, 194, 12, 166]. [187] introduced the SQuAD dataset, which is composed of Wikipedia articles. The answers are specified as spans from a text passage. Similarly, [194]

collected the MCTest dataset, a multiple-choice open-domain reading comprehension dataset. Given a paragraph, a question, and a set of multiple answers, the task of a QA system is to select the correct answer.

2.2.2 Multimodal Question Answering

Recently, question-answering systems have been constructed to answer questions about other modalities, such as images (Visual QA) and video (Video QA). For the former, several datasets have been proposed, such as VQA [4], Visual7W [283], VisDial [43], GQA [92], and DREAM [221]. These benchmarks aim to help build visual understanding systems that can reason about the contents of a given image. Given an image and a question, the system selects a correct answer from multiple choices or generates a free-form textual answer.

Video QA is more challenging because it allows for a broader range of question types and requires temporal information. Many datasets have been proposed for Video QA, such as LSMDC 16 [195], TGIF-QA [99], MovieQA [225], PororoQA [112], MarioQA [163], VCQA [281], TVQA [127], and ActivityNet-QA [263]. LSMDC, TGIF-QA, PororoQA, and MarioQA consist of short video clips (just a few seconds), which makes it difficult to understand what is going on in a scene beyond several actions that can be identified. Additionally, they depend entirely on visual cues, with no presence of speech and other audio cues.

MovieQA and TVQA consist of movies and TV series. The questions and answers were generated based on the dialog and visual information presented in short video clips from TV shows. However, these acted and well-directed video clips are hard to find in the real world. As them, we constructed our questions and answers based on textual and visual cues from short video clips. However, unlike them, our proposed dataset relies on video clips recorded naturally by people without predefined scripts. Therefore, understanding videos requires overcoming environmental noise, camera movements, lighting conditions, and naturally occurring dialogues, among other challenges. In addition, scenes are less defined, with undefined characters, lack of subject permanence, and sometimes incoherent conversations. That makes our dataset more challenging for Visual QA tasks.

2.2.3 In-the-Wild Datasets

Recent work in computer vision has focused on evaluating models “in the wild” – that is, on realistic datasets that depict real-life situations. This phenomenon is evident in recent video datasets, such as Charades [208] and VLOG [58], which include indoor scenes of human activities. These datasets include rich annotations about human actions, objects, and scenes

but do not include questions and answers as in LifeQA. To our knowledge, our LifeQA dataset is the first real-life dataset for Video QA.

ActivityNet-QA comprises short YouTube clips initially selected for an activity recognition dataset [79]. Unlike our dataset, these datasets do not explicitly include videos of real-life settings.

VCQA [281] consists of cooking and in-the-wild YouTube videos (about half of the dataset) and clips from movies (the other half). Questions in VCQA are automatically generated from templates and are not written by humans. Additionally, these automatically generated questions only focus on nouns and verbs and short-term temporal reasoning questions. At the same time, LifeQA has more challenging questions about reasons, emotions, and locations. Moreover, VCQA does not consider dialogues, texts, and audio information equally crucial to understanding real-life scenes.

2.3 LifeQA Dataset

2.3.1 Dataset Collection

To collect this dataset, we begin by searching for videos on YouTube, using manually chosen keywords that lead to videos of people living out their daily lives in varied settings (e.g., “my morning routine,” “dialogue,” “kids playing,” “class in elementary school” and “watching TV”). We then hand-pick 59 such videos showcasing recordings of natural interactions in natural settings. We explicitly exclude videos that do not include language interactions.

Identifying such videos turns out to be challenging, requiring significant manual effort. This issue occurs primarily because most of the recordings available online are vlogs, which include video recordings with voice layovers and are, therefore, not typical of natural interactions.

We manually split the source videos into 275 video clips so that each clip includes coherent scenes and lasts 1–2 minutes. We obtain transcriptions for the video clips using the Google Cloud Speech-to-Text platform. We also collect manual transcriptions for each video.

Next, two annotators write five questions per video. For each question, we ask the annotators to write the correct answer and three distractors (which we define as incorrect but semantically related answers). We instruct the annotators to formulate diverse questions that require understanding the video’s visual and linguistic content. We then instruct a third annotator to merge the two sets of questions from the original annotators, manually eliminate duplicate questions, and correct typographical errors. Using this procedure, we collected 2,326 questions in total.

We present a dataset summary in Table 2.1. Figure 2.1 shows an example from the LifeQA

Source videos	59
Clips	275
Clips per source video	4.7 ± 3.6
Clip duration	$1\text{m } 14\text{s} \pm 16\text{s}$
Modalities	video, audio, text
<hr/>	
Questions	2326
Questions per clip	8.5 ± 2.0
<hr/>	
Candidate answers	4
<hr/>	
Tokens per question	6.7 ± 2.1
Tokens per correct answer	1.5 ± 1.1
Tokens per incorrect answer	1.4 ± 0.9

Table 2.1: Statistics of the LifeQA dataset. Here, we report totals and averages along with standard deviation.

dataset, showing two sample questions that require either linguistic or visual clues to be answered. Additional questions are illustrated in Fig. 2.5.

2.3.2 Dataset Analysis

We examine LifeQA’s common question types in Fig. 2.2. Most of the questions are “what” questions, previously acknowledged as among the most frequent and ambiguous types of questions. We find that “what” questions most frequently reference “color”, “number”, and “kind”, each requiring visual clues from the video. Not pictured in Fig. 2.2: we find that nouns referring to people, such as “girl”, “woman”, “man”, and “boy” are the first nouns in more than 21% of the questions, and we find very few proper names.

We then analyze the data type required to answer the questions, as shown in Fig. 2.3. To obtain these results, we manually inspect each question and answer to determine whether the question requires the visual (video) or speech (audio or transcription) modalities to answer. We find that 61% of questions need the video to be answered, 29% require speech or audio information, and 10% need both modalities.

In addition, we analyze the questions based on the expected answer types, as shown in Fig. 2.4. [225, 127] inspire this analysis to understand better the information needed to answer each question. The graph shows that many questions reference basic visual features, such as count (how many), color (what color), and location (where) answers. However, many questions require both language and visual features. For example, abstract (“what”) questions (“What is the job of the woman?”) can require more than one mode of information to answer.



Figure 2.2: Distribution of the LifeQA questions' tokens.

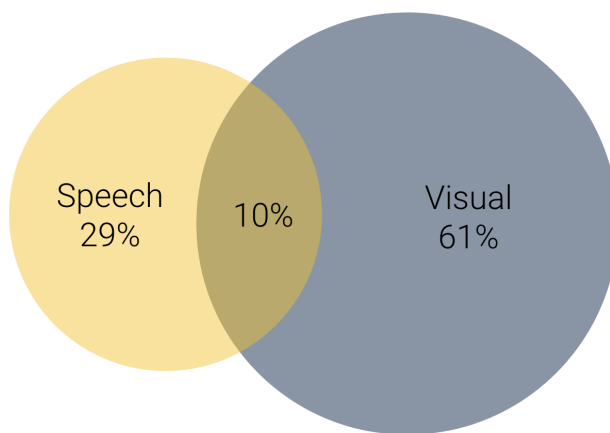


Figure 2.3: Venn diagram at scale showing the number of questions by answer type.

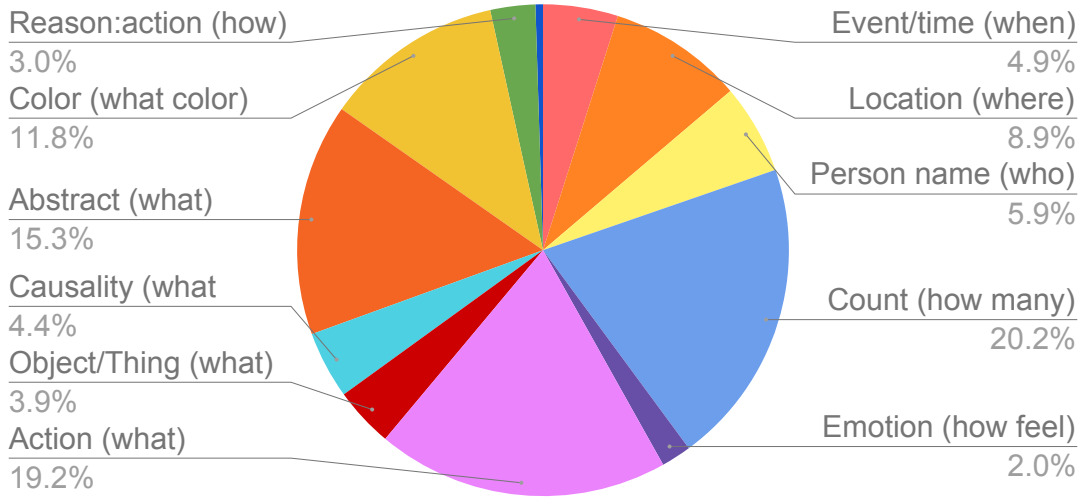


Figure 2.4: Distribution of the LifeQA questions by type.

Dataset	Task	Source	Answer	Questions	Samples	Avg dur. (s)	Real life?	D	T	A	I	V
DREAM	Reading Comprehension	Exams	MC	10,197	6,444	-		✓	✓			
VisDial	Dialog QA	Images	MC	1,261,510	133,351	-		✓	✓		✓	✓
LSMDC 16	Video Description	Movies	Text	128,118	128,085	4.1				✓	✓	✓
TGIF-QA	Temporal Reasoning	Tumblr	MC/Txt	165,165	71,741	~3.6					✓	✓
MovieQA	Story Underst.	Movies	MC	14,944	6,771	202.7		✓	✓	✓	✓	✓
PororoQA	Story Underst.	Cartoons	MC	8,913	16,066	4.6		✓	✓	✓	✓	✓
MarioQA	Temporal Reasoning	Games	MC	187,757	187,757	4.5					✓	✓
TVQA	Story Underst.	TV Series	MC	152,545	21,793	76.2		✓	✓	✓	✓	✓
VCQA	Temporal Reasoning	Movies/Web	FB/MC	390,744	109,895	~30.0					✓	✓
LifeQA	Real-life Underst.	YouTube	MC	2,326	275	74.0	✓	✓	✓	✓	✓	✓

Table 2.2: Video and Dialog QA datasets comparison. Answer = answer type, h = hours of video, s = seconds per video clip, D = dialog, T = text, A = audio, I = image, V = video, MC = multiple choice, FB = fill in the blanks.

Dataset Comparison. In Table 2.2, we compare our dataset with other Video QA datasets. We highlight the presence of multiple modalities and their real-life nature, differentiating it from prior work. Specifically, LifeQA is the only existing Video QA dataset focusing on real-life understanding and is carefully constructed from hand-picked in-the-wild videos. In addition, it spans all typical audio and visual modalities and contains videos that are much longer than those in many other datasets. These qualities lead to a diverse, high-quality video dataset that is suitable for benchmarking current video QA systems and serves as a complement to existing QA datasets. Please refer to Section 2.2 for more details on the comparison.

More Examples. In Fig. 2.5, we present additional examples of instances in LifeQA. These examples demonstrate the variety of scenes and question types in LifeQA.

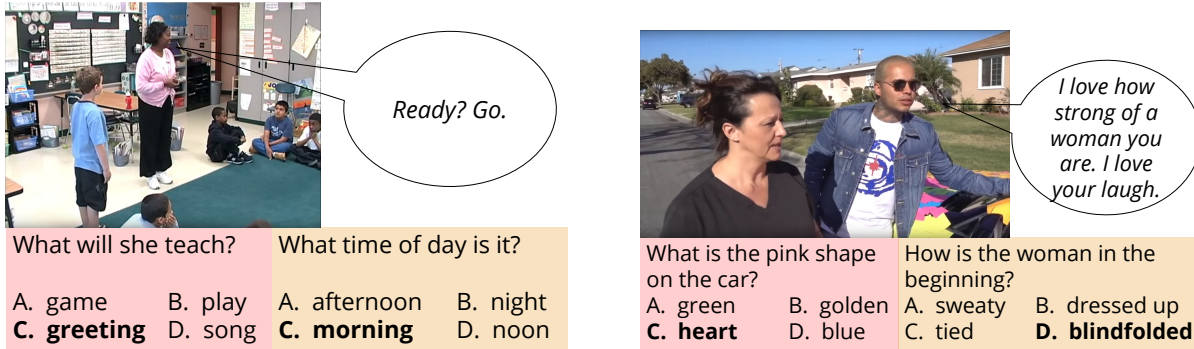


Figure 2.5: Additional instances from LifeQA. The videos capture various indoor and outdoor scenes, and the questions refer to visual and auditory concepts.

2.4 Experiments

We implement several models to demonstrate the task’s difficulty and explore biases, and compare their performance by measuring question-answering accuracy.

2.4.1 Baselines

We implement and evaluate several baselines, including simple heuristics and neural methods. We categorize these baselines according to their inputs (the question, the transcriptions, or the visual content) and whether they are trained from scratch or pretrained. By analyzing these baselines, we demonstrate the differences between evaluations of our data versus other non-real-life datasets.

Human baseline. We provided a human baseline, in which two workers were asked to answer a random sample of 101 questions. One worker first listened to the audio in the video without looking at the visual content, then answered the questions, and then repeated the same task using both modalities (i.e., listen to the audio and watch the video). The other worker did the same, but using the visual content – i.e., they first watched the video without listening to the audio. They answered the questions and then repeated the same task with both modalities. Note that this differs from the previous analysis in Fig. 2.3 as workers answer the questions using one modality at a time without knowing the correct answer a priori.

Question-only. We implement several baselines that use only the questions and their candidate answers. Three of these baselines use only the answers without any question;

Random chooses one out of the four options uniformly at random, and *Longest answer* and *shortest answer* choose the answer with the most or fewest number of tokens, respectively.

The first two baselines using the question are based on computing some similarity measure between the question and candidate answers. The first is *Word matching*, as defined by [258], which finds the answer with the most overlapping words with the question. The second is *Most similar answer*, which looks at word-level similarity, which we compute by using the average GloVe [178] embedding of the question and each answer and selecting the answer with the highest cosine similarity with the question. We use GloVe [178] embeddings with size 300 pretrained on 6B tokens from Wikipedia 2014 [187] and Gigaword5 [172].

Finally, we implement *ST-VQA-Text*, a variant of Spatio-Temporal VQA (ST-VQA) [99], which uses no visual information. It encodes the question with a 2-layer LSTM, then encodes the candidate answers and assigns a score to each one. The text is tokenized and represented using GloVe [178] embeddings of size 300 pretrained on the Common Crawl dataset.

Question + Transcriptions. We present several neural baselines that use the questions, answers, and transcriptions but omit the videos and audio.

Text-only LSTM and *text-only CNN* use neural models to encode the transcript, question, and answers separately. The former is a one-layer BiLSTM of hidden size 100. The latter is a 1D CNN with 100 filters of size two tokens and 100 filters of size three tokens. We then concatenate the transcript and question encodings and embed them with a two-layer network. We compute the dot product similarity between the question + transcription encoding with each possible answer and select the one with the highest score.

Second, we use a variant of BiDAF [205] in which we remove the component that predicts the likelihood of each token being the start and end of the span that is needed for SQuAD [187] because in LifeQA there are no such spans. We then compute the dot product between the final hidden state of the Modeling Layer and the representation of each answer choice, which serves as a score. We repeat this same process for both the question and the transcript.

Finally, we use a modified version of the end-to-end Memory Network (MemN2N) proposed by [225] based on [218] to handle multiple-choice question answering. The input to the model is the transcriptions, questions, and candidate answers. The transcription segments are obtained by mean-pooling the GloVe representation of the words for each segment. Our network has an attention layer over the transcriptions to pick the segments most relevant to the given question and is trained in an end-to-end fashion to select the correct answer.

Question + Vision. We use two variants of ST-VQA [99]. Both encode the video using a CNN followed by an LSTM, whose final hidden state is used as in *ST-VQA-Text*. *ST-VQA-*

Tp. uses the concatenation of the output of an ImageNet [44] pretrained ResNet152 [78] pool15 layer and of a Sports1M [106] pretrained C3D [230] fc6 layer as the video encoder. *ST-VQA-Sp.Tp.* computes a spatial attention map to decide what parts of the image are most useful and uses the `res5c` and `conv5b` of the two CNN encoders. Both use temporal attention maps to pool important information across video frames. We also tried a variant that uses RGB-I3D [24] (with `avg_pool` and `mixed_5c` layers respectively) instead of C3D, pretrained on ImageNet [44] and Kinetics [108] but do not report it because we obtained similar results.

Question + Transcriptions + Vision. We implement two neural models that use all modalities, TVQA [127] and MovieQA [225]. Both models use object detection networks to identify visual concepts in the corresponding video frames, allowing them to use the visual modality. For both, we use as visual inputs the output predictions of a Faster R-CNN [193] object detection model pretrained on Visual Genome [118].

Pretrained Baselines. Finally, we utilize the TVQA model pretrained on the TVQA dataset and evaluate it on two versions: with and without fine-tuning on LifeQA.

2.5 Results

In Table 2.3, we evaluate each model with a five-fold cross-validation, grouping by source video.¹ Similar to TVQA [127], the baselines trained from scratch do not generally benefit from visual information. Most models do not surpass ST-VQA-Text, a baseline that uses only the question and the available answers as input. This issue shows the presence of biases in the dataset, including the multiple-choice setup as opposed to free answer, which allows models to overfit to obtain better-than-random performance. It also demonstrates that leveraging real-life video data challenges existing systems.

The TVQA model shows a significant gain in performance when pretrained on the TVQA dataset, possibly due to the significantly larger training size. However, there is still a big gap between its performance and that of a human, providing evidence that this is a challenging benchmark. The same model can obtain 66.5% accuracy on the TVQA dataset with five answer choices instead of four. Moreover, the model cannot perform better even when fine-tuning, showing that the task is still challenging when given in-domain training data and

¹Note: we used 221 of the 275 video clips (50 out of 59 source videos) available when running the experiments.

Inputs	Model	Accuracy
	Random	25.0
A	Longest answer	30.6
	Shortest answer	21.5
Q+A	Word matching	24.8
	Most similar answer	35.2
	ST-VQA-Text	45.4
T+Q+A	BiDAF	43.3
	Text-only CNN	43.5
	Text-only LSTM	44.0
	Text-only Memory Network	37.9
	Human	63.4
V+Q+A	ST-VQA-Tp.	45.0
	ST-VQA-Sp.Tp.	44.6
	Human	48.5
V+T+Q+A	Multimodal Memory Network	38.2
	TVQA from scratch	41.1
	Pretrained TVQA w/o fine-tuning	51.8
	Pretrained TVQA w/ fine-tuning	51.6
	Human	90.6

Table 2.3: Baselines on the LifeQA dataset. In the first column, “A” stands for *answer*, “Q” for *question*, “T” for *transcripts*, and “V” for *visual modality*. When the transcripts are part of the input, the human performance is measured by using the audio instead.

hints that more robust models should be considered to close the gap instead of labeling a more significant amount of data to train on.

2.6 Conclusion

In this work, we introduced LifeQA, a real-life dataset for evaluating Video QA systems in real-life scenarios. Through several analyses and experimental evaluations, we showed that LifeQA presents a challenging task for existing models, with a significant gap in accuracy compared to human performance, thus suggesting that future research is necessary to leverage the multimodal features in this domain. The dataset is publicly available at <https://lit.eecs.umich.edu/lifeqa/>.²

This chapter considered video understanding in real-life settings, specifically addressing people’s daily lives. Still, there are other overlooked real-life domains to take into account, such as nature.

²Given the relatively small amount of video data we share and the fact that it is drawn from public sources, the sharing of this data falls under “fair use.”

CHAPTER 3

Video Understanding in In-The-Wild Scenarios

3.1 Introduction

Video understanding plays a vital role in developing competent AI systems, enabling the effective processing of different modalities of information [136]. Various tasks have been proposed to examine the ability of models to understand videos, including video question answering (Video QA), video captioning, and fill-in-the-blank tasks [249, 229, 29]. Recent years have witnessed significant progress in video understanding, including new benchmarks [225, 76] and advanced sophisticated models [102, 183].

However, there are several drawbacks associated with existing video understanding research. First, existing video understanding benchmarks focus on everyday human activities as typically appearing in cooking videos [281] or in movies [225], leading to a limited set of video domains. Second, most video understanding benchmarks adopt a multiple-choice format, where models select an answer from a set of candidates [99, 26]. Models trained under such a setting cannot be used in real-life applications because candidate answers are not provided [29]. Third, videos included in existing benchmarks are typically short [112], and the performance of models on longer videos is not well studied.

We address these challenges in our dataset construction process. First, we propose the WILDQA dataset in which we collect “in the wild” videos recorded in the outside world, going beyond daily human activities. Figure 3.2 shows the difference between the WILDQA dataset and previous question answering datasets. Second, we adopt the challenging answer generation approach, aiming to build a system that can answer questions with an open-ended answer rather than selecting from a predefined set of candidate answers. Third, the average video length in our dataset is one minute, longer than the video clips in most of the existing datasets in Table 3.3, which presents a novel challenge for video understanding algorithms.

Using the WILDQA dataset, we address two main tasks. First, we address the task of video question answering (**Video QA**), aiming to generate open-ended answers. Second,

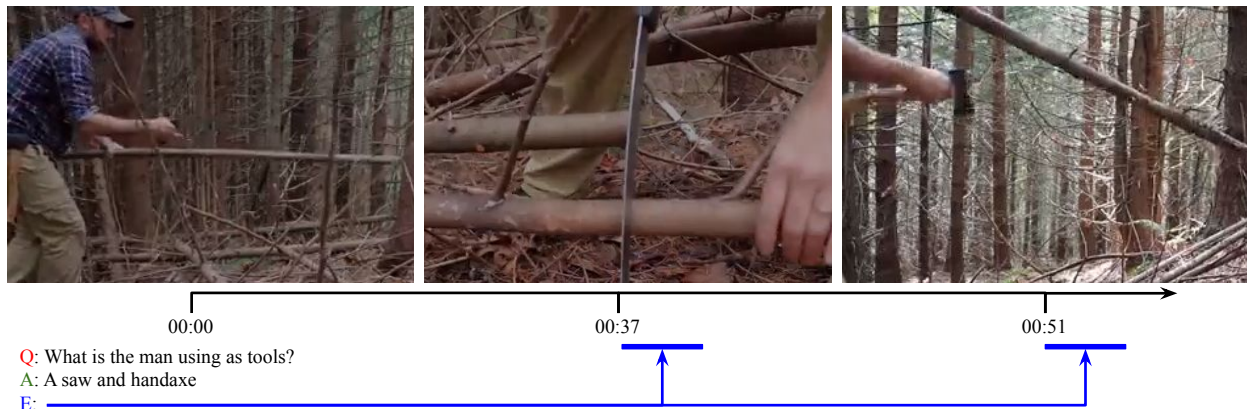


Figure 3.1: An example from our WildQA dataset, showing a question (Q), an answer (A), and evidence (E) that supports the answer. The corresponding part of the videos is provided as evidence for the question.

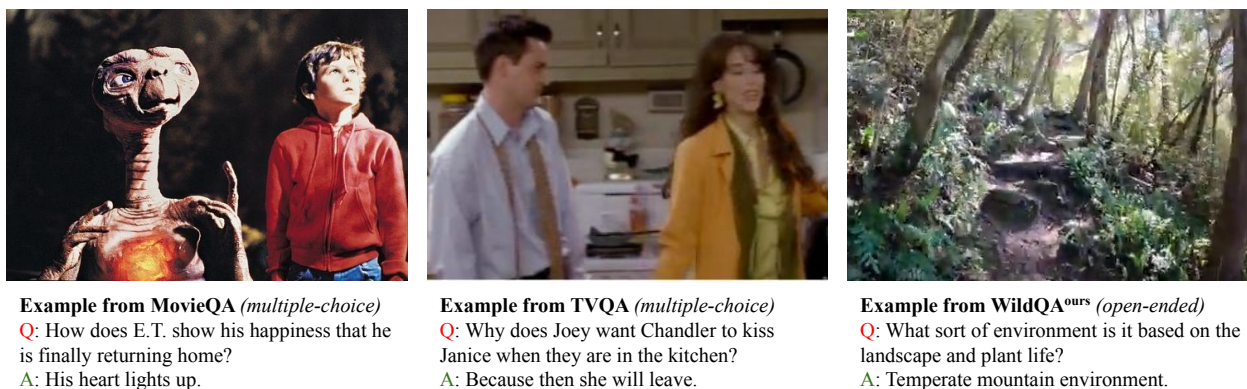


Figure 3.2: Examples from MovieQA [225], TVQA [127], and our WildQA dataset. The previous datasets mostly focus on human interactions in a multiple-choice setting, while ours focus on scenes recorded in the outside world in an open-ended setting. We only list a single answer here for illustration purposes.

we introduce the task of retrieving visual support for a given question and answer (**Video Evidence Selection**). Finding the relevant frames in a video for a given question-answer pair can help a system in its reasoning process. It aligns with ongoing efforts to build interpretable models [97]. We evaluate several baseline models for each task, including multi-task models that combine the two tasks. Figure 3.1 shows an example from our dataset, including an example of a question, answer, and supporting video evidence. To summarize, the main contributions of this chapter are:

1. We propose WILDQA, a multimodal video understanding dataset where video scenes are recorded in the outside world.
2. We propose two tasks for WILDQA: Video QA and Video Evidence Selection, aiming to build more interpretable systems.
3. We test several baseline models; experimental results show that our dataset poses new challenges to the vision and language research communities.

3.2 Related Work

Multimodal Question Answering. Two popular and representative tasks are Visual Question Answering (Visual QA) on images and Video Question Answering (Video QA) on videos. Visual QA has attracted attention for a long time [153, 271, 192, 283]. Recently, much progress has been made in Video QA. Researchers proposed various datasets such as TVQA that contain videos from movies or TV series [225, 127, 128] or videos from the Internet spanning from YouTube videos to Tumblr GIFs [268, 257, 99, 263]. Other datasets such as MSVD-QA [249] contain videos from the existing corpus [30] or cartoon videos [112]. Recent Video QA datasets have stronger focuses such as temporal relations [163], multi-step and non-factoid answers [38], natural interactions [265], characters in the video [36], question answering in real life [26], incorporating external knowledge [69], and videos recorded from the egocentric view [54, 76]. To our knowledge, we are the first to collect videos from the outside world.

Researchers have also developed various methods to handle the Video QA task, including joint reasoning of the spatial and temporal structure of a video [277, 66, 89, 101], integrating memory to keep track of past and future frames [112, 63, 278, 55, 260], various attention mechanisms [281, 273, 137, 261, 111, 102], and others. Recently, pre-trained models have proved helpful in multiple visual and language tasks [183, 33, 267]. However, the pre-trained visual and language models are typically encoder-only and cannot generate an answer in natural language on their own. Thus, encoder-only models do not fit our task’s open-ended video question-answering setting.

Previous work has also investigated various reasoning tasks in a multimodal setting [67, 256, 68, 266]. Although it is not our focus, some questions in our dataset require a high reasoning ability. Moreover, since domain experts created our dataset, domain knowledge is also involved in the questions.

Moment Retrieval. Moment Retrieval is the task of retrieving a short moment from a large video corpus given a natural language query [51, 129]. Researchers have proposed or adapted various datasets for this task [117, 83, 64, 129]. The task of retrieving relevant parts in the video given the question (Video Evidence Selection) in our proposed dataset is akin to Moment Retrieval. However, moment retrieval focuses on retrieving the part of videos that the question describes, while Video Evidence Selection is to find parts of videos that can support the answer to the questions as shown in Fig. 3.1. Prior work such as Tutorial-VQA [38] also adopted the setting of providing parts of the videos as answers to the question, but they did not include any text answers in their dataset.

Few-shot Learning. Recently, there has been a trend to evaluate neural models in a few-shot learning setting [90, 162, 222, 139, 124, 180], where the model is tuned with a small portion of the data and tested against the rest. We adopt the few-shot learning setting for our dataset for both Video QA and Video Evidence Selection.

3.3 WildQA Dataset

Video Selection and Processing. Following [265, 26], we start by collecting videos from YouTube. First, we identify five domains that primarily consist of outdoor scenes and are representative of the outside world, namely, *Agriculture*, *Geography*, *Human Survival*, *Natural Disasters*, and *Military*. We then manually collected videos from relevant YouTube channels for each domain.

Because the raw videos can be as long as an hour, we split the raw videos into short clips using PySceneDetect,¹ and concatenate these short clips so that the output video is approximately one minute. We use the output videos for the annotation process described below. More details for the video selection and processing steps are discussed in Appendix A.1.1.

Question, Answer, and Evidence Annotation. Our annotation process has two phases, as shown in Fig. 3.3. In **Phase 1**, annotators watch the video clips and develop a hypothetical motivation. They ask one or more **questions** and provide an **answer** to each of the questions

¹PySceneDetect uses the OpenCV [17] to find scene changes in video clips (py.senedetect.com).

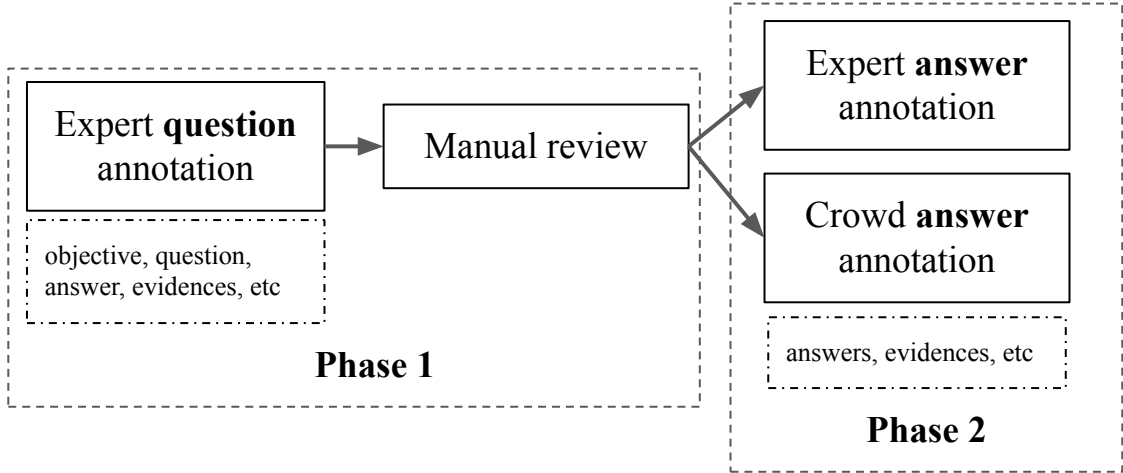


Figure 3.3: The two phases of data annotation.

Domain	Video count	Question count
<i>Agriculture</i>	85	109
<i>Human Survival</i>	95	309
<i>Natural Disaster</i>	70	187
<i>Geography</i>	46	110
<i>Military</i>	73	201
Total	369	916

Table 3.1: Video and question count for each domain.

they ask. We also instruct annotators to provide all the relevant parts in videos as pieces of **evidence** to support the answer to their question. After this step in the data collection, three researchers manually reviewed all the question-answer pairs for quality purposes. Next, in **Phase 2**, we collect more **answers** and **evidences** for each question from Phase 1. Over the entire annotation process, annotators spent a total of **556.81 annotation hours**, split into 77.05 hours in Phase 1 and 479.76 in Phase 2. Appendices A.1.2, A.1.3, and A.1.5 present the annotation instructions, annotation interfaces, and reviewing process for question-answer pairs, respectively.

Because we want to collect questions that domain experts are interested in, as opposed to arbitrary questions, domain experts carry out the Phase 1 annotation. We conducted a pilot study to demonstrate the difference in the quality of questions collected from domain experts versus non-experts. Appendices A.1.4 and A.1.6 discuss the pilot study and the annotators’ expertise, respectively.

Videos	369
Duration (in seconds)	71.22 ± 26.47
Questions	916
Question per video	2.48 ± 1.38
Question length (#tokens)	7.09 ± 2.60
Answer per question	2.22 ± 0.69
Answer length (#tokens)	9.08 ± 8.15
Evidence per answer	1.18 ± 0.80
Evidence length (s)	9.64 ± 10.96

Table 3.2: Dataset statistics for WildQA.

Dataset	Domain	VE?	#Videos	# Q	Avg dur. (s)	Annotation	QA Task
MovieQA [225]	Movies	✓	6.7K	6.4K	203	Manual	MC
VideoQA (FiB) [281]	Cooking, movies, web		109K	390K	33	Automatic	MC
MSRVTT-QA [249]	General life videos		10K	243K	15	Automatic	OE
MovieFIB [151]	Movies		128K	348K	5	Automatic	OE
TVQA [127]	TV shows	✓	21.8K	152K	76	Manual	MC
ActivityNet-QA [263]	Human activity		5.8K	58K	180	Manual	OE
TVQA+ [128]	TV shows	✓	4.2K	29.4K	60	Manual	MC, ES
KnowIT VQA [69]	TV shows		12K	24K	20	Manual	MC
LifeQA [26]	Daily life		275	2.3K	74	Manual	MC
TutorialVQA [38]	Instructions	✓	76	6.2K	–	Manual	ES
NExT-QA [248]	Daily life		5.4K	52K	44	Manual	MC, OE
FIBER [29]	Human actions		28K	2K	10	Manual	OE
WildQA	In-the-wild	✓	369	916	71	Manual	OE, ES

Table 3.3: Comparison between our WILDQA and other existing datasets. **VE?**: Whether the dataset provides “Video Evidences”?; **MC**: “Multiple Choice” question answering; **OE**: “Open Ended” question answering; **ES**: “Evidence Selection”. We adapt the comparison table from [279].

Dataset Statistics. Tables 3.1 and 3.2 present statistics of the videos and associated questions for each of the five domains, along with other relevant statistics. Figure 3.4 shows the distribution of question types. Appendix A.1.7 discusses more statistics.

Dataset Comparison. Table 3.3 shows the comparison between WILDQA and other existing datasets.

3.4 Video Question Answering

Following [252], we adopt **free-form open-ended** video question answering for our video question answering (Video QA) task. Given a question \mathbf{q} and a video \mathbf{v} , the task is to

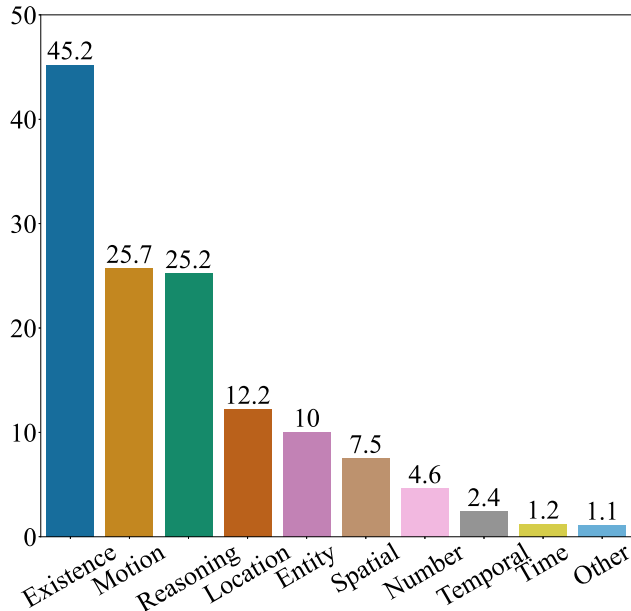


Figure 3.4: Percentage distribution of question types. Because one question might be classified into multiple categories, the scale summation is larger than 100%.

generate an answer \mathbf{a} in natural language.

We adopt a **few-shot learning** setting on our dataset, where models are fine-tuned on question-answer pairs corresponding to 30% of the videos for each domain. The tuned models are tested on data for the remaining 70% videos. The reason is that the time to annotate 30% of the data is around 23 hours, during which there are around 50 data points annotated for each domain, which is acceptable. We hypothesize that it is realistic to have such a setting because the potential end-users could spend around a day or two collecting data, and we can then quickly tune a model using it. Moreover, no repeated videos appear in different splits, following [127]. We end up having 264 question-answer pairs for 108 videos in our dev set and 652 pairs for 261 videos in the test set. We adopt BLEU [170] and ROUGE [140] as the metrics to measure the quality of the generated answer. We run each model 3 times and report the scores of mean \pm standard deviation in Table 3.4.

3.4.1 Baselines

Human Baselines. We report the average BLEU and ROUGE scores by leaving one annotator out in Table 3.4 (**Human**).

Text-only Models. We implement several baselines that only use the question-answer pairs in the dev set. **Random** randomly chooses answers from the dev set. **Common** always



Q: What type of weather is happening?

A: Flooding and rain.

The weather is rain and flood.



Q: Where is the road at?

A: It is in a tundra environment

The road zig-zags across the landscape.

The road winds through a mountainous landscape.

The road is in an elevated area.

Figure 3.5: Examples of questions (Q) and answers (A) from WildQA. The first answer is collected during Phase 1 of the annotation process; all remaining answers are collected in Phase 2. More analyses in Appendix A.1.7.

Model name	ROUGE-1	ROUGE-2	ROUGE-L
Random	5.0 ± 0.2	0.5 ± 0.1	4.9 ± 0.2
Common	10.6 ± 0.0	0.0 ± 0.0	10.6 ± 0.0
Closest	19.5 ± 0.0	6.2 ± 0.0	18.7 ± 0.0
T5 T ^{0-shot}	0.8 ± 0.0	0.0 ± 0.0	0.8 ± 0.0
T5 T	33.8 ± 0.2	17.7 ± 0.1	32.4 ± 0.3
T5 T+V	33.1 ± 0.3	17.3 ± 0.4	31.9 ± 0.2
Multi _{T+V,I0}	34.0 ± 0.5	18.8 ± 0.7	32.8 ± 0.6
Multi _{T+V,SE}	33.8 ± 0.8	18.5 ± 0.7	32.5 ± 0.8
Human	40.8 ± 0.0	18.1 ± 0.0	36.3 ± 0.0

Table 3.4: ROUGE scores for the task of Video Question Answering. For comparison, we test the out-of-box T5 model under the zero-shot setting (T5 T^{0-shot}).

predicts the most common answer in the dev set; **Closest** employs embedding produced by a pretrained **roberta-base** model [144]. In the inference, **Closest** retrieves the answers for the dev set question whose embedding has the highest cosine similarity to the test question. We also fine-tune T5 [184] using question-answer pairs from the dev set (T5 T).

Text + Visual Models. Following [29], we concatenate the text features with the visual features and input the concatenated features to the T5 model (T5 T+V). We extract I3D [24] video features and take one feature per second.

Multi-task Learning. Multi-task learning has succeeded in various domains [39, 45, 73]. Following [25], we train Multi_{T+V,SE} which combines T5 T+V and T5 SE (the Video Evidence Selection model described in Section 3.5) with a shared T5 encoder between the tasks of Video Question Answering and Video Evidence Selection. We also train Multi_{T+V,I0} which combines T5 T+V and T5 I0 (another Video Evidence Selection model described in Section 3.5) in a similar way. The loss function during the fine-tuning is:

$$L = \alpha L_1 + \beta L_2 \tag{3.1}$$

Where L_1, L_2 are the losses for Video Question Answering and Video Evidence Selection, respectively; α, β are the weights for the two tasks. We present the selection process behind the values of α and β in Appendix A.3.

3.4.2 Results

Table 3.4 reports F1 scores of ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) for our baseline models. For comparison, we also test the out-of-box T5 model on our test split under the zero-shot setting (T5 Text^{0-shot} in Table 3.5).

T5-based models significantly outperform the random baselines and the out-of-box T5 model, suggesting that the T5-based models acquire certain levels of question-answering ability in the tuning stage. However, adding visual features does not improve the model’s performance. This issue might be due to the *challenges of attending to the visual features at the corresponding parts in the video* because both models under multi-task learning outperform the text-only baseline, suggesting that attending to the correct part of the video helps the answer generation process.

All baseline models underperform human baselines on ROUGE scores, especially ROUGE-1 and ROUGE-L scores, suggesting room for improvement. However, the ROUGE-2 score for human annotators is low because although human annotators tend to use the same word to describe the object that appears in the video, there are significant variations in expressing the ideas of their answers. More discussions on the diversity of the answers are in Appendix A.1.7.

3.5 Video Evidence Selection

Similar to [38], given a video \mathbf{v} and a question \mathbf{q} , the video evidence selection task consists of predicting $\{(\mathbf{s}_1, \mathbf{e}_1), (\mathbf{s}_2, \mathbf{e}_2), \dots\}$, where $(\mathbf{s}_i, \mathbf{e}_i)$ represents the time for start \mathbf{s} and end \mathbf{e} of a single span within the video \mathbf{v} . We also adopt the few-shot learning setting as described in Section 3.4 for the task of Video Evidence Selection. Similar to [48], we design an Intersection-Over-Union (IOU) metric borrowed from [52]. We define IOU as follows: given two time spans in the video, IOU is defined as the length of their intersection divided by the length of their union. The prediction matches if it overlaps with any ground truth span by more than the threshold (0.5, following [48]). We use these partial matches to calculate an F1 score (IOU-F1 scores). As described in Section 3.4, we run each model three times and report the scores of mean \pm standard deviation in Table 3.5.

3.5.1 Baselines

As described in Section 3.4, we compute the average IOU-F1 score on the annotations from one annotator against the remaining annotators. We denote this metric as **Human**. The **Random** baseline consists of randomly choosing the start and end of a part within the original video as evidence. Similar to the structure [47] experiment on SQuAD [187], we build T5 SE; here,

Model name	IOU-F1
Random	2.5 ± 0.3
T5 IO	1.1 ± 0.2
T5 SE	4.5 ± 0.8
Multi _{T+v,IO}	1.4 ± 0.3
Multi _{T+v,SE}	3.7 ± 2.4
Human	18.4 ± 0.0

Table 3.5: IOU-F1 scores for Video Evidence Selection.

we feed the concatenated question embeddings and I3D visual features to the T5 encoder, and the T5 encoder outputs a sequence of the encoded states. We treat the subsequence corresponding to the visual features as the encoded hidden sequence $T_m \in R^H$ for the video frames (H denotes the dimension of the hidden sequence). We then multiply the sequence with two vectors $S, E \in R^H$. The T_i and T_j that maximize the likelihood are predicted as the **start and the end of the evidence**, respectively. During the training, we maximize their joint probability:

$$P_i P_j = \frac{e^{S \cdot T_i}}{\sum_m e^{S \cdot T_m}} \frac{e^{E \cdot T_j}}{\sum_m e^{E \cdot T_m}}$$

where P_i and P_j are the probability for the i being the start and j the end of the evidence, respectively.

Inspired by the Inside-Outside-Beginning (“IOB”) tagging scheme [188], we also formulate the evidence finding as a task of tagging whether a video frame is inside (“I”) the evidence or outside (“O”) the evidence. We then build T5 IO by feeding the concatenated features to a T5 encoder. Similar to T5 Start End, we have an encoded sequence of $T_m \in R^H$ corresponding to the video frames. We then multiply the sequence with a vector $L \in R^H$ and apply a sigmoid function to the multiplication result. The model predicts the frame as “I” if the value at the corresponding position is greater than or equal to 0.5. Otherwise, it predicts “O”. We test Multi_{T+v,IO} and Multi_{T+v,SE} described in Section 3.4 on Video Evidence Selection as well.

3.5.2 Results

Table 3.5 shows the performance of the baseline models on the Video Evidence Selection task. All the baseline models perform significantly worse than the human annotators and sometimes worse than the random baseline. This result is understandable because selecting evidence from a long video can be complex. Additionally, multi-task learning makes the model’s performance

Type	R1	IOU-F1
<i>Existence</i>	33.3 ± 0.3	5.3 ± 0.3
<i>Motion</i>	32.8 ± 0.6	3.1 ± 2.0
<i>Reasoning</i>	33.3 ± 0.4	3.1 ± 1.3
<i>Location</i>	26.2 ± 10.7	4.4 ± 1.4
<i>Entity</i>	33.2 ± 0.7	5.2 ± 0.7
<i>Spatial</i>	32.2 ± 0.6	2.4 ± 1.7
<i>Number</i>	33.8 ± 0.4	4.5 ± 0.7
<i>Temporal</i>	33.8 ± 0.6	3.8 ± 0.5
<i>Time</i>	33.1 ± 0.8	5.7 ± 1.0
<i>Other</i>	33.2 ± 0.6	5.3 ± 0.9

Table 3.6: $\text{Multi}_{T+V,SE}$ performance on different question types for Video QA (ROUGE-1) and for Video Evidence Selection (IOU-F1).

worse. However, this could be because the Video Evidence Selection is challenging, and all the baseline models struggle with such a task. Although multi-task learning does not help Video Evidence Selection, as mentioned in Section 3.4, training with Video Evidence Selection does help Video QA. Thus, Video Evidence Selection is still essential to improve a model’s ability to answer questions. We include more ablation studies in Appendix A.4.1.

3.5.3 Analysis and Discussion

Model Performance vs. Question Types. Table 3.6 shows $\text{Multi}_{T+V,SE}$ ’s performance on different question types for Video QA and Video Evidence Selection respectively. Other ROUGE scores for Video QA follow similar trends as shown in Fig. A.7. According to Table 3.6, the model achieves good ROUGE-1 scores for Video QA when the model has a good IOU-F1 score for Video Evidence Selection such as its performance on *Existence*. The model has the highest ROUGE-1 variation on *Location* question types, with a relatively large variation for IOU-F1. The model’s ROUGE-1 score on *Spatial* questions is relatively low, with the lowest IOU-F1 score. $\text{Multi}_{T+V,SE}$ excels at question type *Entity* and *Existence* with relatively high IOU-F1 scores. One possible explanation could be that the average length of the answers generated for *Entity* and *Existence* are around eight tokens, which might be easier for the model to ground to the relevant part in the video.

Interestingly, even if the answers have similar lengths, the model struggles on *Motion* questions (with a relatively low IOU-F1 score). A possible reason could be that this type of question provides a very abstract description of the action, which makes it hard for the model to attend to the relevant part of the video. For instance, an example of a *Motion* question is “Are there any structure or natural features being affected?”. To attend to the corresponding

Model name	R1	R2	RL
T5 T ^{0-shot}	0.8 ± 0.0	0.0 ± 0.0	0.8 ± 0.0
T5 T ^{0-shot} _{TVQA}	9.1 ± 0.0	1.2 ± 0.0	8.8 ± 0.0
T5 T _{TVQA,ours}	32.4 ± 0.2	17.5 ± 0.2	31.6 ± 0.2
T5 T _{ours}	33.8 ± 0.2	17.7 ± 0.1	32.4 ± 0.3
T5 T+V ^{0-shot} _{TVQA}	20.3 ± 0.0	8.1 ± 0.0	20.1 ± 0.0
T5 T+V _{ours}	33.1 ± 0.3	17.3 ± 0.4	31.9 ± 0.2
T5 T+V _{TVQA,ours}	33.7 ± 0.2	18.3 ± 0.1	32.6 ± 0.1

Table 3.7: ROUGE scores for the task of Video Question Answering for few-shot learning setting (the standard setting in our WildQA dataset introduced in Section 3.4) and zero-shot learning setting (“0-shot” in the superscript). Subscript “TVQA” means pre-training on the TVQA [127] dataset; subscript “TVQA,ours” means first pre-training the model on TVQA, then tuning the model on our WildQA dataset; subscript “ours” means tuning the model directly on our WildQA dataset.

period in the video, the model needs to understand the word “affected” and the objects that are actually affected, which can be very difficult. The model also struggles to attend to the correct places in the video for the *Spatial* type of question. This issue might occur because there is more than one entity in *Spatial* type of questions, and the model needs to locate all the objects appearing in various parts of the video, which is similarly complex. For instance, for the question “*What effects did the weather have?*”, the model needs to attend to “*debris in the air*”, “*truck turnover*” and “*destruction of buildings*”. For *Location* type of questions such as “*What sorts of terrain is the vegetation present in?*”, it might be difficult to attend to all the terrains of “*forest*”, “*plateaus*”, “*mountainous*”, “*valleys*”, and “*arboreal*” and to include them in the answer.

Domain Adaptation. Furthermore, we tune the `MultiT+V,SE` model on the dev set data from a single domain and test it against data from other domains. Figures 3.6 and 3.7 show the model’s performance in different tuning and testing domains. Interestingly, the diagonal cells do not always have the darkest color, which indicates that inter-relations exist across domains. For instance, the model tuned on *Geography* performs relatively better for Video QA on *Human Survival* and *Agriculture* rather than itself. This result suggests that the questions and videos from *Geography*, *Agriculture*, and *Human Survival* exhibit some similarity so that the model tuned on one domain can answer questions from the other domains relatively well. But answering questions from *Geography* can introduce the domain knowledge; an example of the answer is “*Mountainous, temperate forest.*”, where “*temperate forest*” is one of the terminologies specific to *Geography* domain. Training on these terminologies might

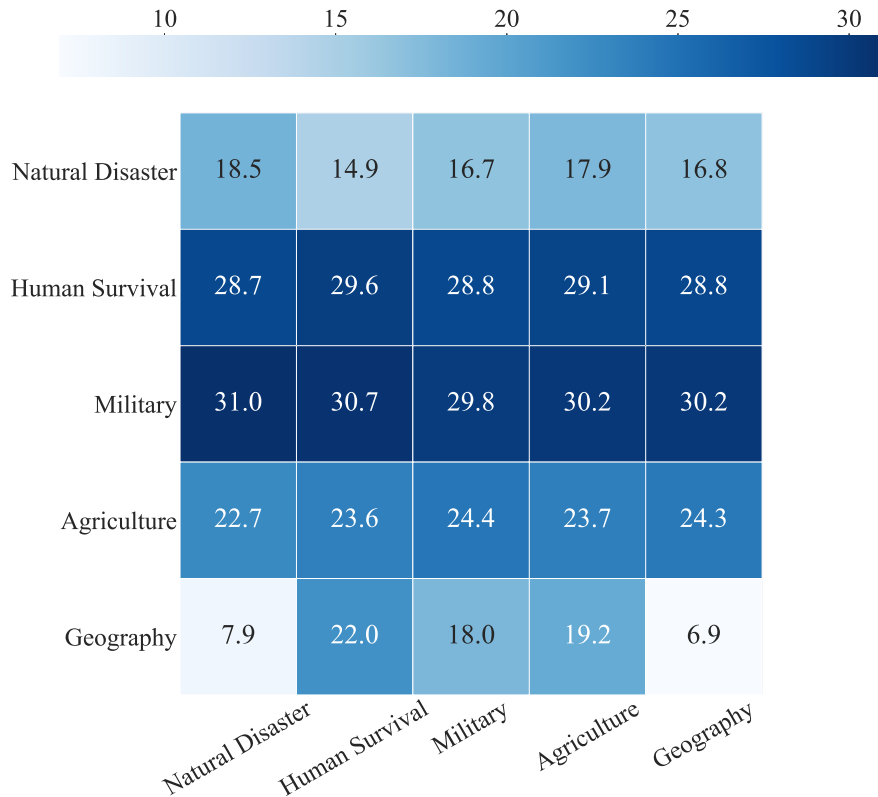


Figure 3.6: $\text{Multi}_{T+V,SE}$ performance (ROUGE-1) for Video QA when tuned on a single domain (y-axis) and tested against each domain (x-axis). The performances by the rest of the metrics for Video QA resemble the pattern here and are reported in Appendix A.4.

confuse the model and hurt the performance. Thus, future research might be needed to better incorporate domain knowledge into multimodal question answering.

As for Video Evidence Selection, the patterns generally resemble the pattern in Fig. 3.6, which means that the model typically answers a question better if it can attend to the relevant part in the video. However, when tuned on *Human Survival* and tested on *Natural Disaster*, the model performs relatively well on Video QA (with a 28.7 ROUGE-1 score) but less well on Video Evidence Selection (with a 0.7 IOU-F1 score). This phenomenon might indicate that the model picks up some common patterns in the text rather than reasoning about the video and the question in an expected manner.

Pre-training on Other Datasets. We also pre-train the T5 T and T5 T+V using TVQA [127], a large-scale multimodal question-answering dataset with videos from TV series. We report the zero-shot learning performances as well as the few-shot learning performances for T5 T and T5 T+V in Table 3.7. We can see that pre-training on TVQA for text-only T5 T does not help, which shows that our dataset’s question styles might differ

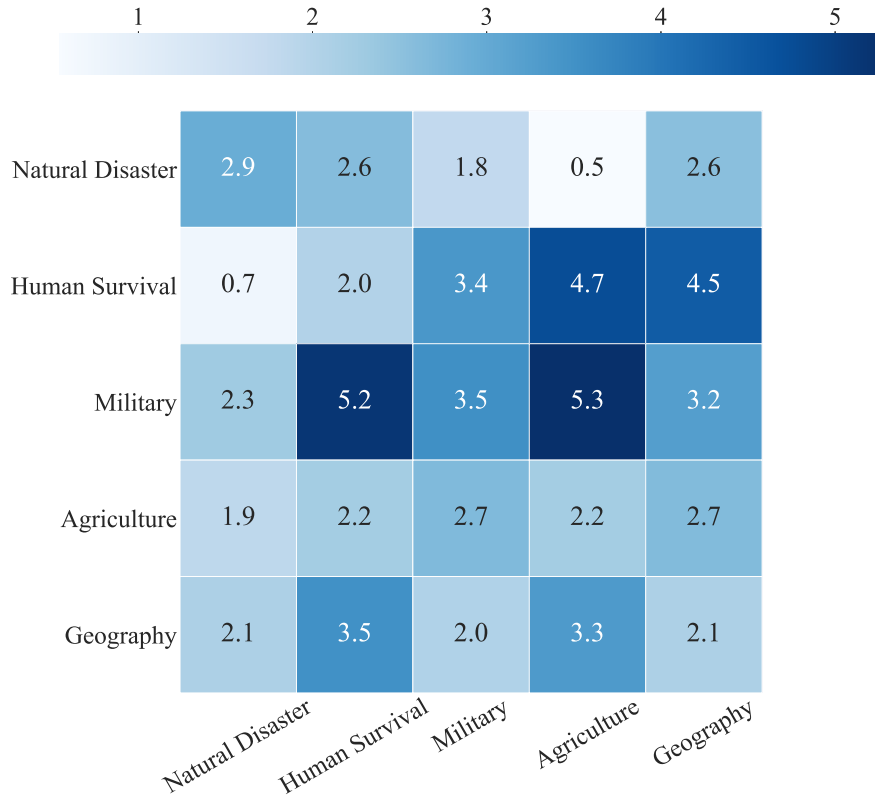


Figure 3.7: $\text{Multi}_{T+V,SE}$ performance (IOU-F1) for Video Evidence Selection when tuned on a single domain (y-axis) and tested against each domain (x-axis).

from TVQA. For T5 T+V, which uses both text and visual features, pre-training on TVQA does help the model, which suggests that the pre-training helps the model take advantage of the visual features. T5 T+V pre-trained on TVQA underperforms T5 T+V trained together with T5 IO (the $\text{Multi}_{T+V,SE}$ model) according to Table 3.4 and Table 3.7, suggesting that attending to the relevant part in the video helps the model better than training the model on more data. However, pre-training the model on the TVQA dataset reduces the variance of model performance, which suggests that training the model with more data helps the model perform consistently.

3.6 Conclusion

This chapter introduced a new and challenging benchmark, WILDQA, to promote domain diversity for video understanding. Specifically, we focused on five domains that involve long videos recorded in the outside world, which can be helpful for applications in these domains. We proposed generating open-ended answers instead of the traditional multiple-choice setting for Video Question Answering. We believe open-ended answer generation can help construct

systems that answer end users' questions more naturally. We also proposed the task of video evidence selection to help the model focus on the relevant parts of the videos. Through experiments, we showed the feasibility of these tasks and showed that joint training for both Video Question Answering and Video Evidence Selection can improve the models' performance. In addition, we found that it is easier to understand models' behavior by knowing which part of the video they attend to when answering a question. We believe this is a crucial step towards a trustworthy, explainable multimodal system. The dataset is available at <https://lit.eecs.umich.edu/wildqa/>.

So far, we considered the study overlooked domains. Can we consider other aspects, such as complex ways of human behavior?

CHAPTER 4

Addressing Real-Life Human Behavior

Contributions Contents in this chapter come from a published work [27] with joint contributions from collaborators in the University of Michigan, the Singapore University of Technology and Design, and the National University of Singapore. The following parts constitute novel contributions to this thesis: Sections 4.2.1, 4.3.1, 4.3.2, and 4.5.1.

4.1 Introduction

Sarcasm is a form of expression commonly used by people to express their contempt. The speaker usually employs irony along with negativity. For instance, in the following sarcastic utterance: “*Nice perfume. How long did you marinate in it?*”; the sarcasm is explicitly expressed as the speaker appears to praise the other person when, in reality, they mean the opposite (negative) as they associate it with food. However, in other cases, speakers express sarcasm without explicit linguistic markers, demanding further signals to reveal the speaker’s true intentions. For example, a speaker can express sarcasm by combining verbal and non-verbal signals, such as changing their tone, overemphasizing a word, or with a “poker” face. Furthermore, detecting sarcasm requires finding a linguistic or contextual contraction, which involves further information, either from multiple modalities [200, 159] or from the dialogue’s context history.

This chapter studies the importance of conversational context and multimodality in detecting sarcasm and presents a new benchmark to facilitate research in this area. Concretely, this chapter’s contributions are threefold:

1. We devise a new corpus, MUSTARD, for multimodal sarcasm research with high-quality annotations. It includes conversational context and multimodal (video, audio, and language) features.
2. We present multiple baselines and find that multimodal models are more effective than unimodal variants.

3. We provide preceding turns in the dialogue as context information. Consequently, this property of MUsTARD leads to a new sub-task for future work: *sarcasm detection in a conversational context*.

4.2 Dataset

Here, we present a new dataset and benchmark for studying sarcasm in short videos called MUsTARD (MUltimodal SARcasm Detection). Our data collection consisted of obtaining and segmenting episodes of The Big Bang Theory and gathering YouTube videos from different moments of the TV series Friends, the Golden Girls, and a one-time sketch dubbed Sarcasmaholics Anonymous. We collected some videos from MELD [181] to obtain non-sarcastic videos. We conducted a manual annotation as described next.

4.2.1 Annotation Process

We built a web-based annotation interface that shows each video along with its transcript and requests annotations for sarcasm. We also ask the annotators to flag misaligned videos, i.e., cases where the audio or video is not synchronized correctly. The interface allows the annotators to watch a context video consisting of the previous video utterances whenever necessary. Given the large number of videos to be annotated, we request annotations in batches of four videos at a time. We show our web interface in Fig. 4.1.

We conduct the annotation in two steps. First, we annotate the videos from The Big Bang Theory, as it contains the most extensive set of videos. Second, we annotate the remaining videos belonging to the other sources. The annotation is conducted by two graduate students who were first provided easy examples of explicit sarcastic content to illustrate sarcasm in videos. Each annotator labeled the complete set of videos independently.

For the first step, after annotating the first part – consisting of 5,884 utterances from The Big Bang Theory – we noticed that most were labeled as non-sarcastic (98% were considered non-sarcastic by both). In addition, our initial inter-annotation agreement was low (Kappa score is 0.1463). We thus decided to stop the annotation process and reconcile the annotation differences before proceeding further. The annotators discussed their disagreements for a subset of 20 videos and then re-annotated the videos. This time, we obtained an improved inter-annotator agreement of 0.2326. A third annotator reconciled the annotation disagreements by identifying the disagreement cases, watching the videos again, and deciding the correct label for each one.

Next, we annotate the second part, comprising 624 videos drawn from Friends, The

Video 1



“Can we maybe put the phones down and have an actual human conversation?”

[Click here to show video context.](#)

Does the video contains sarcasm?

- Yes
- No

Are the video and audio correctly aligned?

- Yes
- No

Figure 4.1: Graphical user interface used by the annotators to label the videos in our dataset.

Golden Girls, and Sarcasmaholics Anonymous. As before, the two annotators label each video independently. The inter-annotator agreement has a Kappa score of 0.5877. Again, a third annotator reconciled the differences.

The resulting set of annotations consists of 345 videos labeled as sarcastic and 6,020 videos labeled as non-sarcastic, for a total of 6,365 videos.

4.2.2 Transcriptions

Since we collected videos from several sources, some had subtitles or transcripts readily available. This is particularly true for videos from Big Bang Theory and MELD [181]. We use MELD’s transcriptions directly. For Big Bang Theory, we extracted the transcript by applying manual sub-string matching on the episode subtitles. The remaining videos are manually transcribed.

4.3 Multimodal Feature Extraction

We extract video, text, and speech features for the videos in MUsTARD. Here we describe the process we followed to obtain the text and video features:

4.3.1 Text Features

We represent the textual utterances in the dataset using BERT [47], which provides a sentence representation $\mathbf{u}_t \in \mathbb{R}^{d_t}$ for every utterance u . In particular, we average the last four transformer layers of the first token ([CLS]) in the utterance – using the BERT-base model – to get a unique utterance representation of size $d_t = 768$. We also considered averaging Common Crawl pre-trained 300-dimensional GloVe [178] word vectors for each token; however, it resulted in lower performance than BERT features.

4.3.2 Video Features

We extract visual features for each of the f frames in the utterance video using a pool5 layer of an ImageNet [44] pretrained ResNet-152 [78] image classification model. We first preprocess every frame by resizing, center-cropping, and normalizing it. To obtain a visual representation of each utterance, we compute the mean of the obtained $d_v = 2048$ dimensional feature vector \mathbf{u}_i^v for every frame: $\mathbf{u}_v = \frac{1}{f} (\sum_i \mathbf{u}_i^v) \in \mathbb{R}^{d_v}$. While we could use more advanced visual encoding techniques (e.g., recurrent neural network encoding techniques), we decided to use the same averaging strategy as the other modalities.

4.4 Experiments

We perform several experiments to evaluate the modalities separately and combined. On top of this, we study the importance of the speaker and context.

4.4.1 Experimental Setup

We split the evaluation into two parts. First, we perform a stratified five-fold cross-validation. In some experiments, we further divide each fold’s training set to obtain a validation set (e.g., for hyper-parameter selection; see below). Given that there is likely a speaker overlap across training and test sets in the folds (*speaker-dependent* setup), the second part of our evaluation separates training and test sets to avoid speaker overlap (*speaker-independent* setup; see Section 4.5). In this case, the training set uses utterances from The Golden Girls, Sarcasmaholics Anonymous, and The Big Bang Theory. For the test set, we use Friends.

Our evaluation metrics are micro-averaged F-score, Precision, and Recall. When performing cross-validation, we average the results across the folds.

4.4.2 Baselines

Our baselines are the following:

Majority: Set the prediction to the majority class (non-sarcastic).

Random: Uniformly sample the binary prediction.

SVM: Our main baseline uses Support Vector Machines (SVM) [40]. According to [22], SVMs are robust predictors in low-resource regimes, sometimes outperforming neural networks. We leverage scikit-learn [175] for its implementation, with its default kernel setting. The only hyper-parameter we tune per experiment is the penalty term C (1, 10, 30, 500, and 1000). We employ standardized features for the speaker-dependent setup. We concatenate the features from the different modalities.

4.5 Multimodal Sarcasm Classification

Table 4.1 shows the classification results for binary sarcasm prediction in the speaker-dependent setup. The Majority baseline performs worst (33.3% F-Score; 0% for sarcastic and 66.7% for the non-sarcastic class). For the unimodal baselines, the visual modality performs best.

	Modality	Precision	Recall	F-Score
Majority		25.0	50.0	33.3
Random		49.5	49.5	49.8
SVM	T	65.1	64.6	64.6
	V	68.1	67.4	67.4
	A	65.9	64.6	64.6
	T+V	72.0	71.6	71.6
	T+A	66.6	66.2	66.2
	V+A	66.2	65.7	65.7
	T+V+A	71.9	71.4	71.5
$\Delta_{multi-unimodal}$		$\uparrow 3.9\%$	$\uparrow 4.2\%$	$\uparrow 4.2\%$
Error rate reduction		$\uparrow 12.2\%$	$\uparrow 12.9\%$	$\uparrow 12.9\%$

Table 4.1: Speaker-dependent setup. We conduct a five-fold cross-validation with micro-averaged metrics.

The best results are achieved by concatenating the textual and visual features, surpassing all unimodal baselines with a 12.9% reduction in relative error rate. Interestingly, the baseline that uses all modalities performs slightly worse than the previously mentioned one.

We conduct error analysis for the utterances that the best unimodal baseline fails to predict correctly while the best multimodal model succeeds. We find that the textual component does not reveal explicit sarcasm in most of the sampled cases. Consequently, we hypothesize that a successful classification requires more signal than text.

The second part of our evaluation, which evaluates the speaker-independent setup, is more challenging since the model can no longer rely on speaker-distinctive features. The classification model needs to be able to generalize to new speakers. Furthermore, this is also a source-independent setting, given that a different show (Friends) is in the test set, which requires the model to generalize beyond the speaker. We consider this setup a challenging benchmark for future research in multimodal sarcasm. We also noticed the task’s increased difficulty during the SVM baseline training, which required a small error margin (a higher C value) for the best performance.

We present the results of the speaker-independent setup in Table 4.2. Unlike the previous setup, the multimodal and the unimodal models have a smaller gap. In this scenario, the audio channel is more meaningful and narrowly improves when we include text. After conducting an error analysis of the true positive examples captured by the T+A baselines but not by T, we see a higher mean pitch (mean fundamental frequency) regarding those incorrectly predicted, as suggested by [10]. We observe particular patterns of high pitch for the failure cases as well, but on average, they have a typical pitch, which is a scenario also considered by

	Modality	Precision	Recall	F-Score
Majority		32.8	57.3	41.7
Random		51.1	50.2	50.4
SVM	T	60.9	59.6	59.8
	V	54.9	53.4	53.6
	A	65.1	62.6	62.7
	T+V	62.2	61.5	61.7
	T+A	64.7	62.9	63.1
	V+A	64.1	61.8	61.9
	T+V+A	64.3	62.6	62.8
$\Delta_{multi-unimodal}$		$\downarrow 0.4\%$	$\uparrow 0.3\%$	$\uparrow 0.4\%$
Error rate reduction		$\downarrow 1.1\%$	$\uparrow 0.8\%$	$\uparrow 1.1\%$

Table 4.2: Speaker-independent setup.

the same authors. We encourage future work to focus on analyzing temporal pitch patterns.

Unlike the previous setup, the video modality does not seem to be quite helpful. We believe this is the case since the visual features represent low-level object features (far from high-level sarcasm characteristics). These features may make the classification recognize biases that do not allow it to generalize. Figure 4.2 supports this as evidence, which we describe in the next section. By looking at the incorrect predictions by the best model, we infer that models should better capture the mismatches between the main speaker’s facial expressions and the emotions of what is being said.

4.5.1 The Role of Context and Speaker Information

We investigate whether additional information, such as an utterance’s context (i.e., the preceding utterances) and speaker identification, helps with predictions. Context features are generated by averaging the representations of the utterances (as per Section 4.3) present in the context. We represent the training fold speakers with a one-hot encoding vector.

Table 4.3 shows the results for both evaluation settings for the textual baseline and the best multimodal variant. For the context features, the best variant of the speaker-independent setup (text plus audio) shows a slight improvement; however, other models have no improvement. A possible reason could be losing temporal information when pooling across the conversation.

For the speaker features, we see an improvement in the speaker-dependent setup for the textual modality. Due to the speaker overlap across splits, the model can leverage speaker regularities for sarcastic tendencies. However, we observe a different trend for the best

Setup	Features	Precision	Recall	F-Score
Speaker Dependent	T	65.1	64.6	64.6
	+ context	65.5	65.1	65.0
	+ speaker	67.7	67.2	67.3
	Best (T + V)	72.0	71.6	71.8
	+ context	71.9	71.4	71.5
	+ speaker	72.1	71.7	71.8
Speaker Independent	T	60.9	59.6	59.8
	+ context	57.9	54.5	54.1
	+ speaker	60.7	60.7	60.7
	Best (T + A)	64.7	62.9	63.1
	+ context	65.2	62.9	63.0
	+ speaker	64.7	62.9	63.1

Table 4.3: Role of context and utterance’s speaker. Note: T=text, A=audio, V=video.

multimodal variant (text + video), where the score barely improves. To understand this result, we visualize the correct predictions made by this model. As seen in Fig. 4.2, the results show a correlation between the class distributions among the overall ground truth and the correctly predicted instances per speaker. As this model does not use speaker information, this correlation indicates that the multimodal variant can learn speaker-specific information transitively through the input features, rendering additional speaker input redundant. Lastly, in the speaker-independent setup, the speaker information does not lead to improvement. This finding is also expected as there is no speaker overlap between the splits.

4.6 Conclusions

In this chapter, we study multimodal sarcasm classification. We present a new multi-source dataset and benchmark, MUsTARD, containing videos annotated with their binary sarcastic nature. This asset allows future research in this area. Through multiple evaluations, we show the importance of multimodality for sarcasm detection. We develop models that leverage text, speech, and visual signals. In addition, we studied the importance of speaker and context.

Our experiments support the premise that multimodality is essential for understanding sarcasm. The multimodal models considerably surpass the unimodal ones several times, reducing the relative error rate by up to 12.9%. We consider the assets from this chapter to be essential for future research avenues in multimodal sarcasm classification.

Up to this point, we have addressed aspects of evaluation for new domains or behaviors. In the next chapter, we propose a new evaluation procedure for already existing tasks.

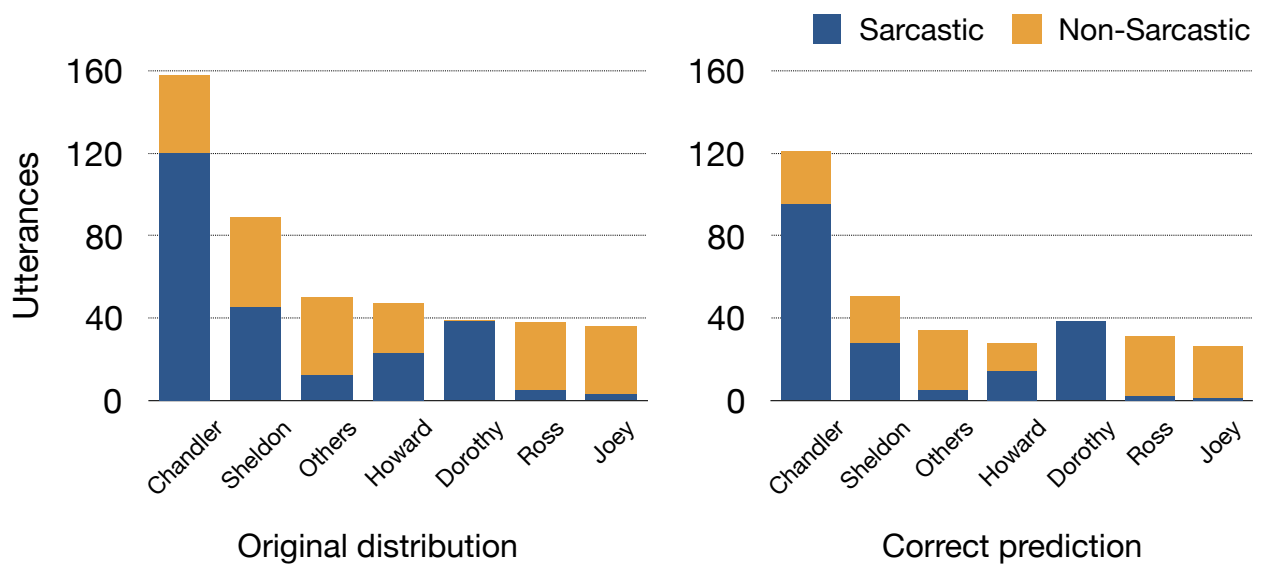


Figure 4.2: Correlation in speaker-specific sarcastic tendencies of the top-7 speakers. Predictions are obtained from the best performing model from Table 4.1. Speaker identifier features are not used.

CHAPTER 5

Realistic and Robust Video Understanding Evaluation

5.1 Introduction

Despite current progress in multimodal (textual and visual) representations, *language-informed video understanding* is still a very challenging task for machine learning systems [274, 134]. This issue is mainly due to the task setup and the dataset construction. Current video understanding datasets often have at least one of two significant limitations. First, they have limited application value. For example, multiple-choice questions [127, 225, 99, 26] do not reflect real-world tasks. Second, they are based on subjective evaluation metrics, e.g., video captioning [229, 117, 280, 241]), and are therefore hard to evaluate automatically, as the ground truth can be expressed in different ways. In this chapter, we address these limitations by introducing a new dataset named FIBER that collects multiple perspectives on the same video, focusing on noun phrases as a proxy for different entities and their interactions in the video. Our data focuses on recall and tests the ability of models to capture a wide range of possible interpretations for a particular aspect of a video.

We construct the FIBER dataset by systematically blanking captions from an existing video captioning dataset named VaTeX [241] and providing additional correct answers for the blanks. VaTeX is a video captioning dataset that contains 40,000 10-second YouTube videos with 10 English captions per video.¹ We build our video fill-in-the-blanks dataset by blanking random noun phrases from one of the English captions for each video from a subset of VaTeX consisting of 28,000 videos. Through extensive analyses, we show that the blanked noun phrases are essential for understanding critical visual aspects of the video.

We propose a Transformer-based [233] multimodal model to address the fill-in-the-blanks task. Our experiments show that our best multimodal model achieves a token-level F1 score

¹Licensed under Creative Commons, more information here: <https://eric-xw.github.io/vatex-website/index.html>.



Two children throw _____ at each other as a video is captured in slow motion.

Correct answers: balloons, balloons filled with water, balloons of water, pink balloon, pink water balloon, things, water, water balloons, water-filled balloons



_____ sits at a drum set and practices playing the drums.

Correct answers: child, drummer, future drummer, girl, kid, little girl, little kid, musician, small child, young girl



A boy is trying to comb his hair while _____ dries it.

Correct answers: another person, friend, girl, his sister, his sister with hairdryer, person, young woman

Table 5.1: Three examples from the FIBER dataset, each including three video frames, the caption, the blanked answers from the original caption together with the collected answers (all answers normalized, see Section 5.3.2).

of 71.4 while the F1 score of crowd workers is 82.5, indicating that this task is challenging for video and text understanding.

The contribution of this work is threefold:

1. We propose a novel fill-in-the-blanks task as an evaluation framework that addresses the drawbacks of previous video understanding approaches. In support of this framework, we introduce FIBER, a novel dataset of 28,000 videos and fill-in-the-blank captions with multiple correct answers.
2. We propose several unimodal baselines and two multimodal models for solving this task.
3. We provide a detailed analysis of the data to measure the diversity and complexity of the answers and also conduct an error analysis of the models' performance to gain insights into the blanked captions and videos that are hard for the models to solve.

5.2 Related Work

Language-informed video understanding is a complex task extensively addressed in the multimodal (natural language and computer vision) machine learning research through diverse tasks and benchmarks.

Multiple-Choice Video Understanding. Multiple-choice benchmarks consist of identifying the only correct answer from a set of distractors, where the set of possible answers varies depending on the input. Video Question Answering (Video QA), a popular format, consists

of answering questions based on video content. Numerous multiple-choice Video Understanding benchmarks have been proposed such as TVQA [127], MovieQA [225], TGIF-QA [99] (Repetition Action and State Transition tasks), LifeQA [26], PororoQA [112], MarioQA [163], VCQA [281], VideoMCC [229], and ActivityNet-QA [263]. However, they provide choices and are thus easier to solve than generating arbitrary text. A further drawback is that the performance without the visual input is generally already high as models can exploit biases in the dataset [3] or count on other modalities that overlap functionality with the visual one.

Video Captioning. Video Captioning consists of generating text that describes a given video. This task can be carried out using multiple datasets such as ActivityNet Captions [117] (also features Dense-Captioning), YFCC100M [227], [6], DiDeMo [7], MSR-VTT [251], YouCook2 [280], How2 [199], HowTo100M [158], VaTeX [241], TGIF [138], MovieNet [91], LSMDC [195], TGIF-QA [99] (Frame QA task). Due to the diversity of captions provided, Video Captioning benchmarks do not present a high human agreement and are thus hard to evaluate automatically with certainty [1].

Video Understanding Based on Filling Blanks. VideoBERT [220], CBT [219], UniVL [148], ActBERT [282], and HERO [133] methods propose masking random parts of the input from text and video pairs for training. However, they do this only for system training and not use the framework to test and evaluate video understanding. The only exception is MovieFIB [151], which employs a video fill-in-the-blanks scheme based on LSMDC [195] for training and evaluation. However, these methods have several drawbacks. They blank a single word, which makes it easier to guess; they evaluate correctness with a single ground-truth answer per caption; and they focus on the movies domain (we focus on YouTube videos).

Concurrent Work. The most similar work to ours is VidQAP [197], which presents an evaluation framework to fill in blanks with phrases using semantic roles based on ActivityNet Captions [117] and Charades [208]; unlike this existing work, we design our benchmark to feature a high human accuracy (avoiding ActivityNet Captions as it is contextualized, collecting multiple correct answers, and showing a high human performance). Our work is also close to [254] on evaluating the use of free-form QA; however, they employ a small vocabulary and no human accuracy that serves as an upper bound for the task.

The novelty of our work lies in our use of a challenging task (a considerable gap between human and best model performance) that measures a form of video understanding while at the same time yielding a high human performance due to the large number of possible correct

answers we collected (~ 13 per caption) from multiple annotators (~ 9 per caption).

5.3 Video Fill-in-the-Blanks Dataset

We construct FIBER – a large video understanding dataset that can evaluate the ability of a model to interpret and use a multimodal context by requiring the models to “fill in” (generate) a “blank” (a missing constituent) in this context. We build FIBER by following two main steps: (1) data generation, where we compile a large set of video-caption pairs with selectively blanked words, and (2) data annotation, where crowd workers provide additional valid answers for these blanks.

Note that we could also develop a fill-in-the-blanks dataset by completing only the first step: the data generation. However, this would result in only one valid answer (the original blanked word or phrase), which can lead to unfair evaluations that are too strict because of alternative correct answers being dismissed (e.g., “child” provided as an answer where the blanked word was “kid”). Besides manual annotations, we found no high-quality method to obtain additional correct answers automatically. For example, “building” and “t-shirt” in Table B.3 are too dissimilar, but both are correct, “pink” and “yellow” in Table 5.1 are semantically close, but only one is correct.

5.3.1 Data Generation

The dataset is constructed starting with the VaTeX [241] dataset. VaTeX is a multilingual video captioning dataset consisting of over 41,250 video clips, each taken from a unique public YouTube video lasting around 10 seconds. Each video clip has 10 English and 10 Chinese captions associated with it.

We produce blanked captions by blanking noun phrases in the English captions in VaTeX. We chose to mask only noun phrases for three main reasons. First, noun phrases often require visual information for identification or understanding. They cover a large variety of information regarding visual content, as their head nouns can describe people, objects, scenes, events, and more. A model often needs to identify the related objects in the videos and the properties of objects (e.g., color, number, or size) to fill the blank correctly.

Second, nouns are usually essential to the understanding of *visual* content and serve as reliable predictors of the ability of a system to understand a video. Other phrases, such as verbs or adjectives, can more easily be guessed from the text only while ignoring the visual information. To illustrate, consider the example “A woman _____ in the pool,” where a model can easily predict that the blank should be “swims” from the textual content only,

which would not be the case for “A woman swims in _____”, where the blank could be completed by sea, pool, lake, water, and other similar nouns.

Third, in preliminary experiments, we found that nouns lead to more robust annotations as compared to e.g., adjectives, which can have low inter-annotator agreement due to their subjectivity. As an example, consider the phrase “A _____ hill stands behind the house.” where the blank could be filled with a color property, a size property, or another attribute.

For each video, we choose the first English caption containing at least one noun phrase detected by spaCy² [87], and randomly blank one of these noun phrases to generate an instance. Accordingly, we generate our training, validation, and test data starting with the VaTeX v1.1 training set, a random subset of size 1,000 from the validation set, and a random subset of size 1,000 from the test set, respectively.

5.3.2 Data Annotation

We performed a crowdsourced annotation procedure to collect additional correct answers for each blank in the validation and test sets. As highlighted earlier, the main reason for collecting these additional annotations is to reflect the natural diversity of language and have multiple alternative answers for each blank.

We use Amazon Mechanical Turk (AMT) for the annotation. Figure 5.1 shows the annotation interface and a highlight of the data collection instructions (additional guidelines were provided, not shown here for space reasons). Workers were presented with a video clip and the corresponding masked caption for each blanked caption. They were then asked to fill in the blank with a noun phrase.³ We also asked annotators to provide answers in a confidence-descending order (the first answer should be the most natural one to the annotator).

We presented five videos for each Human Intelligence Task (HIT). Nine workers annotated each with at least two answers for each blank. We paid a bonus for each extra answer for each blanked caption, from the second one to the fifth one, to encourage them to provide more answers. We calculated a \$12 hourly rate for a worker that provides at least five answers. We estimated the time to annotate one video to be 30 seconds. Consequently, the HIT pay rate was \$0.2, which could result in a total of \$0.5 with the bonus. Additionally, we offered another type of bonus of \$0.2 to the worker with the largest number of correct answers for

²We used the model `en_core_web_trf` from spaCy v3. An error analysis identified only three tagging errors in a sample of 247 sentences.

³We blanked multi-word spans for the task, rather than single-word noun phrases, because blanking a single noun at a time led to a lower annotator agreement in preliminary experiments, likely due to the lower likelihood of overlap. For example, annotator one might write “young boy” and annotator two might write “young child”, which would have at least some overlap compared to “boy” and “child” (no overlap).



-5s Speed: ⌵ Speed: ⏩ +5s

Please watch this 10-second video while you listen to its [audio](#) (you may need to adjust your device volume).
When the video ends, **do not** click on any other video. Click the [buttons above](#) to control the video.
Please, **do not** consider the YouTube title information when filling the blank.

Fill in the blank:

The person drinks at the bar.

Your answers:

Please fill in at least 2 answers.

Figure 5.1: Annotation interface.

every HIT to encourage them to provide more than five answers.

We required workers to be in Canada or the United States,⁴ and to have completed at least 1,000 HITs on AMT with at least a 92% approval rate. The interface also checked that the answers differed for a given worker and caption. For this, we first normalized the answers by lower-casing, stripping punctuation and extra spaces, and removing the determiners “the”, “a”, and “an.”

During the annotation, we manually reviewed a sample to identify cases of incorrectly tagged noun phrases (e.g., “inside” marked as a noun when it should be a preposition) and factually incorrect noun phrases (e.g., referring to bags as “eggs” without any information on the contents of the bags); we disqualified workers who consistently provided incorrect annotations. After collecting annotations, we filtered for noun phrases using the same method as before, based on whether the text is parsed as a noun phrase (including bare nouns, e.g. “man is walking”), a wh-phrase (“who is speaking”), a simple gerund (“eating is a good way

⁴We restricted the task to these countries because it is a good proxy for proficient English speakers and because our task received lower-quality responses otherwise.

Statistic	Original phrases	Annotated
Noun phrases (before filtering)	100%	95%
Unique answers per caption	~	13.0 ± 4.14
Unique answers per caption per annotator	~	2.63 ± 0.49
Characters per token	5.09 ± 1.89	5.27 ± 2.00
Tokens	1.47 ± 0.68	1.36 ± 0.68
Visual word use (color, number, or size)	8.21%	3.31%

Table 5.2: Summary statistics for the originally blanked phrases and the annotated answers. The token counts are computed after the text normalization. The statistics for the annotated answers correspond to the ones after filtering for noun phrases (see Section 5.3.2), except for the noun phrases percentage.

to stay healthy”), or infinitive (“to eat is wonderful”).

We compute summary statistics on the annotated data to determine the degree of similarity with the initially blanked phrases. We show the statistics in Table 5.2. We find that, in general, annotators tend to provide ~ 3 unique answers for the provided data. Compared to the original phrases, annotators use about the same number of tokens. Annotators also use visual words at a much lower rate than the original phrases, possibly because the task encouraged the annotators to generate as many distinct nouns as possible without regard to descriptive information.

5.3.3 Data Analysis

To further validate the utility of the annotations collected in this study, we provide an extensive analysis of the answers (which are obtained from the union of the annotations and the initially blanked phrases).

We compute the most-frequent answers and find, as expected, that noun phrases related to “person” are the most frequent: the word “man” appears in 5.7% of total original phrases and 1.2% of total annotations (see Fig. B.1 in the Appendix). Note that our annotations have a long tail distribution, as the most frequent noun phrase appears in only 1.2% of total annotations. In addition, we find that answers related to “person”, such as “another person” are not trivial. On the contrary, in the third example in Table 5.1, for example, a model has to reason about the actions of both persons and distinguish between them. The other two examples in Table 5.1 also reflect how a model must understand both the video and the text to complete the blanks.

Figure 5.2 shows the kind of answers depicted in the videos. This analysis shows the diversity and complexity of answers a model needs to fill in, demonstrating a strong video

Statistic	%
F1 Score first answers (per caption)	82.6 \pm 15.7
Exact Match first answers (per caption)	75.3 \pm 19.7
F1 Score first answers (per answer)	70.0 \pm 11.9
Exact Match first answers (per answer)	58.1 \pm 16.3

Table 5.3: Agreement statistics for the answers for a leave-one-worker-out comparison, including the standard deviation.

understanding. As expected, the cluster *Person-related* has the most answers, followed by the clusters: *Objects* (e.g., shoes, glasses), *Places* (e.g., mountain, street), *Materials* (e.g., metal, wood), and *Body parts* (e.g., fingers, head). Note also that the *Person-related* cluster, among more typical answers such as “male” and “female”, also contains complex and diverse answers such as “dancer”, “workers”, “musician” or “audience”.

5.3.4 Human Agreement

To establish a reference for the machine models, we compute the agreement among annotators using the evaluation metrics described in Section 5.5.1, which we also use for model evaluation (Section 5.5.2).

Specifically, we apply a leave-one-out strategy to construct the “test set” and the “ground truth set.” We compare the first answer provided by each crowd worker (which is their most natural/confident answer) against the complete set of answers provided by the other crowd workers, using the maximum F1 score (token overlap) and the maximum exact match (EM) as agreement metrics, as described in Section 5.5.1.

Table 5.3 shows the inter-annotator agreement. We show the mean values of the agreement metrics per caption and answer (recall there are multiple answers per caption, so in the former case, we first average among the answers within the caption and then across the captions). Compared to the answer level, the higher rates of agreement at the caption level indicate a high amount of answer diversity among the workers.

In Fig. 5.3, we show the agreement as a function of the number of annotations per caption, leveraging a few captions that ended up with fewer than nine annotations. As the figure depicts, the agreement swiftly soars when reaching 8–9 annotators. Note the plot would always be non-decreasing as the maximum score is taken when comparing a given answer with the set of references. Future work can have the same captions annotated further to reach even higher agreement scores. We speculate that we could get a 90% F1 score with up to five more annotators per caption. Still, as the number of annotators is increased, it



Figure 5.2: The 2D t-SNE [232] representation of the clustering of the top 100 most frequent answers provided for the blanks. The answers are first converted to a singular form to avoid showing redundant information. The answers are represented using the pre-trained model `stsb-roberta-base` [144] with Sentence-BERT [191]. Each color represents a different cluster. One of the authors manually mapped the answers to the clusters.

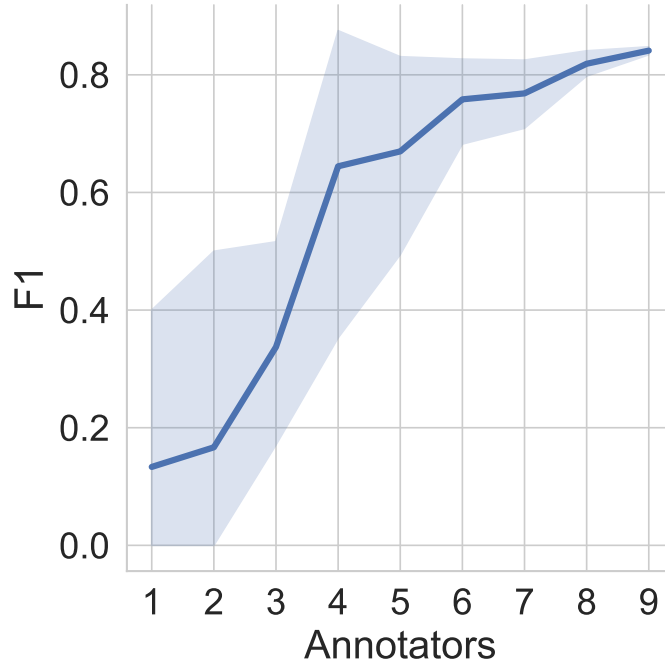


Figure 5.3: Answer agreement per caption grouped by the number of annotations of its caption. The shaded area represents 95%-confidence intervals.

would be essential to measure the number of answers that are considered wrong by other annotators as it may become non-trivial (i.e., consider not only the answer coverage but also the presence of wrong answers within the ground-truth annotations).

To further validate the quality of the crowdsourced annotations, we compare them against human annotations collected from two trusted annotators (both researchers at the University of Michigan). We sample 200 captions from the validation set, ask these two annotators to perform the same labeling task that the MTurk workers performed, and then compare their agreement with the crowdsourced data. The annotators obtain a per-caption average of 90.2% F1 score and 49.0% exact match accuracy, comparable to the workers’ agreement scores.

5.3.5 Limitations

We identify several limitations of our benchmark, which can be the objective of future work.

NPs vs. other phrases. By looking at a video and filling a blank caption with a noun phrase, it can sometimes indirectly capture other aspects, such as actions (verbs, adverbs) and object quality (adjectives, modifiers). However, this is not always the case. This is

especially true for noun phrases that are easier to guess (cf. Table 5.5).

Focus on human actions. Our data focuses primarily on human-related activities (e.g., sports) and may lack general representation available in other datasets related to animals, nature, and technology, to name a few.

Availability of the videos. Some videos may become unavailable over time since we build upon VaTeX [241] and YouTube. To mitigate this issue, the VaTeX website offers to download pre-extracted video features.⁵

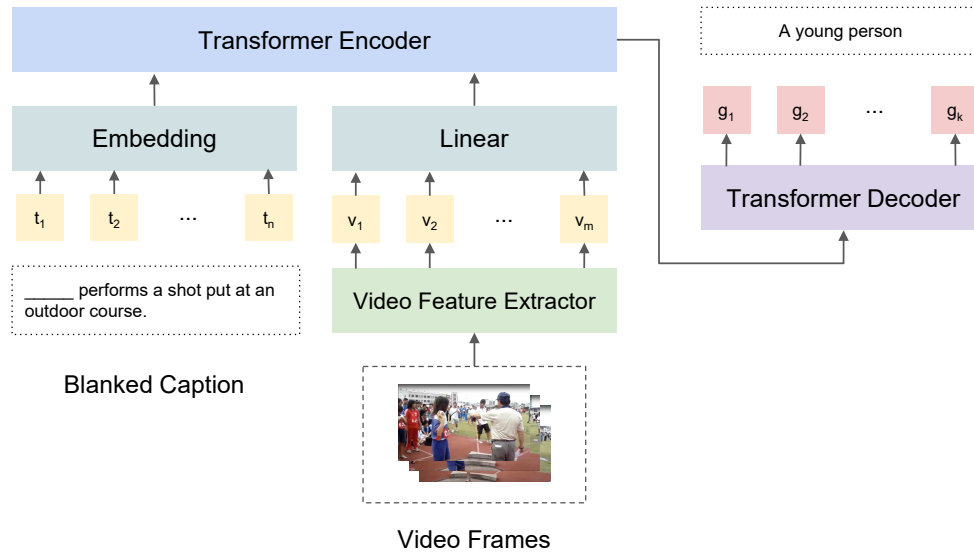
Efficiency of the data annotation process. Not all videos have multiple possible captions for noun phrases. For example, “the fork” may be the only reasonable answer for a given video and blanked caption, and annotators may not have anything else to add.

5.4 Multimodal Method for Video Fill-in-the-Blanks

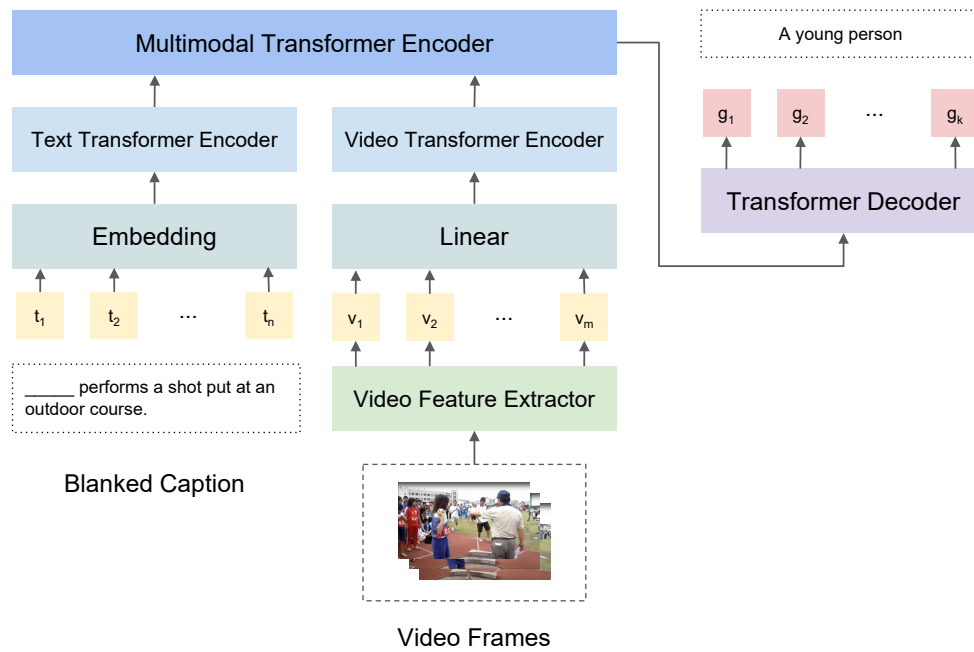
We propose an encoder-decoder multimodal method to perform the task of video fill-in-the-blanks. We first encode the text and visual modalities together to obtain a semantic representation of the blanked caption and video. The decoder uses the semantic representation to generate text corresponding only to the answer to the blank. To correctly generate an answer, a model needs to learn which parts of videos relate to the missing parts of the caption. To accomplish this, we use the original Transformer architecture [233], whose self-attention mechanism is particularly effective for encoding relations within an input sequence and has been shown to perform well in many language understanding tasks.

We consider two encoders: the early-fusion encoder and the late-fusion (two-stream) encoder. The structure of our multimodal model with an early-fusion encoder is shown in Fig. 5.4a. The input to the model consists of the tokenized blanked caption-text t_1, \dots, t_n , as well as a representation of the video consisting of multiple video sequence features v_1, \dots, v_m from a video feature extractor. An embedding layer embeds the blanked captions. The video features are projected into the encoder by a linear layer. We use a special token to represent the masked phrase and another to separate the input text and video sequences. We add positional embeddings to each input token or video feature to represent the sequence order and another embedding to indicate whether it belongs to the text or video sequence, similarly to BERT [47].

⁵<https://eric-xw.github.io/vatex-website/download.html>



(a)



(b)

Figure 5.4: (a) Early-fusion multimodal model for video fill-in-the-blanks. (b) Late-fusion multimodal model for video fill-in-the-blanks.

The late-fusion model is shown in Fig. 5.4b. The late-fusion model encodes the language and video first separately and then jointly. This decision is because the modalities may benefit from learning independently about their own context before using them together.

5.4.1 Implementation Details

For the video encoder, we use the existing I3D [24] features (size 1024 every eight consecutive frames) provided by the VaTeX dataset [241], in which videos were sampled at 25 fps. We initialize our multimodal model using T5 [184], given its ability to fill in variable-length blanks. T5 is an encoder-decoder Transformer [233] model that is a good starting point as it provides state-of-the-art performance on text-only tasks, and it was pretrained to fill arbitrary-length text spans that were previously masked. Building upon T5 allows our model to leverage a pre-trained large-scale language model with solid language abilities and fuse it with visual inputs. We initialize the early-fusion model with pretrained T5-**base** weights. For the late-fusion model, we use T5-**base** for the text encoder and the decoder. We use two one-layer transformers to encode videos and fuse text and video features, and the weights of these two transformers are randomly initialized. Following the T5 model implementation, the special token `<extra_id_0>` is used to represent the blanked phrase, and `<\s>` is used to separate the text and video sequences. The generated output follows T5 output format: the special token `<extra_id_0>` followed by the predicted text for the blanked phrase. See Appendix B.2.1 for more details.

5.4.2 Baselines

We compare our model to the following baselines.

Most Frequent Answer. The baseline uses the most frequent answer in the training set (“a man”) to answer all the blanked captions during evaluation.

Text-based Transformer. Previous visual question-answering datasets found that a text-only model can nearly match the performance of the multimodal system [8]. We conduct experiments based on text-only models to analyze how language alone can contribute to our video understanding framework. We use the off-the-shelf T5-base transformer model [184] as our baseline model. We use both a zero-shot model (not trained on our data) and a fine-tuned model. For the latter, we use the **base** model v1.1 because it performed better in our experiments on the validation set. The decoding hyperparameters are the same as in the

multimodal models, except the beam size is 8 for the zero-shot one and 2 for the fine-tuned variant, as we obtained the best validation results for each one using these beam sizes.

Single video feature. We consider using a single I3D feature per video to determine how well the model does with a small portion of the video. Based on a study of 50 randomly sampled videos, the blanked entity in the caption appeared 95% of the time in the third second of the video (see Fig. B.7 in the Appendix). For this method, we pick the I3D feature that corresponds roughly to it and apply it to the proposed multimodal methods instead of using all the video features. Note that I3D takes a window of 16 frames as input, which corresponds to 640 milliseconds, centered at the mentioned moment within the video. This can be seen as a small generalization of the Image Understanding task, which considers a single image (frame).

5.5 Experiments and Results

We perform experiments and evaluations using the dataset described in Section 5.3.

5.5.1 Evaluation Metrics

We use exact match accuracy and ROUGE-1 F1 score (token-level) [140] to evaluate the output of the generation models and to evaluate human agreement (Section 5.3.4). We count a generated text string as correct for the exact match if it has at least one string-level match among the provided annotations. For the token-level F1, we compute the token overlap (true positives) between the generated text string and each annotation, normalized by the sum of the true positives and the average of the false negatives/positives. We then compute the maximum across all annotations. For all evaluations, we computed the metrics based on the normalized text (i.e., without articles).

5.5.2 Results

We evaluate the visual understanding ability of our multimodal model by comparing its performance with the text-only baseline and human performance. The results from the fill-in-the-blanks task are shown in Table 5.4. The accuracy of the text-only model and F1 score are low, indicating that the language bias is controlled in our dataset. The multimodal model outperforms the text-only baselines in both exact match accuracy and F1 score, meaning that our multimodal model can learn video features relevant to caption language during training. We also note that the early-fusion multimodal model (T5 + I3D) slightly outperforms the

Method	val		test	
	EM	F1	EM	F1
BASELINES				
Most Frequent Answer	15.4	45.1	16.4	45.3
T5 zero-shot	39.3	52.0	37.4	49.2
T5 fine-tuned	58.0	73.8	54.5	70.9
OUR MULTIMODAL MODELS				
T5 + 1f I3D	59.2	74.7	54.3	70.5
T5 + I3D	60.2	75.0	56.2	71.4
Late-fusion T5 + 1f I3D	53.7	70.3	50.3	67.6
Late-fusion T5 + I3D	53.5	69.7	51.6	67.8
UPPER BOUND (HUMAN AGREEMENT)				
leave one worker out	75.3	82.6	75.0	82.5
new humans*	49.0	90.2	n/a	n/a

Table 5.4: Results on the validation set. EM stands for Exact Match, and F1 is the token-level F1 score (both in percentage). *1f* refers to the variant of the multimodal model with a single I3D feature. We measured the new humans’ performance from a random sample of 200. See Section 5.3.4 for more details on the human baselines.

late-fusion multimodal model, which suggests that the model learns more effectively without extra encoders (see Fig. 5.4b). The early-fusion and the late-fusion multimodal models perform worse with a single I3D feature. This suggests that the model benefits from the whole video in answering the caption correctly.

We also find a considerable performance gap between the multimodal model performance and human performance. Therefore, plenty of space exists to improve human performance, and the video fill-in-the-blanks task is worth investigating in future visual understanding research.

5.5.3 Error Analysis

Results per Semantic Label. To measure how well the model understands different patterns in the caption data, we compare the predictions generated for blanks corresponding to words of different semantic categories (the rest of the answers generally belong to the same category as the blanked words). Two authors of the paper related to this chapter annotated the initially blanked phrases for common non-overlapping semantic categories, including people, passive entities, and locations.

We list the categories and their distribution/size in Table 5.5, and we also show the

Category	Size (%)	T5 zs	T5 ft	T5 + I3D
Passive entity	40.4	52.9	63.6	63.6
Person	33.4	37.0	81.8	83.2
Pronoun	6.1	73.5	85.6	84.3
Location	5.5	55.1	74.5	75.4
Preposition	4.5	81.6	95.7	97.5
Action	3.9	47.8	65.5	59.9
Audio	2.5	56.4	73.0	63.6
Abstract	2.2	59.6	70.0	77.9
Other	1.5	56.9	75.0	83.7
Event	1.0	70.0	68.0	84.0

Table 5.5: F1 scores on the validation set for blanks with different semantic categories, in descending order based on their size. The results correspond to the best T5 zero-shot, T5 fine-tuned, and T5 + I3D models. *Person* corresponds to answers related to people, *Passive entity* represents passive entities such as objects, *Pronoun* includes subject or object pronouns, *Location* corresponds to places in general, *Preposition* includes noun phrases inside prepositional phrases (e.g., “order” in “in order to”), *Action* involves activities (“a handstand” in “perform a handstand”), *Audio* refers to noun phrases indicated through audio (“the procedure” in “the person describes the procedure”, which can only be understood through access to the audio modality), *Abstract* corresponds to high-level concepts (e.g., “a great time”), *Event* are long-running processes (“a party”), and *Other* correspond to instances hard to label for the annotators (e.g., “a video”).

performance for the best text-only zero-shot method (T5 zero-shot), text-only fine-tuned method (T5 fine-tuned), and multimodal method (T5 + I3D). The zero-shot results from T5 show that some categories can be easily predicted without fine-tuning on the dataset, namely *Preposition*, *Pronoun*, and *Event*. However, fine-tuning T5 on our dataset yields improvements for nearly all categories. The multimodal (T5 + I3D) model improves the categories of *Person* and *Abstract* nouns but performs worse for others, namely *Audio* and *Action*. This finding follows that understanding higher-order audio and visual concepts requires complex reasoning, for which the video-aware model may need more training. In general, *Action* and *Passive entity* will likely require extra attention in future work, considering the comparatively low performance for these categories.

Best Model vs. Human Performance. To gain insights on improving our models for future work, we measure where our best model (T5 + I3D) fails and humans perform well. We find three main types of wrong predictions. The most common error is predicting “man” instead of “woman”, followed by predicting “person” instead of “child” or “baby”. The majority of the remaining errors are predictions close to the ground truth answers such as “dance” instead of “exercise”, “pillow” instead of “sheets”, “rug” instead of “sand”, “floor” instead of

“court”, “knife” instead of “spatula” or “basketball game” instead of “wrestling”.

Based on these types of errors, in future work, the model would benefit from pre-training on unbiased data (both gender and age) and also from pre-training on a large-scale multimodal (language and video) dataset to learn about more diverse situations and objects.

5.6 Conclusions

This chapter introduced the fill-in-the-blank evaluation framework for video understanding. The framework addresses drawbacks of alternative video understanding tasks, such as multiple-choice visual question answering or video captioning.

We make three noteworthy contributions. First, we introduced FIBER, which is a large dataset consisting of 28,000 videos and tests based on filling in blanks, building upon an existing video captioning dataset with a new set of manual annotations, and using a modified annotation framework to encourage diverse responses among annotators. Others can easily replicate this process to create new fill-in-the-blank data for other datasets and tasks. Second, we conducted extensive analyses of the dataset to evaluate the quality of the annotations and to understand the patterns and limitations of the data. Finally, we introduced a multimodal model that fuses language and visual information and found that the video-aware models significantly outperform the text-only models. Notably, we found a consistent gap between model performance and human performance, which suggests room for improvement in future models addressing video understanding through the lens of the fill-in-the-blanks task. The FIBER dataset and our code are available at <https://lit.eecs.umich.edu/fiber/>.

In this chapter, we employed a pre-trained video encoder. Yet, more recent work has shown that an image encoder can outperform it [183]. Can we leverage it and even improve upon it? We address this question in the next chapter.

CHAPTER 6

Practical and Scalable Video Understanding

6.1 Introduction

Imagine it is winter season, and we aim to develop an auto-tagging system that recognizes all the activities in our winter vacation footage. Luckily, there have been tremendous advances in the action recognition community [237, 24, 9]. For instance, we could leverage one existing model that recognizes up to 700 human actions [115]. Sadly, our family’s favorite activity, sledding, is not on the list of categories that these models can recognize. To train a new model, we must collect many sledding examples in a traditional supervised setting. Such a process is labor-intensive, costly to create, and difficult to scale to recognize further new activities. Instead, zero-shot models [122, 210, 18] can alleviate such a burden by enabling recognition of unseen concepts.

Large pre-trained image-text models, such as CLIP [183] and ALIGN [100], have shown outstanding zero-shot capabilities on a handful of visual tasks, including video tasks such as Action Recognition and Text-to-Video Retrieval. Such models have overcome the limitations of traditional zero-shot learning algorithms by using abundant images (on the internet) with (free) natural language supervision. Despite their remarkable zero-shot performance in video tasks, there is room for improvement to close the image-to-video domain gap. For instance, recent studies have shown that fine-tuning CLIP yields significant improvements in target video tasks [149, 239]. Unfortunately, fine-tuning and improving performance in a target dataset comes with a cost: harshly penalizing the model’s zero-shot capabilities [247].

There have been multiple efforts to train video-language models that can be employed for various downstream video understanding tasks. Even though these approaches use video data, their zero-shot capabilities remain poor compared to those exhibited by CLIP [183]. It would be unfair not to mention that video-language pretraining methods train with clean yet two orders of magnitude smaller datasets [11] or large datasets with unaligned natural language supervision [158]. The alternative is to scale up further the amount of unaligned

natural language supervision abundant on internet videos. In comparison, ALIGN [100] (in the image space) has shown the ability to cope with noisy supervision by scaling up to the billion-samples scale. However, replicating such experiments with video data would only be possible for selected (if any) industrial players.

This work introduces FitCLIP, a fine-tuning strategy to adapt large-scale image-text pre-trained models for zero-shot video understanding tasks. The goal of FitCLIP is to retain the knowledge of CLIP [183] while gently adapting and learning how video data looks. Our method leverages relatively small labeled and extensive pseudo-labeled video data to train a student network. To validate FitCLIP’s effectiveness, we designed and set zero-shot benchmarks for two popular video understanding tasks: action recognition and text-to-video retrieval. Our experiments empirically validate the effectiveness of distillation to train better and fine-tune multimodal video models and show that FitCLIP establishes a new state-of-the-art for zero-shot video recognition and retrieval. Our design strategically incorporates model patching, which has not been explored before, as far as we know.

Contributions. Our key idea is to develop a method to refine large-scale pretrained image-language models to zero-shot video use cases. Our work brings two contributions:

1. We introduce FitCLIP, a refinement strategy and model for zero-shot video understanding. The model leverages abundant knowledge in large-scale image models and a distillation strategy to learn *new* video knowledge. We describe FitCLIP in (Section 6.3).
2. We evaluate FitCLIP and competitive baselines in a newly designed zero-shot benchmark (Section 6.4). Our experiments include results for two sets of video understanding tasks, action recognition, and text-to-video retrieval, where we show the value of FitCLIP (Section 6.5).

6.2 Related Work

Zero-shot Video Understanding. Multiple zero-shot methods have been proposed to tackle popular tasks such as action recognition [18, 31], text-to-video retrieval [250], and localization-related tasks [98, 270]. Most of the zero-shot action recognition literature either follows an attribute-based approach or leverages word embedding to transfer knowledge [142, 98, 62, 65, 154, 18, 31]. Differently, in the text-to-video retrieval task, zero-shot methods leverage large-scale natural language supervision to pre-train video-language models. After pretraining, these models can be employed and tested in text-to-video retrieval tasks. Similar to [11, 250], our work leverages natural language supervision from video titles to unlock

zero-shot capabilities. However, we focus on adapting well-trained image-text models to videos rather than learning a video-language model from scratch.

One of our goals is to establish a benchmark for zero-shot action recognition and text-to-video retrieval. Previous efforts have devoted insightful analyses to creating *true* zero-shot evaluation for action recognition [75]. These efforts are valuable for the traditional zero-shot setting, where methods use a close vocabulary of (seen) actions. Still, they do not fit when zero-shot models learn with natural language supervision. Instead, we follow standard (full) tests on popular action recognition datasets and well-established text-to-video retrieval datasets.

Visual-Language Pretraining. Pretraining visual models with natural language became a popular learning strategy in the image domain [161, 215, 104, 46, 183]. The idea of matching images with text dates back to the late 90s when Mori et al. trained models to predict nouns and adjectives from image-text pairs [161]. Others modernized this idea using large-scale datasets to train CNNs [104]. However, only recently, Radford et al. took this idea to the next level [183]. They trained CLIP, a dual image-text encoder, with more than 400M images and text descriptions using a contrastive objective [168]. Our work builds upon CLIP and adapts it to video use cases while preserving its zero-shot capabilities.

Video-language pretraining also gained traction in the video space. Despite the progress, it has been hard for video-language methods to compete in zero-shot settings with image-language pre-trained models. We argue this is due to the limited availability of videos with clean (and aligned) natural language supervision. For instance, Frozen in Time [11] trains a transformer-based architecture on the WebVid dataset, which contains 2.5M humanly curated video-title pairs. The dataset is at least two orders of magnitude smaller than the dataset to pre-train CLIP [183]. The importance of large and diverse data emerges when we compare Frozen in Time with CLIP in zero-shot video tasks. Others [158, 157] have trained with the relatively larger HowTo100M dataset, which contains 100M unaligned video-text pairs. Still, the zero-shot capabilities of these models remain subpar to what CLIP can provide. Our approach, FitCLIP, leverages the WebVid [11] dataset as a rich source to adapt CLIP for zero-shot video understanding tasks.

Refining Large-scale Image Models. DistInit [72] explored distilling image models for video. More recently, CLIP’s strong visual representation inspired multiple researchers to explore its usage for video tasks [182, 149, 34, 239, 183, 56]. CLIP4Clip, for instance, proposed a straightforward strategy to fine-tune CLIP for the text-to-video retrieval task [149]. Surprisingly, their simple method sets a new state-of-the-art in various datasets. Similarly,

ActionCLIP introduced a novel action-recognition paradigm, harnessing CLIP’s general visual knowledge [239]. While existing approaches effectively boost performance on target datasets and tasks, they have not been shown to preserve the original CLIP zero-shot capabilities (also based on early experiments we ran).

6.3 Method: FitCLIP

Our goal is to train a model that *expands and complements* large image-language models [183, 100] for zero-shot video (see Fig. 6.1). To do so, we introduce FitCLIP, a refinement strategy that leverages small labeled and extensive pseudo-labeled data and existing knowledge acquired from large image-text pairs. FitCLIP includes two steps. The first step trains a model in a Teacher-Student fashion, leveraging both labeled video-text pairs and pseudo-labels generated by a teacher model. The second step fuses the existing knowledge of the teacher, a large-scale pre-trained image-language model, with the student trained on video data. We call the resulting model the same as our refinement strategy, FitCLIP.

6.3.1 Teacher-Student Fine-tuning

We aim to train a model using video-text pairs while leveraging knowledge from image-language representations. One alternative is to reuse image-language encoder weights and fine-tune them in a target dataset [149, 239]. Such an approach is effective in boosting performance for in-distribution datasets. Still, it tends to fail at preserving the zero-shot capabilities of the original model’s weights due to catastrophic forgetting [59]. Instead, we gently refine the original image-language model’s weights by incorporating a two-fold strategy. We use a small sample of labeled data to avoid model drift [196] (because of using a much smaller batch size and less diverse dataset). We also regularize the learning process by adding pseudo-labels generated with the original image-language model. Note that our strategy shares intuitions with the Knowledge Distillation literature [86], where a Teacher-Student analogy is used to describe the process of training a Student with priors derived from a robust Teacher model. Figure 6.1 (step 1) illustrates the process to train our Student model.

Data Subsets. Our fine-tuning strategy relies upon two subsets of data: a small labeled dataset of video-text pairs and an unlabeled (unaligned) set of video-text candidate pairs. The labeled subset contains videos matched with one text describing their visual content. These video-text pairs are of high quality and made by a human. The unlabeled subset also includes a list of videos and a list of text descriptions. However, the match between a video

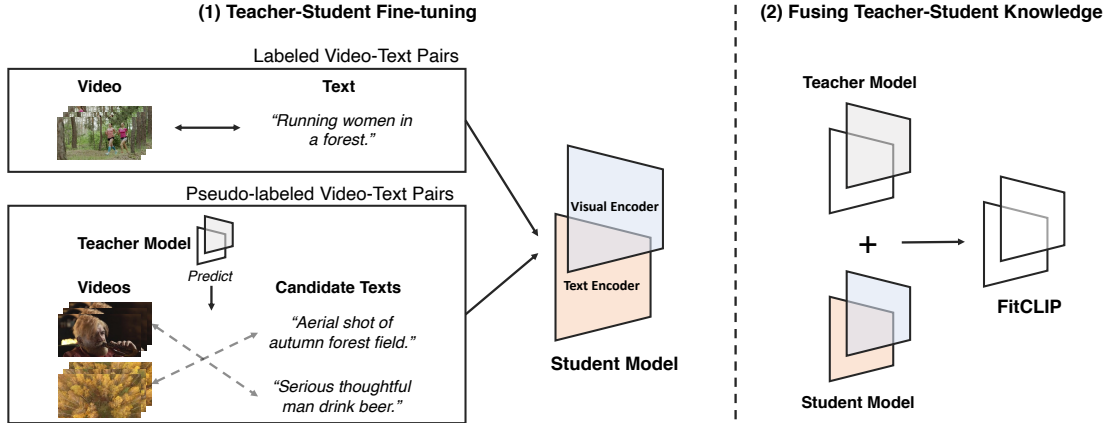


Figure 6.1: **FitCLIP refinement strategy and model.** We propose a refinement strategy to adapt large-scale image-text pretrained models. Our strategy first trains a model in a Teacher-Student fashion. To do so, we leverage labeled and pseudo-labeled (with a teacher) video-text pairs. This process, which we call step (1), yields a Student model that captures video-relevant knowledge while being compatible with the teacher. In step (2), similar to [247], we combine the Teacher and Student weights to create our final model, FitCLIP.

and the best-describing text does not exist in this subset.

Teacher Model. The teacher aims to provide *soft* pseudo-labels on unlabeled sets of video-text candidate pairs. We adopt CLIP [183] as a teacher. CLIP includes an image encoder and a text encoder, which were trained to predict the correct pairing of image-text pairs using a contrastive objective [168]. In practice, we use CLIP to compute the similarity between a subset of videos (within a sizable unlabeled set) and a set of candidate texts. Given that CLIP only takes individual images as input, we pass N frames from the video through its visual encoder and mean-pool the outputs into a single visual feature. We then use these similarity scores as target soft pseudo-labels.

Student Model. We aim to train a student model that learns from video-text pairs and distills knowledge from large pretrained image-language models. As the student, we choose the same dual architecture proposed by CLIP [183]. To train the model, we leverage two types of supervision: samples from the manually labeled video-text pairs dataset and soft pseudo-labels from the unlabeled set. Like the Teacher model, the student’s visual stream takes N frames from each video and mean-pools the resulting representations into a single feature.

Student’s Training Objective. We train the student model with two losses: a loss to learn from labeled samples and a loss to distill the teacher’s knowledge via pseudo-labels. Given a video-text pair denoted (v, t) , our student’s dual encoder extracts a video representation z_v and a text representation z_t . We use the InfoNCE [168] loss for labeled samples to learn a video-text correspondence. We follow [11, 250] and minimize the text-to-video and video-to-text contrastive losses:

$$\mathcal{L}_{v2t} = \sum_{(v,t) \in B_l} \log \frac{e^{z_v \cdot z_t^+ / \sigma}}{\sum_{z \in \{z_t^+, z_t^-\}} e^{z_v \cdot z / \sigma}} \quad (6.1)$$

$$\mathcal{L}_{t2v} = \sum_{(v,t) \in B_l} \log \frac{e^{z_t \cdot z_v^+ / \sigma}}{\sum_{z \in \{z_v^+, z_v^-\}} e^{z_t \cdot z / \sigma}} \quad (6.2)$$

Where σ is the temperature hyper-parameter, B_l is a batch of video-text pairs, z_t^+ is the positive text for the candidate video z_v , z_v^+ the positive video for candidate text z_t , and $\{z_v^-, z_t^-\}$ the negatives sets to contrast the candidate video and text representations. Then $(\mathcal{L}_{v2t} + \mathcal{L}_{t2v})$ is the final labeled (contrastive) loss.

To distill knowledge from soft pseudo-labels generated by the teacher, we use the teacher’s predictions as pseudo-labels [86] and minimize the cross-entropy of the student’s scores relative to those from the teacher:

$$\mathcal{L}_{distill,v2t} = \sum_{(v,t) \in B_l} \frac{e^{x_v \cdot x_t / \sigma}}{\sum_{x \in T} e^{x_v \cdot x / \sigma}} \log \frac{e^{z_v \cdot z_t / \sigma}}{\sum_{z \in T} e^{z_v \cdot z / \sigma}} \quad (6.3)$$

$$\mathcal{L}_{distill,t2v} = \sum_{(v,t) \in B_l} \frac{e^{x_v \cdot x_t / \sigma}}{\sum_{x \in V} e^{x \cdot x_t / \sigma}} \log \frac{e^{z_v \cdot z_t / \sigma}}{\sum_{z \in V} e^{z \cdot z_t / \sigma}} \quad (6.4)$$

Where x_v and x_t are the teacher’s video and text representations, and V and T are the sets of videos and texts in the batch.

Our final objective combines the contrastive and distillation losses as in Eq. (6.5). We scale the distillation loss with λ to prevent over-fitting to noisy pseudo-labels.

$$\mathcal{L} = \lambda(\mathcal{L}_{distill,v2t} + \mathcal{L}_{distill,t2v}) + (1 - \lambda)(\mathcal{L}_{v2t} + \mathcal{L}_{t2v}) \quad (6.5)$$

6.3.2 Fusing Teacher-Student Knowledge

We aimed to train a competent student compared to the teacher. However, competing with the 400M image-text pairs used to train CLIP [183] is challenging. Therefore, our goal is to fuse both the general visual knowledge encapsulated by the teacher and the video-specific

properties learned by the student. There are multiple ways to ensemble models [49]; however, given that our fine-tuning strategy gently adapts the teacher to video use cases, we can leverage elegant model patching techniques [70, 247, 94].¹ We follow the same approach in [94] to linearly combine the teacher and student weights (by α) and create our final model, FitCLIP.

6.3.3 FitCLIP’s Implementation Details

We uniformly sample $N = 4$ frames from each video, similarly to TSN [238]. The Teacher and Student models use a ViT-B/16 architecture initialized with OpenAI’s publicly released weights [183]. We empirically set $\lambda = 10^{-4}$ to smooth the training process (note the labeled and pseudo-labeled loss magnitudes may be wildly different). We consistently use $\sigma = 0.05$ as the temperature value. At training time, we randomly crop the frames to a size of 224×224 and perform random horizontal flips. We use the AdamW [146] optimizer with a learning rate equal to 3×10^{-5} . We use the same tokenizer as in CLIP [183]. We conduct our experiments using 8x A100 (40GB) GPUs. We use 4.5K labeled videos, randomly sampled from the WebVid-2.5M dataset [11], to compute the losses in Eqs. (6.1) and (6.2). The entire WebVid-2.5M dataset (which contains paired data) is used to compute the distillation losses – Eqs. (6.3) and (6.4). We choose the (labeled) validation loss in the WebVid-2.5M dataset as a criterion to select the best student models. Finally, to fuse the teacher and the student weights, we use $\alpha = 0.4$. We encourage the reader to refer to the Appendix analyses of some hyperparameter values. We wrote our code in Python using PyTorch [174] and Lightning [53].

6.4 Zero-shot Video Understanding Benchmark

6.4.1 Baselines

CLIP [183]. This model has been pre-trained with the WIT dataset [214], which contains about 400M image-text pairs. We re-implement the zero-shot inference of this baseline model. To deal with video, we encode $N = 4$ uniformly sampled frames per video and average their features to obtain the final video representation. We use the publicly released CLIP ViT-B/16 [50] model in all our experiments. Note that our CLIP adaptation is equivalent to ActionCLIP [239] (see the Appendix).

¹In an earlier version of this manuscript, I have referred to model patching [94] as weight-space ensembling [247]. These two methods are very similar, but model patching is a more accurate term here, even if this name was introduced later than the paper on which this chapter is based. See <https://github.com/mlfoundations/patching/issues/2#issuecomment-1365474483> for an explanation of their differences.

CLIP4Clip [149]. This method proposes changes on top of CLIP. In particular, they suggest something the authors call *post-pretraining* that fine-tunes CLIP on the category “Food and Entertaining” (380k videos) from the HowTo100M [158] dataset. The authors have not provided this checkpoint, so we cannot evaluate it on our benchmarks. Still, we decided to include the results they reported. Nevertheless, note the evaluation conditions are not the same to constitute a fair comparison (e.g., the authors sample more than four frames per video clip).

Frozen in Time [11] (Frozen). This model was pre-trained by leveraging video-text pairs from the WebVid dataset. It has multiple pre-trained versions, including one that leverages the well-curated CC3M image-text pairs dataset. In our (main) experiments, we use the model that trains using the WebVid-2.5M, COCO, and CC3M datasets (note that this is much less data than CLIP’s pretraining dataset). Results for other versions of Frozen in Time can be found in the Appendix.

VideoCLIP [250]. This baseline uses a Transformer [233] on top of a frozen HowTo100M-pre-trained S3D [269] video model from MIL-NCE [157] and a fine-tuned BERT [47] text model. This method trains on HowTo100M. A notable difference is that VideoCLIP samples 32 clips of size 32 frames (1024 frames) for each video, while we sample only four for each video.

VIOLET [61]. This method uses a video-language transformer trained end-to-end by masking discrete visual tokens. The authors use multiple training datasets, including CC3M and WebVid.

BridgeFormer [71] (BF). This model leverages a multimodal encoder on top of the unimodal encoders and a method that masks the main verb and nouns as a form of multiple-choice questions as a pre-text task. The authors find this method to be more sample-efficient than vanilla NCE.

6.4.2 Zero-shot Tasks and Datasets

Action Recognition. We aim to classify a video with one of C possible action classes. To do so, we form pretext language queries with predefined prompts. An illustrative example is the prompt: “a video of a person $\{c_i\}$ ”, where c_i is the i -th class out of the C candidate action categories. Given the visual representation of the target video, we compute its similarity with the language feature of each candidate action class prompt. We predict the action class

by selecting the visual-text pair with the highest similarity. We report the top-1 and top-5 accuracy. We evaluate zero-shot action recognition in two datasets:

- *Moments in Time (MiT)* [160] consists of 3-second YouTube clips that capture the dynamics of actions performed by varied subjects, including animals and humans. The dataset includes 339 categories and 33,900 validation videos.
- *UCF101* [211] contains 101 action classes. Our zero-shot experiments in this dataset aim to classify all the 1794 available test videos from split 1.

Text-to-video Retrieval. Given a text query, text-to-video retrieval aims to find a video from a collection that visually matches the text description. Given that the concept of classes does not exist in this task, previous methods [11, 149] denote experiments as zero-shot when the visual-language models are not fine-tuned on the downstream datasets. We report recall at $k = \{1, 5, 10\}$ and the median ranking (MdR) to measure performance. We evaluate zero-shot text-to-video retrieval in three datasets:

- *MSR-VTT* [251] contains video clips of up to 30 seconds paired with captions. We adopt the 1K-A test split [262], which contains 1,000 video-text pairs.
- *YouCook2* [280] comprises challenging cooking videos depicting fine-grained human actions. We test on 3305 clip-text pairs [157].
- *DiDeMo* [7] contains mostly unedited video clips from Flickr. We follow [143, 126, 11] and cast a video-paragraph retrieval problem. We evaluate on 4021 test samples.

6.5 Experimental Results

In this section, we conduct zero-shot experiments in two popular video understanding tasks and then a diagnostic analysis of FitCLIP. First, we study the performance of the zero-shot baselines described in Section 6.4.1 in the action recognition task. The second analysis summarizes the baseline performance in diverse datasets for text-to-video retrieval. We run diagnostic experiments to validate the importance of fusing teacher knowledge to a competent zero-shot model, as in [247]. Finally, we run performance analyses on FitCLIP that study per-class gains in the action recognition task and the shift in ranking distributions for the text-to-video retrieval tasks.

Method	Top 1	Top 5	Method	Top 1	Top 5
Supervised			Supervised		
VATT [5]	41.1	67.7	SMART [74]	98.6	–
Zero-shot			Zero-shot		
Frozen	14.0	31.8	Frozen	51.9	76.1
CLIP	19.9	40.3	BF [71]	51.1	–
FitCLIP	21.8	44.6	CLIP	74.5	94.3
			FitCLIP	73.3	95.3

(a) **Moments in Time (MiT)**

(b) **UCF101**

Table 6.1: **Zero-shot action recognition results.** (a) FitCLIP improves performance upon CLIP and significantly outperforms Frozen. (b) FitCLIP slightly improves upon CLIP; Frozen lags in zero-shot performance. Reported numbers in both tables are percentages and compute the top-1 and top-5 accuracy.

6.5.1 Zero-shot Action Recognition Results

We compare the zero-shot performance of FitCLIP and different baselines using two popular action recognition datasets. We describe the results and provide our analysis.

Analysis on Moments in Time. Table 6.1a summarizes the zero-shot results in the moments in time dataset. We also report VATT [5], the state-of-the-art using full supervision to establish a reference point. In this dataset, FitCLIP remarkably outperforms both baselines, CLIP and Frozen. It is noteworthy that CLIP outperforms Frozen by 11% at top-5 accuracy without seeing video data at training time. Despite CLIP’s good performance, FitCLIP further improves performance by 4.3% (top-5), setting a new state-of-the-art in this dataset. While FitCLIP achieves outstanding zero-shot results, a e.g. 44.6% top-5 accuracy, there is still an ample gap compared to approaches that leverage supervision from the target dataset.

Analysis on UCF101. Table 6.1b shows the results on the UCF101 zero-shot benchmark. FitCLIP outperforms CLIP at Top 5 accuracy and slightly underperforms at Top 1. All the findings remain consistent: a not-so-large gap between the best zero-shot and supervised approaches and Frozen underperforming compared to CLIP-based methods. We attribute FitCLIP and CLIP close performance (when looking at both top-1 and top-5) to the characteristics of UCF101, which contains a lot of common actions, including many sport-related actions. These types of actions often appear in photographs, and chances are, they are well-represented in CLIP’s training set.

Method	R@1	R@5	R@10	MdR
Supervised				
CAMoE [34]	52.9	78.5	86.5	1
Zero-shot				
VideoCLIP [250]	10.4	22.2	30.0	–
Frozen	21.3	43.6	55.9	7
VIOLET [61]	25.9	49.5	59.7	–
BF [71]	26.0	46.4	56.4	7
CLIP via [149]	30.6	54.4	64.3	4
CLIP4Clip [149]	32.0	57.0	66.9	4
CLIP	30.4	55.1	64.1	4
FitCLIP	33.8	59.8	69.4	3

(a) **MSR-VTT**

Method	R@1	R@5	R@10	MdR
Supervised				
TACo [255]	29.6	59.7	72.7	4
Zero-shot				
VideoCLIP [250]	22.7	50.4	63.1	–
Frozen	3.2	10.1	16.2	135
CLIP	5.3	14.6	20.9	94
FitCLIP	5.8	15.5	22.1	75

(b) **YouCook2**

Method	R@1	R@5	R@10	MdR
Supervised				
CAMoE [34]	43.8	71.4	79.9	2
Zero-shot				
VideoCLIP [250]	16.6	46.9	–	–
Frozen	23.2	45.8	56.8	7
VIOLET [61]	23.5	49.8	59.8	–
BF [71]	25.6	50.6	61.1	5
CLIP	26.2	49.9	60.6	5
FitCLIP	28.5	53.7	64.0	4

(c) **DiDeMo**

Table 6.2: **Zero-shot text-to-video retrieval results.** In all datasets, FitCLIP improves upon CLIP by significant margins. (a) FitCLIP shows the best zero-shot results, though there is an important gap with the supervised state of the art. (b) In this dataset, YouCook2, FitCLIP exhibits the most significant gap between fully supervised approaches and VideoCLIP, which is pretrained on HowTo100M. We attribute this result to the dataset’s fine-grained nature. (c) FitCLIP consistently boosts upon CLIP even for the DiDeMo (paragraph-retrieval) task, which includes extended language queries. R@k denotes recall at the top- $k = \{1, 5, 10\}$ predictions, and MdR refers to the Median Ranking metric.

6.5.2 Zero-shot Text-to-video Retrieval

To compare FitCLIP and the baselines, we report the experimental results and analysis for the text-to-video retrieval task.

Analysis on MSR-VTT. Table 6.2a summarizes results in the MSR-VTT dataset. We observe that Frozen performance is poor compared to CLIP’s and FitCLIP’s. Even though Frozen was trained on video data with similar properties to MSR-VTT, it is hard for this model to compete with the general knowledge encoded in CLIP-like models. We observe that FitCLIP consistently improves performance upon CLIP across all the retrieval metrics. These results suggest that FitCLIP captures complementary video-language information that CLIP lacks. Concerning the gap to reach the performance of the best-supervised approach, CAMoE [34], FitCLIP is not that far behind. Even though there is a 16.1% gap at R@10, we see that FitCLIP closely approaches supervised performance at the MdR metric.

Analysis on YouCook2. We report zero-shot results for the YouCook2 dataset in Table 6.2b. From the get-go, we observe the difficulty of this dataset. Even the state-of-the-art, TACo [255], struggles to achieve more than 30% R@1. While we observe that FitCLIP’s performance consistently outperforms other zero-shot baselines, we have observed a large overall gap between our method and those that are supervised or pretrained on HowTo100M [158]

(VideoCLIP [250] in the table). We hypothesize this is due to the fine-grained nature of the language descriptions in YouCook2 and HowTo100M. Moreover, many videos in this dataset are captured from an egocentric view.

Analysis on DiDeMo. Table 6.2c summarizes DiDeMo’s paragraph retrieval task results. First, we observe that the performance of Frozen, VIOLET [61], and BridgeFormer [71] approach the one achieved by CLIP in this dataset. Unlike other datasets, DiDeMo contains unedited, human-centric footage that shares commonalities with the WebVid dataset used to train Frozen. Conversely, FitCLIP, which leverages the knowledge from CLIP and the WebVid dataset, achieves the best overall performance. For completeness, we report the CAMoE’s supervised performance [34], which is 15.9% better than FitCLIP, the most competitive zero-shot alternative.

The results of these three datasets empirically demonstrate the value of FitCLIP in pushing the limits of zero-shot text-to-video retrieval. FitCLIP establishes a new state-of-the-art zero-shot text-to-video retrieval across three different datasets. Despite such a milestone, there is still room for improvement, especially in fine-grained datasets such as YouCook2. We hope this benchmark promotes more work on zero-shot text-to-video retrieval.

6.5.3 Diagnostic Analysis

Impact of Fusing the Teacher-Student Knowledge (Table 6.3). One of FitCLIP’s fundamental properties is its ability to incorporate student learning from video data and the CLIP teacher’s knowledge. Here, we report the performance of both our Student and Teacher (CLIP) and contrast that with the final zero-shot performance obtained with FitCLIP. Table 6.3 summarizes the results. Although the Student’s performance remains inferior to the Teacher’s, it is close enough in various datasets, e.g., MiT, MSR-VTT, and DiDeMo. Δ denotes the difference in performance between FitCLIP and the teacher and indirectly measures the contribution of the student learning. We observe that improvements are consistent across all tasks and datasets. These results suggest that the Student effectively passes complementary information to the teacher after the model patching.

Additional Ablations. We include additional analysis in the Appendix. We compare the properties of FitCLIP vs. CLIP, do a deep-dive on the impact of fusing the Teacher-Student knowledge, ablate model patching parameters, and report comparisons with additional methods trained on HowTo100M.

	Action Recognition		Text-to-video Retrieval		
	UCF101	MiT	MSR-VTT	YouCook2	DiDeMo
Teacher (CLIP)	74.5	19.9	55.1	14.6	49.9
Student	64.7	17.7	52.6	9.7	42.4
FitCLIP	73.3	21.8	59.8	15.5	53.7
Δ	$\downarrow 1.2$	$\uparrow 1.9$	$\uparrow 4.7$	$\uparrow 0.9$	$\uparrow 3.8$
Err. rate red.	$\downarrow 4.7$	$\uparrow 2.4$	$\uparrow 10.5$	$\uparrow 1.1$	$\uparrow 7.6$

Table 6.3: **Impact of fusing teacher-student knowledge.** Δ denotes the absolute difference in performance between FitCLIP and the Teacher model. We report the top-1 accuracy for the zero-shot action recognition datasets and the top-5 recall for the zero-shot text-to-video retrieval ones. Even though the Student model is weaker than the Teacher, it still provides complementary information to FitCLIP, yielding consistent improvements (Δ) across datasets. Full results with all the metrics are available in the Appendix.

6.6 Conclusions

This chapter presents a fine-tuning strategy to adapt large-scale image-text pre-trained models for zero-shot video understanding tasks, dubbed FitCLIP. FitCLIP performs well on zero-shot settings for three Text-to-Video Retrieval and two Action Recognition tasks we evaluated. We show the importance of doing the model patching step of our method to keep or improve the teacher’s robust performance across different datasets, even when the student was trained on other data. We highlight our method introduces no extra inference costs while improving CLIP results overall.

A pending question is how well these methods perform on actions that involve understanding not only the verb but also the object employed (e.g., “reading a newspaper” as opposed to just “reading”), especially those verb-object combinations that are not usually present in the training set).

CHAPTER 7

Compositional Generalization with Image-Text Models

7.1 Introduction

In the last few years, real-life video use cases have benefited enormously from large-scale pre-trained image-text alignment models since they tend to provide an incredible zero-shot performance across multiple video tasks and domains [183, 100, 149]. This success is due, to a large extent, to the fact that many short-video tasks can be successfully solved by modeling the videos as an unordered set of image frames [20, 125]). Such models have been largely influenced by CLIP [183], which is still a state-of-the-art method on many fronts, and as far as I am concerned, further work on the area has not achieved substantial improvements over it. The method employs late fusion by having an image and a text “tower.” When the image and text semantically correspond, these towers are trained to provide a high alignment (similarity) score. Two-tower models (also known as two-stream or dual-stream models) scale better for retrieval tasks than early-fusion multimodal models (often through cross-attention), as they typically employ a matrix multiplication to compute the similarity score between the image and text representations, which today’s available hardware can cheaply compute. Counting with inexpensive and fast text-to-video retrieval systems is vital for video search engines in real-life settings, which, in many cases, need to go through millions of videos for a given text query.

However, such models have fundamental issues that make them impractical in realistic use cases. An important issue is that such models cannot distinguish the object from the subject of an action [171]. Moreover, [228] found that these models are unsuccessful at grasping word order, as they tested CLIP [183] and other methods on examples with a pair of text captions with the same words between each other but in a different order and they show performance slightly above chance, with a significant gap compared to humans. This issue is aggravated when the text gets more complex, such as when involving more than one predicate.

Generally, these methods fail to compose known elements correctly; we say they suffer from *compositionality* issues. The compositionality principle [173] states that the meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined.

The compositionality issue is exacerbated when the concepts are common but the composition is rare, as in “microwaving a shoe.” This aspect can be appreciated when considering the low results models such as CLIP [183] obtained in RareAct [156], a framework built with this purpose in mind, in comparison with benchmarks that reflect standard actions and compositions such as UCF-101 [211] and Kinetics [108]. A model that demonstrates this skill would be said to have compositional generalization [110], as it would be able to manipulate known elements and compose them in uncommon ways (out-of-domain generalization since they are infrequent in the model’s training distribution).

There is no evidence that any VLM, including large-scale single-stream models such as GPT-4V [169], successfully identifies compositions. This assertion is supported by the fact that existing benchmarks that test compositionality continue to be an open challenge [228, 264, 150, 88].¹

To address these limitations, previous work has introduced techniques to increase the compositional capabilities of pre-trained VLMs, such as NegCLIP [264] and REPLACE [88]. However, such methods come at a significant cost: they sacrifice the performance on more common object-centric recognition, as measured by ImageNet [44], EuroSAT [81, 80], and CIFAR100 [119]. For instance, as shown in Fig. 7.2, NegCLIP showed an increase (compared to the pre-trained model) in its ability to address SugarCrepe [88] compositionality benchmark from 72.9% to 82.5% while, at the same time, its performance on ImageNet [44] top-1 accuracy dropped from 63.4% to 55.8%. Similarly, [88] applied REPLACE to reach a high score of 84.7% on SugarCrepe, but at the cost of a significant drop to 52.9% on its ImageNet accuracy.

In this chapter, we introduce a framework to significantly improve the ability of existing two-tower models to encode compositional language while keeping the performance on more standard benchmarks, as qualitatively illustrated in Fig. 7.1 and quantitatively evaluated in Fig. 7.2. These improvements include out-of-domain subject-verb-object compositions, which lay at the core of video understanding. Specifically, our contributions are as follows. First, we propose a scalable method of measuring VLMs’ limitations and use it to find multiple ones related to compositionality and other phenomena. Second, we show that data curation can significantly impact how a model can handle compositional knowledge. Third, we confirm that training along with hard negatives can bring additional improvements. Fourth, we show experimentally that model patching can be employed to preserve model performance

¹See Section 7.2 for details.


	CLIP	CLOVe
 <p>“A white <u>horse</u>.” vs. “A white <u>cat</u>.”</p>	✓	✓
<p>“The <u>horse</u> is eating the <u>grass</u>.” vs. “The <u>grass</u> is eating the <u>horse</u>.”</p>	✗	✓

Figure 7.1: Example of compositions recognized by CLOVE but not by CLIP. The first row shows a pair of compositions where one word varies, while the second one shows a pair that uses the same words but in a different order.

on previous tasks. Finally, we combine these ideas into a new framework called CLOVE and show that it can **significantly improve compositionality over a contrastively pre-trained VLM**. As a case study, we show how our framework can effectively improve CLIP’s compositional abilities while maintaining the performance on other tasks. Our provided checkpoints allow others to substitute their CLIP-like model weights for a version with significantly better language composition abilities.

7.2 Related Work

Benchmarking Compositionality. Several frameworks have been proposed to measure model performance on language compositionality. [207] crafted a benchmark of foil image captions generated by changing a single word from the correct captions. Models must identify if the image-caption pair correspond to each other, among other tasks. Winoground [228] carefully built a high-quality dataset of 400 examples, each consisting of two images and two captions. These two captions contain the exact word but in a different order following one of several strategies (e.g., swapping the subject and the object). Each image must match the correct caption for the models to pass this test. Models cannot simply rely on their ability to recognize concepts in images, as the elements repeat but are composed differently.

[228] found that successfully passing the Winoground benchmark requires composition skills along with many others, such as commonsense reasoning and locating tiny objects. [264] argued that Winoground is too small to draw statistically significant conclusions and built a benchmark called ARO consisting of examples with a single image, a correct caption, and

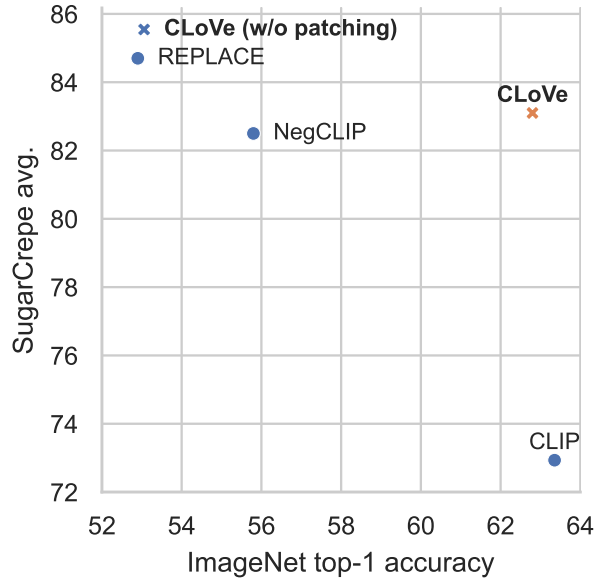


Figure 7.2: Our proposed framework CLOVE significantly improves the compositionality performance (as measured by an average of SugarCrepe’s seven fine-grained tasks) of pre-trained CLIP-like models while preserving their performance on other downstream tasks (as measured by ImageNet). We present comparisons with more benchmarks in Tables 7.2 and 7.3. Baselines: REPLACE [88] and NegCLIP [264].

multiple automatically generated incorrect captions. CREPE [150] crafted a benchmark to measure compositionality in terms of systematicity and productivity. It considers both seen and unseen compounds, among other phenomena. SugarCrepe [88] is a recent benchmark that avoids ungrammatical and nonsensical negative captions while being large. They showed it cannot be easily solved by computing the probability of the text captions without looking at the image.

In VALSE, [171] demonstrates that vision-language models have difficulty counting objects and classifying spatial relations between objects. [198, 276] show that, although state-of-the-art vision-language models can grasp color, they do not fully understand more difficult concepts such as object size and position in the image.

[82] evaluate state-of-the-art vision-language models by building SVO-Probes, a probing benchmark focused on verb understanding. They show that image–language transformers fail to distinguish fine-grained differences between images and find they are worse at understanding verbs than subjects or objects. Our work continues their proposed future work direction by analyzing model performance on fine-grained verb categories.

Other benchmarks have also been created that consider compositionality as well as other phenomena, such as RareAct [156], Cola [190], and CLEVR [103].

Methods to Improve Compositionality. Several works have shown that VLMs cannot recognize compositions successfully [207, 156, 171, 228, 82, 264, 28, 150]. For this reason, [264] proposed NegCLIP to improve how CLIP [183] composes concepts. It consists of adding hard negative texts by taking the captions from the training batch and automatically generating sentences with the exact words but in a different order. This approach makes the model distinguish between an image and the caption in the correct order compared to the exact words in an arbitrary order (as well as the other negative captions within the batch). [88] build upon NegCLIP and CREPE [150] and propose three ways to generate random negatives: REPLACE, SWAP, and NEGATE. All these methods start from a Scene Graph representation of the sentence and operate over it. REPLACE, which had the best overall results, performs single-atom replacements. SWAP exchanges two atoms within the scene graph. Finally, NEGATE introduces negation words (i.e., *no* or *not*). We build upon NegCLIP [264] and REPLACE [88] while we propose to use synthetically-generated captions to scale them up, as well as applying model patching [94] to avoid catastrophic forgetting. To our knowledge, we introduce the first approach that significantly improves the composition skills of contrastively trained models while preserving their zero-shot performance on other downstream tasks.

Cap and CapPa [231] are two recently introduced models that employ captioning instead of contrastive learning (as in CLIP) to train VLMs. [231] showed that they present an excellent performance on compositionality as measured by ARO [264] and SugarCrepe [88]. These models rely on captioning and thus on computing the probability of the text given an image, making them inefficient for retrieval and classification. For ARO, they showed that they can achieve high performance without looking at the image (they call it a “blind decoder”). For SugarCrepe, the authors did not compute this specific baseline. Hence, we cannot infer the extent to which these models handle compositions successfully. Our approach is different from theirs as it builds on top of contrastive two-tower models, which are efficient for retrieval and classification. It does not rely on computing the probability of text, which is generally unimportant for such settings as all texts are equally likely (unlike in image captioning).

7.3 Understanding the Limitations of Vision-Language Models: a Case Study on CLIP

Even when vision-language models are widely used [147, 135, 272, 183, 209], little is known about their limitations. Recent work, such as Winoground [228], SVO-Probes [82], or VALSE [171], have designed benchmark probing tasks by annotating data to follow specific

properties (i.e., object color, location, size, swapping word order, replacing words). This line of research led to valuable insights into the limitations of current state-of-the-art multi-modal models such as CLIP [183] and ViLBERT [147].

Current probing benchmarks rely on time-consuming data annotation procedures, which makes them unscalable and limited in scope. As a complementary solution, we propose a method to probe vision-language models by relying on existing data without requiring extra annotations. Our method consists of extracting a large set of candidate features from a vision-language benchmark and testing their correlation with the output of the target models on the given benchmark.

By applying our method on CLIP [183], a widely used and still state-of-the-art multi-modal model, by leveraging the SVO-Probes [82] dataset, we arrive at several results. We find that CLIP gets confused by concrete words and surprisingly improves performance for more ambiguous words while noting little change from the word frequencies. We confirm the findings of [228] of CLIP behaving like a bag of words model, and that of [171] of CLIP performing better with nouns and verbs. To our knowledge, we are the first to conduct an in-depth analysis of how language semantic properties influence CLIP’s performance.

We summarize our contributions as follows. First, we propose a scalable way of measuring the limitations of vision-language models. Second, we test our method using a state-of-the-art vision-language model (CLIP) and a popular benchmark (SVO-Probes), validate known challenges, and uncover new ones. Third, our work allows future models to focus on solving the new difficulties discovered.

7.3.1 Methodology

We employ a benchmark to measure how a vision-language model performs on various semantic concepts. We aim to quantify which concepts are the most and the least challenging for the model. We illustrate our setting in Fig. 7.3 and we separate it into three main steps.

First, we use CLIP [183] to compute scores for instances from the SVO-Probes [82] dataset and obtain two corresponding alignment scores for each sentence and its corresponding *positive* and *negative* image. Next, we extract and process various semantic features from SVO-Probes. Finally, we compute the correlation coefficients between each feature and the CLIP score. The features with the highest coefficients will represent concepts CLIP performs well on, while features with the lowest coefficients will represent challenging concepts for CLIP.

Image Caption: *Girl is standing in the grass.*

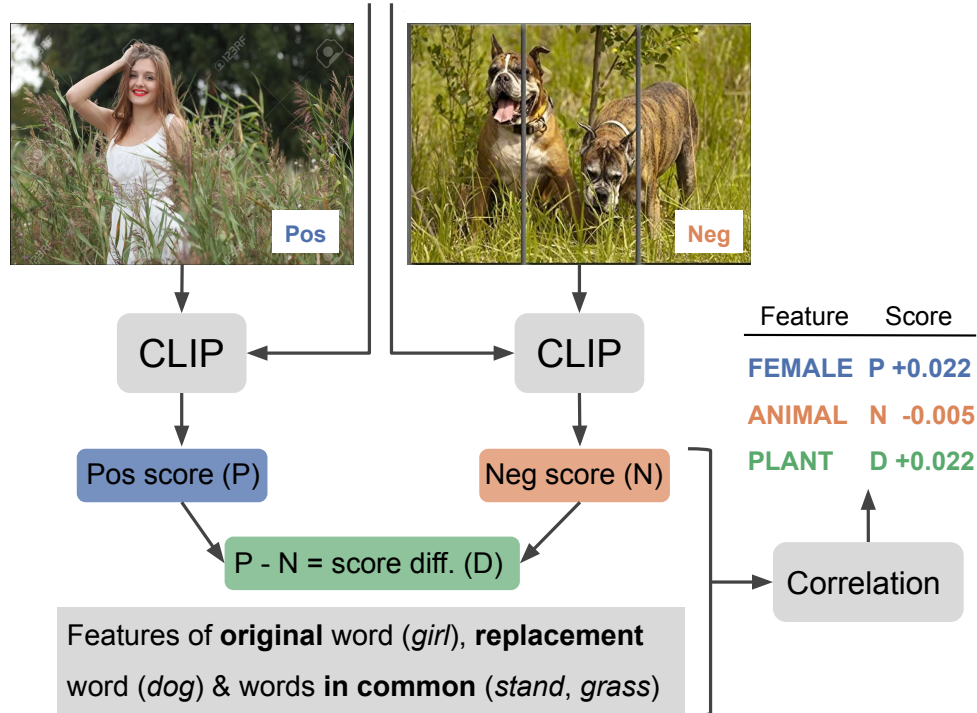


Figure 7.3: We propose a simple framework to analyze CLIP performance on SVO-Probes data. We test CLIP on the benchmark, extract a diverse set of semantic features from the data, and measure the correlation between each feature and the CLIP score (P , N , or D). Features with positive correlation (e.g., *Female*, *Plant*) positively impact the model performance, while features with negative correlation (e.g., *Animal*) negatively impact the model performance.

7.3.1.1 Dataset

We choose the SVO-Probes [82] dataset due to its design and large scale size (421 verbs and over 48,000 image-sentence pairs). The SVO-Probes benchmark was designed for probing image-text models for their understanding of **subject**, **verb**, **object** triplets. Each instance from the dataset consists of a text caption, a *positive* image that matches the caption, and a controlled (adversarial) *negative* image that shares two out of three aspects (subject, verb, and object) from the sentence but does not match the other one, as shown in Fig. 7.3. These controlled examples enable one to probe models for their understanding of verbs, subjects, and objects. The instances also include information about the negative image, such as a (hidden) associated negative caption, which we leverage in this paper.

We propose to use this dataset to evaluate the CLIP [183] model. We choose to test CLIP, as opposed to other language-vision models, due to its widely-spread use and impressive zero-shot performance on a variety of vision-language tasks (e.g., text-to-image retrieval, image question answering, human action segmentation, image-sentence alignment – [23]). Furthermore, [82] test only ViLBERT-based [147] models, which are known to perform worse than CLIP [23].

7.3.1.2 Model Output

As depicted in Fig. 7.3, we obtain three CLIP scores for each pair of *positive* and *negative* images: a *positive* score (P), computed between the caption and the *positive* image; a *negative* score (N), calculated between the caption and the *negative* image; and the *difference* between these scores ($D = P - N$).

Because the text and the positive image are aligned, P represents an absolute alignment score. In the case of the text and the negative image, even though they are similar in some ways (because of SVO-Probes’s design), they do not correspond. Thus, N represents an absolute misalignment score. D represents a relative alignment score. Ideally, CLIP should have a high P score and a low N score, and a high difference between them (a high D). We propose to pay special attention to D given that CLIP is generally used in relative comparisons, such as when using it for classification (choosing the class text that maximizes the alignment score, given an image) or when using it for retrieval (finding the text/image that maximizes the alignment score given an image/text).

7.3.1.3 Feature Extraction

We extract features from the words marked in the SVO-Probes benchmark (i.e., subject, verb, and object) for each given sentence and corresponding image in the benchmark.

If the corresponding image is *positive*, all the extracted features are from words *in common*, i.e., that appear both in the image and the text. Otherwise, if the corresponding image is *negative*, in addition to words *in common*, we also extract features from words present in the sentence and not in the image (*original* word) and words present in the image but not in the text (*replacement* word). As an example, in Fig. 7.3 the words *in common* are “sit” and “grass”, the *original* word is “girl” and the *replacement* word is “dogs”. The *original* and *replacement* words represent what is different between the image and the text, while the words *in common*, as the name suggests, represent what the picture and the text share.

We extract the following **semantic** textual features: [130] verb classes, LIWC psycholinguistic markers [176, 177], General Inquirer [216] semantic classes, WordNet hypernyms [57], word presence, semantic similarity, ambiguity, frequency, sentence length, and concreteness [19].

Levin verb classes. [130] groups verbs according to their semantic content and also according to their participation in argument alternations.

Levin’s semantic content-based taxonomy categorizes 3,024 verbs into 48 broad classes and 192 fine-grained classes.² A verb can belong to one or more classes. Some examples of verb classes are: (1) broad *change of state* (e.g., clean, divide, soak), *manner of motion* (e.g., climb, drop, run) or *social interaction* (e.g., marry, meet, hug); (2) fine-grained: “*roll*” verbs (e.g., bounce, coil, drift), “*run*” verbs (e.g., amble, bolt, race) or “*hug*” verbs (e.g., cover, encircle, touch)

LIWC psycholinguistic markers. Linguistic Inquiry and Word Count (LIWC) [176, 177] is a widely used word-counting software that includes dictionaries of English words related to human cognitive processes. Specifically, we use the LIWC2015 dictionary, which contains 6,400 words and word stems. Each word or word stem defines one or more categories: e.g., the word “mother” is assigned the categories: *female*, *family*, *social*.

General Inquirer classes. General Inquirer [216] is a resource for automatic content analysis. More specifically, it categorizes words into emotional and cognitive states and diverse semantic categories outlined in the Lasswell dictionary [165, pg. 46–53].

WordNet classes. WordNet [57] is an extensive lexical database of English words grouped into cognitive synonyms called synsets. Semantic and lexical relations interlink the synsets. The most frequent relation among synsets is the super-subordinate relation, also called *hyponymy*. It links more general synsets to specific ones: e.g., “building” is a *hypernym* of

²<https://websites.umich.edu/~jlawler/levin.verbs>

“house” and “school”. We collect all the hypernyms of the most common word synset for each given word.

Word presence. For each given word, we use a marker to indicate if the word is present or not in the sentence. Note that studying the effect of specific words does not imply that they have no dependencies with other words. Their role may change depending on the context; however, we study them in aggregate.

Sentence length. We measure each sentence’s length as the number of words in the sentence.

Semantic similarity. In the case of *negative* images, we compute the cosine similarity score between the *original* words and the corresponding *replacement* words. We compute the word representations using Sentence-Transformers [191], with the model `all-MiniLM-L6-v2`, which is based on MiniLM [240].

Concreteness score. We use a dataset of words with associated concreteness scores from [19] to measure words’ concreteness. A human annotator labels each word with a value between 1 (very abstract) and 5 (very concrete). Abstract words (e.g., “beauty”, “sadness”) denote ideas, feelings, or other intangible concepts, while concrete words (e.g., “table”, “write”) refer to objects and actions.

Ambiguity. We measure the ambiguity of a given word by counting the number of synsets in WordNet [57].

Frequency. We measure the word frequency in a subset (~ 13 M image captions) of LAION [203], a dataset representative of CLIP’s training data.

7.3.1.4 Feature Representation

The **binary** features, i.e., Levin, LIWC, General Inquirer, WordNet classes, and word presence, are represented as binary vectors, while the **numerical** features i.e., sentence length, concreteness, similarity, ambiguity, and frequency are standardized. All the features are then concatenated together.

7.3.1.5 Feature Selection

We measure the degree of correlation between each feature and the model performance. For each of the **binary** features, we compute a two-sample, two-tailed t-test [217] along with the model output score. This test evaluates if the means of the populations coming from each feature value (true or false) differ significantly. If so, we compute the difference of means as a reference value. In the case of **numerical** features, we compute the Pearson’s correlation coefficient [14] between each feature and the model performance score.

Next, we employ a one-sample, two-tailed t-test to determine if the coefficient significantly differs from zero, i.e., if there is any correlation according to this metric. We chose a p-value threshold of 0.05 (a confidence level of 95%) to filter out the features.³

7.3.1.6 Experimental Details

We use an OpenAI pre-trained CLIP [183] ViT-L/14 [50] model.

7.3.2 Results

Our main observations and takeaways from this evaluation are the following:

(1) CLIP behaves like a bag-of-words model. As shown in Fig. 7.4, the distributions of P and N highly overlap. The negative image’s adversarial nature may partly explain this as it shares many elements with the text. This finding is consistent with that of [228], that CLIP performs like a bag-of-words model.

This finding is also supported by the fact that many features from words *in common* contribute to increasing both the positive (P) and the negative scores (N): e.g., `hypernym_food.n.02` increases P by 0.042 and N by 0.050; LIWC “money” increases P by 0.036, and N by 0.032. As described in Section 7.3.1.5, we measure the importance of each feature as the difference of means between the CLIP scores when the feature is present and when it is not. We observed that many of the features for the words *in common* appeared to influence both P and N similarly, confirming this hypothesis.

(2) CLIP performs better with nouns than with verbs. When computing the number of times CLIP assigns a higher score to the similarity between the text and the *positive* image than to the similarity between the text and the *negative* image, the verbs obtain 81.45% accuracy. At the same time, the subjects get 86.87% and the objects 88.78%. The number

³See the obtained scores and p-values in the web page of the paper associated with this chapter at github.com/MichiganNLP/Scalable-VLM-Probing.

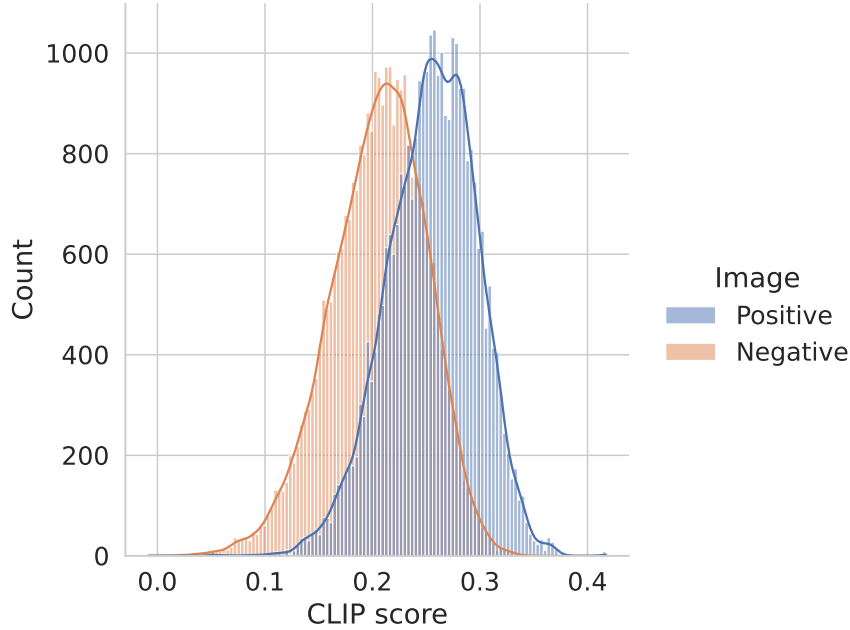


Figure 7.4: Histogram plot of the distribution of CLIP scores between the text with the positive image and the text with the negative image. We include a kernel density estimation curve to aid this visualization.

obtained for verbs is relatively close to that of a similar setting experimented by the VALSE benchmark [171], in which they reported 75.6% accuracy (also considering that we could not determine which pre-trained CLIP variant the authors evaluated). At the same time, the noun (objects and subjects) replacement numbers are consistent with those reported by the same authors (88.8%), obtained from FOIL it! [207].

(3) CLIP gets confused by concrete words. Figure 7.5 shows both the *positive* and *negative* CLIP scores improve the more concrete a word is (words from the caption represented in both the positive and the negative images). However, this figure shows that the *negative* score increases faster. This result implies that, in an image classification or image-to-text retrieval setting, CLIP will more likely consider an incorrect text valid if it has more concrete words than the correct text.

(4) CLIP prefers average-length sentences. We present in Fig. 7.6 how the caption sentence word length affects the score. CLIP presents a low performance when the sentences are very short (around three words long), improving when the sentences are longer since the difference between the *positive* and *negative* scores (D) gets larger with the sentence length.

Figure 7.7 shows how the CLIP scores are distributed for the different number of words,

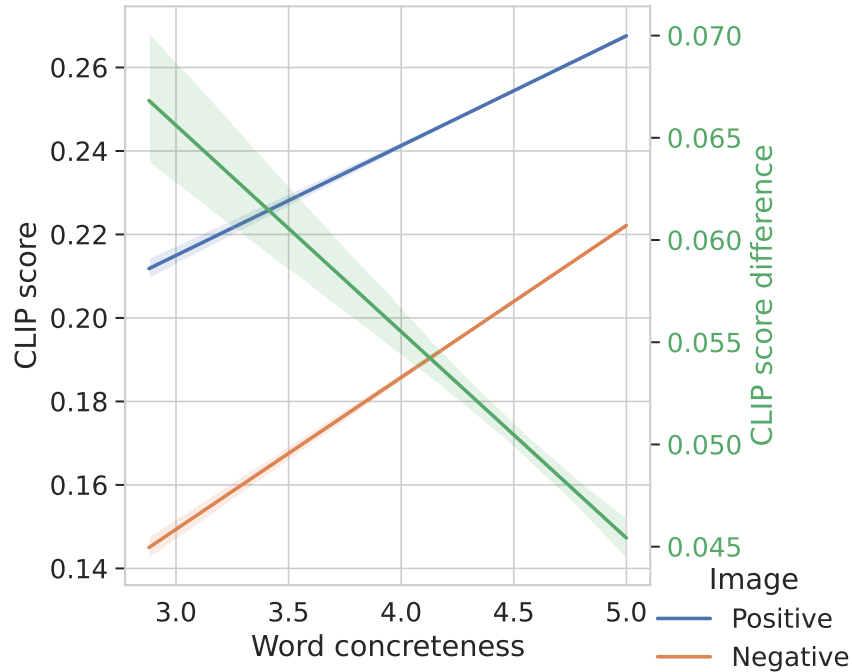


Figure 7.5: Linear regression plot of the average concreteness for the words in the sentence that are common to both images vs. the CLIP score. The shadowed areas are 95%-confidence intervals for the expected value.

showing, for example, that there is a great overlap between the similarity scores between texts of length six and a *negative* image and the similarity scores between texts of length three and a *positive* image. This finding implies CLIP is more likely to select the wrong text when comparing an image with a short correct text and one with long incorrect text.

(5) CLIP is affected by word frequency. Figure 7.8 studies the frequency effect on the score for the words that represent concepts that appear in both the *positive* and *negative* images. The higher the word frequency, the higher the CLIP score. Still, the difference in scores is barely affected.

(6) The score improves for more ambiguous words. Surprisingly, there is a larger gap in the score difference (D) when the words have more meanings associated with them (for the words that represent concepts in both the *positive* and *negative images*), as shown in Fig. 7.9. The positive score seems to remain almost constant while the negative score drops, widening the difference. The word frequency seems not to be a confounding factor based on (5).

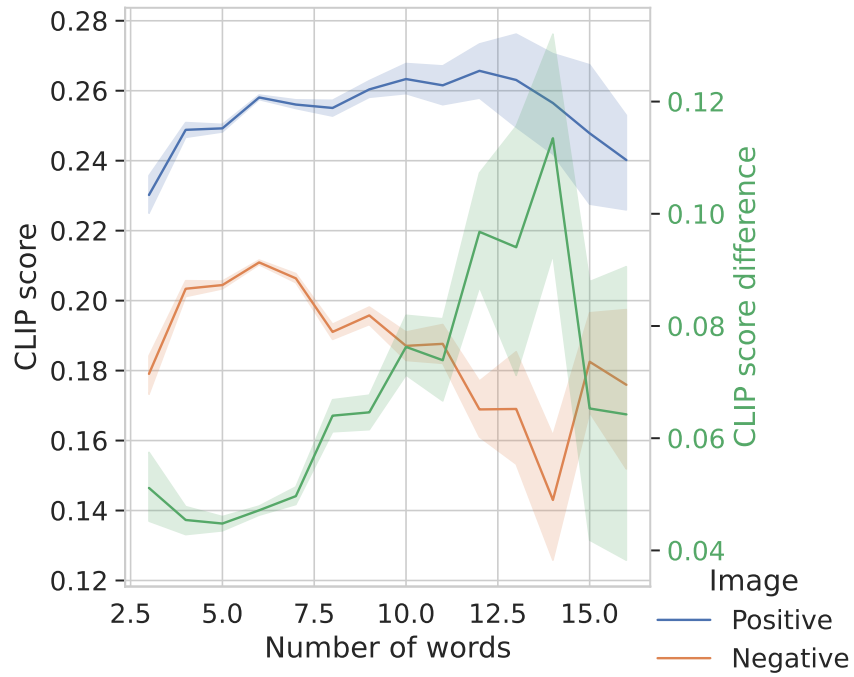


Figure 7.6: Line plot of the number of words in the caption sentence vs. the CLIP score. The shadowed areas are 95%-confidence intervals for the expected value.

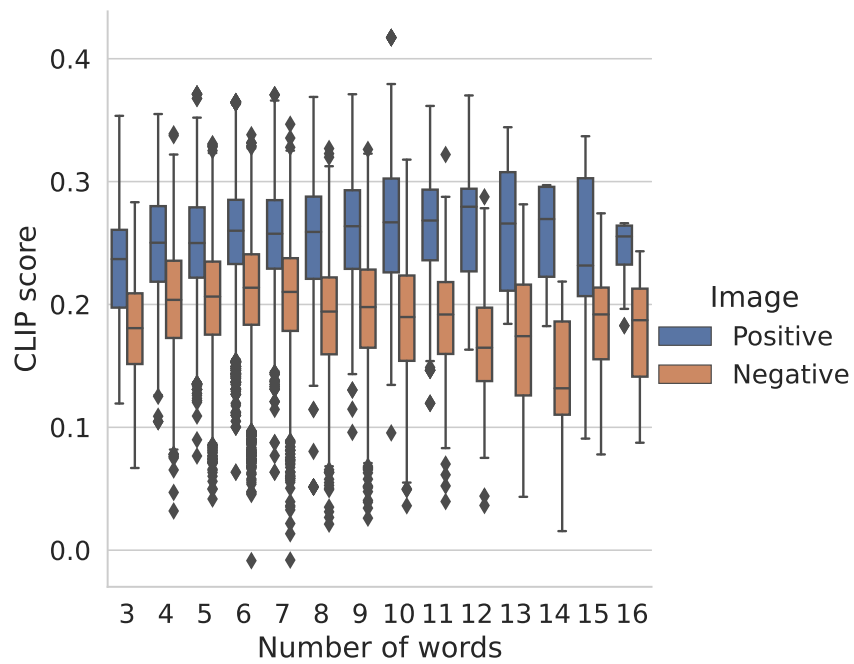


Figure 7.7: Box plot for the number of words in the caption sentence vs. the CLIP score. This plot shows the distributions, unlike Fig. 7.6 that shows the expected values.

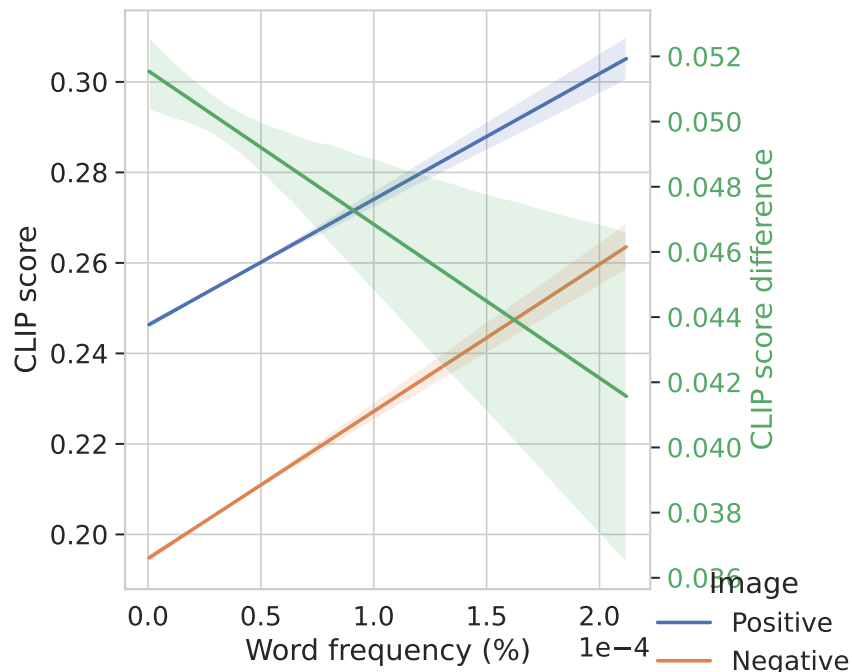


Figure 7.8: Linear regression plot of the average frequency for the words in the sentence that are common to both images vs. the CLIP score. The shadowed areas are 95%-confidence intervals for the expected value.

(7) **Similar situations confuse CLIP.** Unsurprisingly, the higher the similarity between the caption and the negative image caption, the higher the *negative* CLIP score, as depicted by Fig. 7.10.

We also studied the influence of the similarity between the *original* word (from the caption) and the *replacement* word (from the text associated with the negative image) in Fig. 7.11. The effect of the word change seems smaller than that of the whole sentence change.

(8) **CLIP performs relatively better on *nature-related* and *personal care* concepts and relatively worse on *furniture*, *transportation*, *herbivores*, *sports*, *academia*.** As mentioned in Section 7.3.1.2, score D measures the relative CLIP performance, which is more relevant for retrieval models like CLIP. Therefore, we measure the importance of each feature concerning D . Specifically, we compute the mean differences of the D scores when the binary feature is present and when it is not. We show the CLIP performance analysis on **binary** features in Table 7.1. Following the example of SEAL [186], we use ChatGPT to cluster the features under a broad topic automatically.⁴

We find that CLIP performs relatively **better** on topics related to nature: *Natural*

⁴We use the following prompt: “Name a topic for the following words: ...”

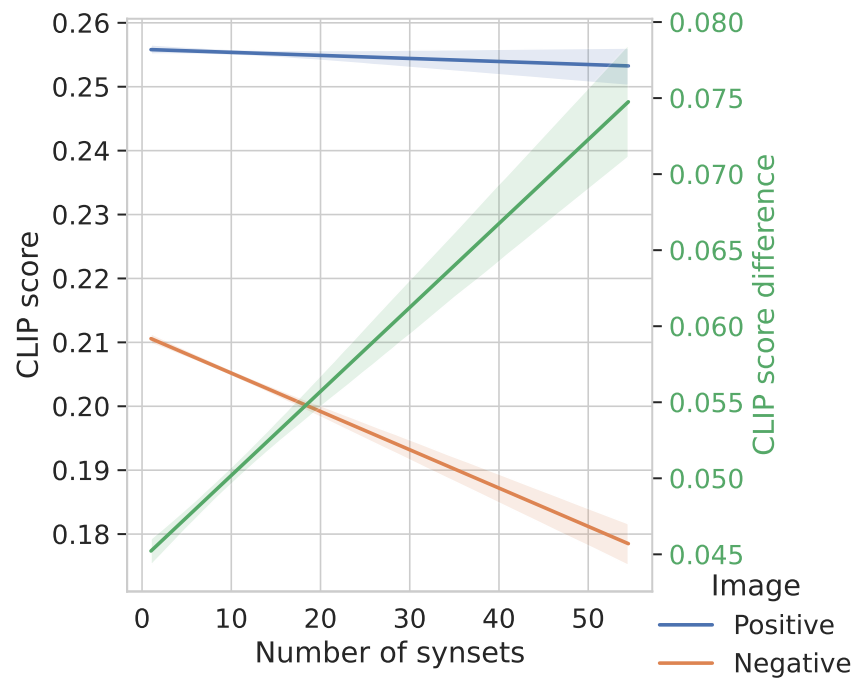


Figure 7.9: Linear regression plot of the average synset count for the words in the sentence that are common to both images vs. the CLIP score. The shadowed areas are 95%-confidence intervals for the expected value.

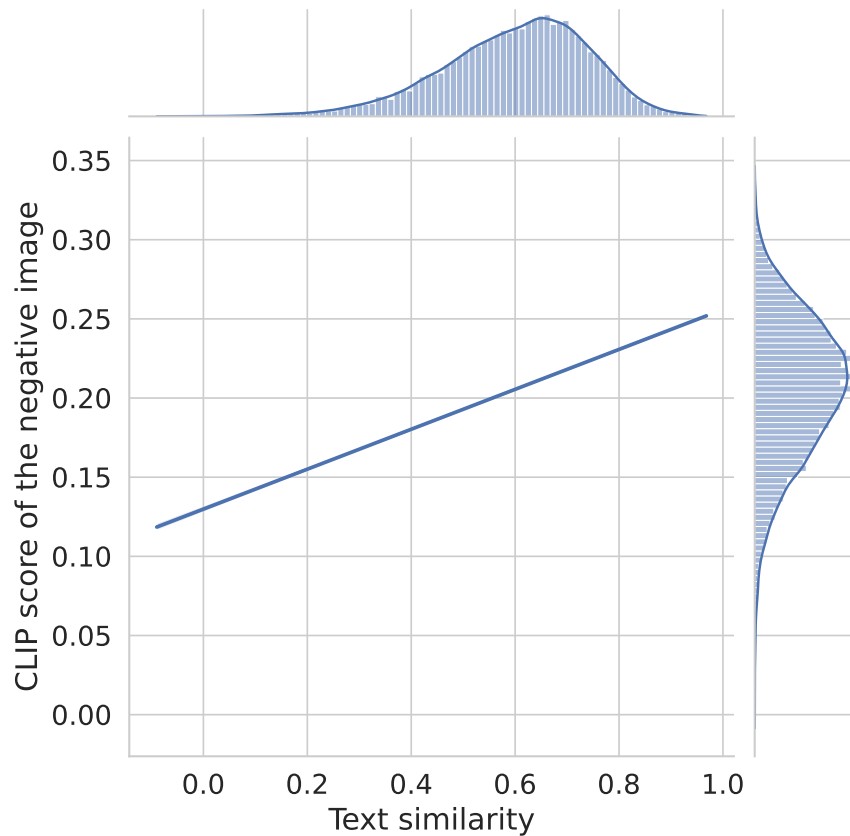


Figure 7.10: Linear regression plot of the similarity between the text caption and the negative image text caption vs. the CLIP score for the negative image. The shadowed areas are 95%-confidence intervals for the expected value. The unimodal distributions are also shown.

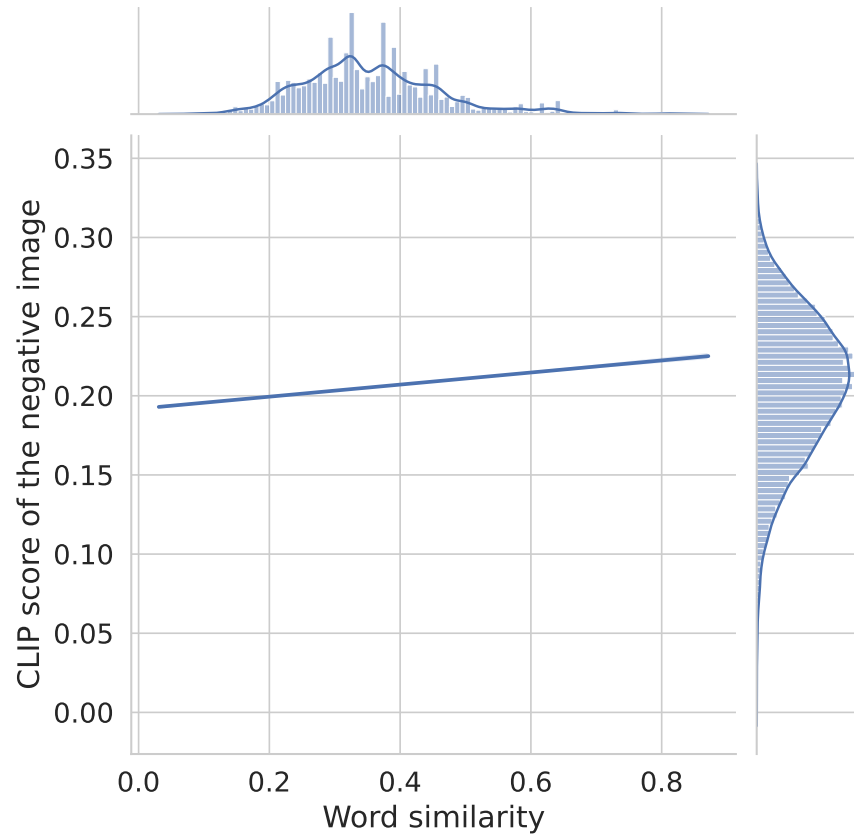


Figure 7.11: Linear regression plot of the similarity between the originally replaced word from the text caption and new word from the negative image text caption vs. the CLIP score for the negative image. The shadowed areas are 95%-confidence intervals for the expected value. The unimodal distributions are also shown.

Topic	Feature	Mean diff.	Example Words
CLIP PERFORMS BETTER ON			
Natural Phenomenon	Hypernym physical_phenomenon.n.01 (original)	0.038	snow, fog, rain, mist
	Hypernym physical_phenomenon.n.01 (replacement)	0.022	snow, rain, cloud, fog, mist
Waterfront Infrastructure	Hypernym platform.n.01 (original)	0.038	pier, deck, podium
	Hypernym horizontal_surface.n.01 (original)	0.032	pier, pavement, quay
Landscapes	Hypernym community.n.06 (original)	0.038	meadow, desert, grassland
	Hypernym natural_elevation.n.01 (original)	0.035	dune, sandbar, reef
	Hypernym geological_formation.n.01 (original)	0.027	beach, shore, cliff
	Hypernym plant.n.02 (original)	0.025	grass, tree, flower
	Hypernym natural_elevation.n.01 (replacement)	0.020	mountain, hill
Grooming	Presence of word “wash” (original)	0.035	wash
	Levin “floss verbs” (original)	0.030	wash, brush, shave
	Levin “wipe verbs”(original)	0.022	wear, sweep, trim, rub
	Levin “dress verbs” (original)	0.027	exercise, bathe, dress
Domestic Animals	Hypernym young.n.01 (original)	0.033	puppy, kitten, foal
	Hypernym domestic_animal.n.01 (original)	0.032	puppy, retriever, pug
	General Inquirer “animal” (replacement)	0.023	dog, animal, cat, goat
	Hypernym canine.n.02 (replacement)	0.021	puppy, retriever, pug
CLIP PERFORMS WORSE ON			
Furniture	Presence of word “sofa” (in common)	-0.032	sofa
	Hypernym bedroom_furniture.n.01 (in common)	-0.026	bed, sofa
	Hypernym furniture.n.01 (in common)	-0.017	couch, bed, sofa, chair, bench
	LIWC “home” (in common)	-0.015	bed, window, sofa, room
Transportation	Presence of word “ride” (original)	-0.027	ride
	Hypernym vessel.n.02 (in common)	-0.019	boat, ship, yacht
	Levin “pedal” verbs (original)	-0.018	ride, drive, fly, sail, cruise
	Hypernym craft.n.02 (in common)	-0.018	boat, balloon, ship, scooter, kayak
Herbivores	Hypernym ungulate.n.01 (in common)	-0.021	horse, cow, camel, goat, deer
	Presence of word “horse” (in common)	-0.019	horse
Sports	Hypernym happening.n.01 (in common)	-0.021	wave, win, tap, slam
	Hypernym contestant.n.01 (in common)	-0.020	footballer, golfer, goalkeeper, cricketer, tackle
	Levin “admire” verbs (original)	-0.017	stand, enjoy, admire, support
Academia	General Inquirer “academia” (in common)	-0.020	student, classroom, library, teacher, book, computer, conference
	Presence of word “student” (in common)	-0.020	student

Table 7.1: CLIP relative performance analysis on a subset of binary features: the top-5 **easier** topics are *Natural Phenomenon*, *Waterfront Infrastructure*, *Landscapes*, *Grooming* and *Domestic Animals*, while the top-5 **harder** topics are *Furniture*, *Transportation*, *Herbivores*, *Sports* and *Academia*.

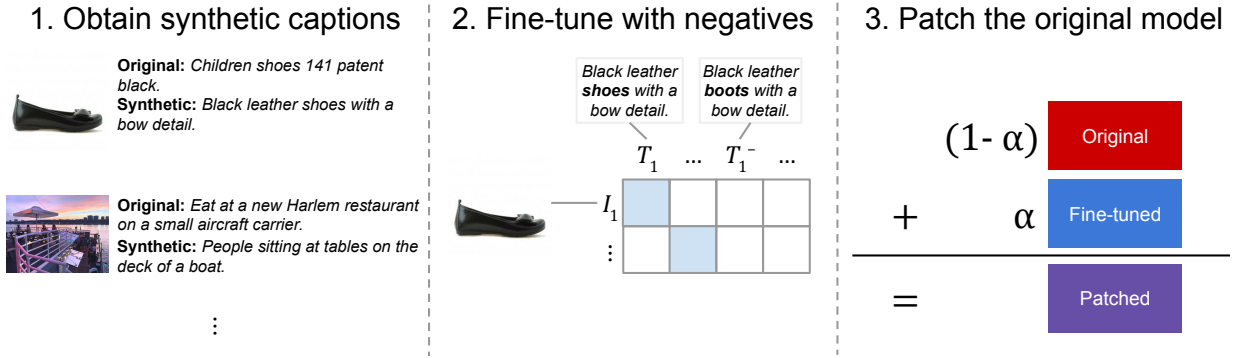


Figure 7.12: Our CLOVE framework consists of three steps. First, obtain synthetic captions for a large image dataset. Second, fine-tune a pre-trained Contrastive VLM on it along with hard negative texts. Third, patch the original model with the fine-tuned one.

Phenomenon, Waterfront Infrastructure, Landscapes, Domestic Animals, and personal care: Grooming, and worse on topics like Furniture, Transportation, Herbivores, Sports and Academia.

7.4 CLOVE: A Framework to Increase Compositionality in Contrastive VLMs

To address the compositionality limitations observed in existing models, we propose strategies for developing a contrastive VLM: data curation, contrastive learning, and model tuning. We introduce CLOVE, a framework that leverages the strengths of an existing pre-trained contrastive VLM and enhances it with language composition skills. Figure 7.12 shows an overview.

CLOVE includes the following steps, presented in more detail below:

3.1 Synthetic Captions. Synthetic data generation can be effectively used to enlarge the training data. We use a large dataset with synthetic captions.

3.2 Hard Negatives. Contrastive VLMs rely on the availability of negative training data. We add randomly generated hard text negatives to the dataset and train a fine-tuned model with increased compositionality capabilities.

3.3 Model Patching. The pre-trained model and the fine-tuned model are combined through model patching. Patching allows us to keep the compositionality obtained with the fine-tuned model while recovering the pre-trained model performance on previously supported tasks.

7.4.1 Synthetic Captions

Synthetic captions provide a great hybrid between the training dataset size and the quality of the captions. We leverage LAION-COCO [202], a 600-million dataset with images from the 2-billion-sized English subset of LAION-5B [201] that were captioned with BLIP ViT-L/14 [132], which was fine-tuned on COCO and filtered with two versions of OpenAI-pre-trained CLIP ([183]; ViT-L/14 and RN50x64). Even though the captions are limited in style (typically following the style of COCO captions), the LAION-COCO authors found that the synthetically generated captions have a similar quality to those written by humans. We believe these captions focus more on describing visual information than the captions from its original dataset (LAION), based on multiple examples from this dataset. See Section 7.5.4 for an ablation of the training dataset.

7.4.2 Hard Negatives

Text hard negatives can help the model to learn the meaning of each word better, as they need to identify whether it relates to the image depending on how it is used in a caption. [264] proposed NegCLIP, an extension of CLIP’s training procedure that generates a hard negative text for each example in the batch by rearranging the image caption words. These generated negatives are included within the negative test sets of the learning objective. [88] proposed an alternative called REPLACE and showed that the model achieves better compositionality skills if such negatives are generated from carefully selected single-word replacements. These replacements are performed on one of the entities, relations, or attributes obtained from first parsing the sentence as a scene graph, then selecting an alternative word from its antonyms or co-hyponyms by leveraging WordNet [57]⁵. These methods rely on high-quality captions. Otherwise, the generated negatives will have changes that cannot be visually appreciated or will mostly be ungrammatical or nonsensical, severely affecting the model’s downstream performance. Take the following example from LAION that accompanies an image of a cardholder: *“5x Orange Ball Wedding Party PLACE CARD HOLDER Table Name Memo Paper Note Clip.”* If we apply REPLACE, supposing we can parse the sentence correctly, the word “table” could be replaced with “bed”. However, this would not make it a negative since the table is additional contextual information the caption included that cannot be visually appreciated. Such a change will introduce more noise to the model’s training process.

For this reason, these works have employed the COCO captions [141, 32] dataset. COCO consists of images along with high-quality human-annotated captions that describe them. Nevertheless, with 600,000 image-text pairs (five captions for each of the 120,000 unique

⁵More precisely, the method proposes to look for words that share a grand-co-hypernym.

images), COCO is at least three orders of magnitude smaller than the typically used image-text training datasets. This issue limits learning and makes models overfit. Additionally, COCO presents a limited number of objects and actions. 700 out of the 1000 object classes in ImageNet-1k are not present in COCO [234]. We propose combining these hard-negative techniques with a synthetic-caption dataset, such as LAION-COCO [202] (introduced in the previous subsection).

7.4.3 Model Patching

Model patching [94] makes a fine-tuned model recover the performance on previously supported tasks while keeping the performance on the target task. NegCLIP [264] and REPLACE [88] fine-tune a model to improve language compositional skills significantly. However, in exchange, they sacrificed their performance on general object recognition, as measured by their ImageNet performance. For this reason, we propose applying one of such methods and subsequently employing model patching. This procedure consists of performing a weight-space average between the pre-trained and the fine-tuned models. Concretely, for each pre-trained model weight w_i^{PT} and fine-tuned model weight w_i^{FT} , we compute their weighted average to obtain a new model weight w_i :

$$w_i = (1 - \alpha)w_i^{PT} + \alpha w_i^{FT} \quad (7.1)$$

In Section 7.5.4, we show that this approach helps the model gain compositionality properties while maintaining its object-recognition performance.

7.5 Case Study on CLIP

To demonstrate the effectiveness of our framework, we apply it to CLIP [183], one of the most widely used contrastive VLMs. Given that previous work has highlighted the tradeoff between compositionality abilities and model performance on previous standard tasks, we evaluate challenging compositionality benchmarks and standard benchmarks for object recognition and image-to-text and text-to-image retrieval. To gain insights into the role played by the three main components of the CLOVE framework, we conduct three ablations studies to (1) determine the role of synthetic captions, (2) evaluate if employing hard negative texts during training improves the recognition performance of compositions, and (3) test the importance of patching the original model after training with hard negative texts. Unless otherwise noted, all evaluations are zero-shot, meaning we do not perform in-domain fine-tuning on benchmark-specific training splits.

	ARO				SugarCrepE			SVO-Probes			avg.
	Attr.	Rel.	C-Ord.	F-Ord.	Repl.	Swap	Add.	Subj.	Verbs	Obj.	
pre-trained	63.5	59.8	47.7	59.9	80.1	62.3	72.8	84.0	79.3	87.8	69.7
NegCLIP	<u>70.5</u>	80.1	87.0	90.1	85.1	<u>75.3</u>	85.9	90.9	84.7	<u>92.3</u>	<u>84.2</u>
REPLACE	71.2	72.9	80.1	86.7	<u>88.2</u>	74.8	<u>89.5</u>	92.0	84.6	<u>93.0</u>	83.3
CLIP+CLoVE w/o patching	69.0	77.4	91.7	93.6	88.6	76.1	90.5	88.2	83.7	91.6	85.0
CLIP+CLoVE ($\alpha = .6$)	69.7	72.7	86.6	92.1	87.0	74.6	85.8	90.5	86.4	93.3	83.9

Table 7.2: Zero-shot compositional evaluation results. The best results are in **bold**. An underline indicates results within 1% of best.

	ImageNet	Cars	CIFAR10	CIFAR100	MNIST	EuroSAT	Flowers	DTD	UCF101	HMDB51	average
	pre-trained	63.4	59.7	89.8	64.2	48.9	50.5	66.6	44.4	69.3	44.3
NegCLIP	55.8	45.6	85.9	60.9	45.3	32.9	55.9	39.0	65.6	42.7	53.0
REPLACE	52.9	42.7	84.6	60.2	36.6	34.3	51.9	34.5	62.2	40.9	50.1
CLIP+CLoVE w/o patching	53.1	48.7	88.5	62.0	40.4	46.9	43.2	36.3	62.3	41.0	52.2
CLIP+CLoVE ($\alpha = .6$)	<u>62.8</u>	56.8	91.4	68.1	<u>48.7</u>	57.4	61.1	41.2	70.4	46.0	60.4

Table 7.3: Zero-shot classification results. The best results are in **bold**. An underline indicates results within 1% of best.

7.5.1 Experimental Setup

Pre-trained Model. Rather than starting from scratch, we aim to enhance the composition capabilities of an existing contrastive VLM. This work uses CLIP (Contrastive Language-Image Pre-training; [183]), a pre-training method demonstrating impressive zero-shot performance on classification and retrieval tasks involving vision or language. It involves learning image and text representations in a joint space by leveraging large-scale weakly-supervised datasets. These datasets contain image-text pairs with varying degrees of correspondence. For each image, the model must learn the corresponding positive text from a set that includes this text and a random sample of $N - 1$ other texts (negative samples) by employing the InfoNCE objective [168]. Similarly, the model must identify which image corresponds to a given text. CLIP is trained with mini-batch gradient descent, where this objective is applied to each pair in the N -sized batch, and the negatives are typically sourced from the rest of the batch.

Implementation Details. Unless otherwise noted, the implementation details are as follows: We write our code on Python 3.10 using PyTorch [174] v2.1, starting from `open_clip`'s [95, 35] codebase. We run the experiments using the AdamW optimizer [146], with a linear learning rate warmup for 2000 steps to $1e-6$, later decayed with a cosine schedule [145]. We use a weight decay of 0.1. Our initial pre-trained model is ViT-B-32 from

OpenAI [183]. We train the models through one billion examples by randomly sampling with replacement from shards of up to 10 000 samples, where the final size of each depends on the image availability at download time. We successfully downloaded about 80% of LAION-400M [203], 80% of LAION-COCO [202], and 60% of COYO-700M [21] images. The text captions are in English. We employ one node with 8x A100 Nvidia GPUs and 96 CPU cores (p4d.24xlarge from AWS) for four days and a half. ⁶ The batch size is 256 per GPU.

The choice of learning rate was based on multiple preliminary experiments to make sure it was not learning too slowly or that it was making the training loss go up. The training steps and samples were selected to ensure we gave enough time for the method to learn and converge. The choice of total batch size and compute budget was determined based on our availability compute and considering that CLIP-like methods need a large batch size. All reported experiments are based on a single run since they are computationally expensive.

We re-implemented REPLACE [88] with the following changes and decisions, primarily because the code for this part is unavailable. We skip employing BERT [47] to filter the generated negatives. Instead, they proceeded to replace words based on the frequency of the new words, which is a first-order approximation of computing probabilities with a contextualized model. For the replacements, given that the authors do not mention prepositions but we find them replaced in the provided data, we proceeded to replace prepositions. For the replacement words, we try to respect the rest of the sentence by conjugating them (e.g., the person for the verbs, and the number for the nouns) and using a similar casing to the replaced word. We used spaCy [87] v3.7.2 (the model `en_core_web_sm`) and `pyinflect` v0.5.1. We employed a different Scene Graph Parsing implementation, `SceneGraphParser` v0.1.0. We avoid replacing a word with a potential synonym by looking at the synsets in common of their lemmas from WordNet [57], leveraging NLTK [16] v3.8.1. We managed to reproduce the same numbers the original authors reported. We will make our code publicly available to make it easy for anybody to reproduce and build on top of our results.

We set $\alpha = 0.6$ for the model patching based on the ablation from Section 7.5.4.

7.5.2 Using CLOVE to Bring Compositionality into CLIP

We compare the CLIP model enhanced with our CLOVE framework against several baselines, as shown in Fig. 7.2: CLIP+CLOVE leads to an average 10% absolute improvement on the challenging compositionality benchmark SugarCreme [88] when compared to a pre-trained CLIP model, all while maintaining its ImageNet performance within 1%. Additionally, we

⁶Our main results can be achieved with about 10% of the training time. We train longer to let some ablations converge and thus establish fair comparisons.

	Text-to-Image/Video				Image/Video-to-Text				avg.
	CC3M	DiDeMo	MSR-VTT	YouCook2	CC3M	DiDeMo	MSR-VTT	YouCook2	
pre-trained	52.3	48.4	54.9	13.8	51.0	40.7	50.8	11.3	40.4
NegCLIP	50.3	48.8	56.9	13.9	47.9	41.9	48.2	09.8	39.7
REPLACE	49.6	50.2	56.2	13.6	44.8	40.8	47.9	09.7	39.1
CLIP+CLoVE w/o patching	47.3	35.0	53.1	11.4	43.4	37.8	42.7	08.0	34.8
CLIP+CLoVE ($\alpha = .6$)	58.7	<u>49.9</u>	60.5	15.7	57.5	47.5	54.5	12.4	44.6

Table 7.4: Recall@5 for the zero-shot retrieval results. The best results are in **bold**. An underline indicates results within 1% of best.

show that our model performs better than others in compositionality when we do not apply the model patching step.

In Table 7.2, we show a comparison of our enhanced CLIP+CLoVE model on others in three compositionality benchmarks: ARO [264], SugarCrepe [88] (over its three coarse-grained tasks), and SVO-Probes [82]. Note that, for SugarCrepe, we employ the macro-average to compute the coarse-grained task results like in [231] and unlike the original paper, since we are interested in measuring the global phenomena instead of giving importance to the task sample sizes. See Appendix D.1 for the performance on SugarCrepe for each fine-grained task.

Since a primary concern in previous work when devising methods that increase model compositionality was the loss in performance on other tasks, we evaluate the CLIP+CLoVE model performance on object recognition and image-to-text and text-to-image retrieval tasks.

In Table 7.3, we compare use the following object recognition benchmarks: ImageNet [44], Stanford Cars [116], CIFAR10 [119], CIFAR100 [119], MNIST [123], EuroSAT [81, 80], Oxford Flowers 102 [167], Describable Textures (DTD) [37], UCF101 [211], and HMDB51 [121]. Following [183], we employ the top-1 accuracy metric, except for Oxford Flowers 102, where we use the mean per class.

In Table 7.4, we present results on zero-shot text-to-image and image-to-text retrieval tasks. The datasets used are: Conceptual Captions [206] (CC3M), Distinct Describable Moments [7] (DiDeMo), MSR-VTT [251], and YouCook2 [280] (YC2). We present the results employing Recall@5 – the same metric used by [183]. Unlike in classification, our approach improves over the rest on average by at least 4% (absolute). We speculate this improvement comes from retrieval captions being longer and more complex than class label templates, which allows us

	mWAP	mSAP
pre-trained	24.8	28.6
NegCLIP	23.3	27.2
REPLACE	24.0	27.7
CLIP+CLoVE w/o patching	23.3	27.4
CLIP+CLoVE ($\alpha = .6$)	27.9	32.4

Table 7.5: Results on RareAct. We employ the same mean Average Precision (mAP) metrics defined by the original benchmark [156].

to appreciate our model’s rich text representations. We also believe using multiple prompts per class in classification tasks averages out the text representation noise from other models (see Appendix D.2 for an analysis of this). By using our CLoVE framework on CLIP, we obtain better performance across all tasks and metrics, except for DiDeMo in text-to-image, whose performance is on par with REPLACE.

7.5.3 Generalization to Unseen Verb-Object Compositions

We believe that CLoVE allows CLIP to recognize compositions better, even when unseen or unusual, and the parts that form them are widespread. For this, we rely on evaluating our method and the baseline on RareAct [156], a dataset with more than a hundred manually annotated actions for 7607 ten-second clips from 905 YouTube videos. The authors obtained these actions by combining verbs and nouns that rarely co-occur. We convert these actions into texts by conjugating the verbs into the gerund form, using an indefinite article, and employing the noun in its singular form. We use the same templates as in UCF-101 (from the original CLIP’s paper [183]). An example text is: “A video of a person cutting a towel.”

We present the results in Table 7.5. Our method surpasses CLIP by 3 points and outperforms existing models as well. Surprisingly, our method without the patching step performs worse than the baseline. We hypothesize that hard negative training introduces a forgetting behavior necessary to succeed in this task, unlike most other compositionality benchmarks we consider in this chapter. Under this scenario, the patching would recover performance on both standard and compositional tasks, being better than both models even when they perform similarly on average since they seem to commit different types of errors. In addition, to close the gap with human performance, besides compositionality, we believe some actions require understanding motion to close the gap with human performance besides compositionality.

Fine-tuning dataset	Attr.	Rel.	C-Ord.	F-Ord.
pre-trained	63.5	59.8	47.7	59.9
<i>Without hard negative texts</i>				
COYO	63.6	55.4	34.8	43.4
LAION (L)	<u>64.9</u>	64.0	40.2	47.0
COCO (C)	62.5	61.6	73.8	39.8
concat. L & C	65.9	59.0	43.7	50.3
sample unif. L & C	64.6	55.7	59.8	29.7
LAION-COCO	<u>65.4</u>	66.0	70.5	76.9
<i>With hard negative texts</i>				
COYO	<u>69.5</u>	75.6	71.7	79.7
LAION (L)	67.9	72.6	78.3	85.4
COCO (C)	70.2	67.6	<u>90.9</u>	74.5
concat. L & C	<u>70.1</u>	76.2	83.4	88.6
sample unif. L & C	<u>69.9</u>	71.6	82.7	60.8
LAION-COCO	69.0	77.4	91.7	93.6

Table 7.6: The zero-shot performance of fine-tuning CLIP with different datasets, with and without hard negative texts. The best results are in **bold**. An underline indicates results within 1% of best.

We note that even when we did not guarantee that our model has not seen such actions ever before (from LAION-COCO or indirectly from COCO’s training set via LAION-COCO’s creation method), most of RareAct’s actions are particularly unusual (e.g., “blending a phone”, “measuring an egg”, and “drilling a laptop”). If they had appeared in any of the directly or indirectly used training sets, we believe it would be reasonable to expect their occurrences to have been minuscule. Still, I intended to check if they were present in LAION-COCO. However, I have lost access to it since I finished my internship at Netflix, and it has not been available to download for a long time.

7.5.4 Ablation Studies

The Importance of Synthetic Captions. We hypothesize that training dataset quality is essential to model compositionality performance. For example, in LAION [203], a dataset commonly used to train Contrastive VLMs, you can find examples that present excessive information that cannot be easily mapped to visual concepts depicted in any image, such

	Attr.	Rel.	C-Ord.	F-Ord.
pre-trained	63.5	59.8	47.7	59.9
fine-tuned	65.4	66.0	70.5	76.9
+ negatives	<u>69.0</u>	77.4	91.7	93.6
+ negatives*	69.4	75.4	77.5	86.1

Table 7.7: The importance of employing negatives to improve the zero-shot performance on recognizing compositions. The best results are in **bold**. An underline indicates results within 1% of best. *The last row shows the results of using half the batch size – there are gains even when the total device memory is the same, given that employing negatives effectively doubles the batch size.

as: *“Platinum Dance Academy T-shirt. Orders must be placed by Friday, September 26th. Delivery approximately 2 weeks or less.”*

We could employ datasets with high-quality annotations such as COCO [141, 32], but such datasets are typically small (less than a million samples). A hybrid approach, with high-quality data and a large dataset, can be obtained using synthetic captions, as described in Section 7.4.1. We are interested in comparing this dataset with LAION-400M or COCO directly, as well as two ways to combine the datasets: a) concatenation and b) sampling with equal probability.⁷ Note that these strategies of combining LAION and COCO are completely different from the LAION-COCO dataset. In addition, we consider COYO-700M [21], a large-scale dataset constructed similarly to LAION-400M.

Table 7.6 compares the performance of fine-tuning a pre-trained CLIP model on different datasets without employing negatives. In this table and subsequent ones, the best results are in **bold**, and an underline indicates results within 1% of best. LAION-COCO [202] presents the best results overall, with a large margin on ARO. For this benchmark, it is the only presented dataset that significantly outperforms the pre-trained model. In the case of the SugarCrepe benchmark, we observe that all datasets provide improvements over the pre-trained model. Interestingly, [15] also found synthetic captions helpful for text-to-image generation models. They show synthetic captions help such models generate images that align better with the input text.

The Importance of Hard Negatives. [264, 88] showed that employing randomly generated text negatives as part of the training process can significantly improve the language compositionality skills of pre-trained models. We apply REPLACE [88] to obtain randomly

⁷Note LAION-400M is about 700 times larger than COCO.

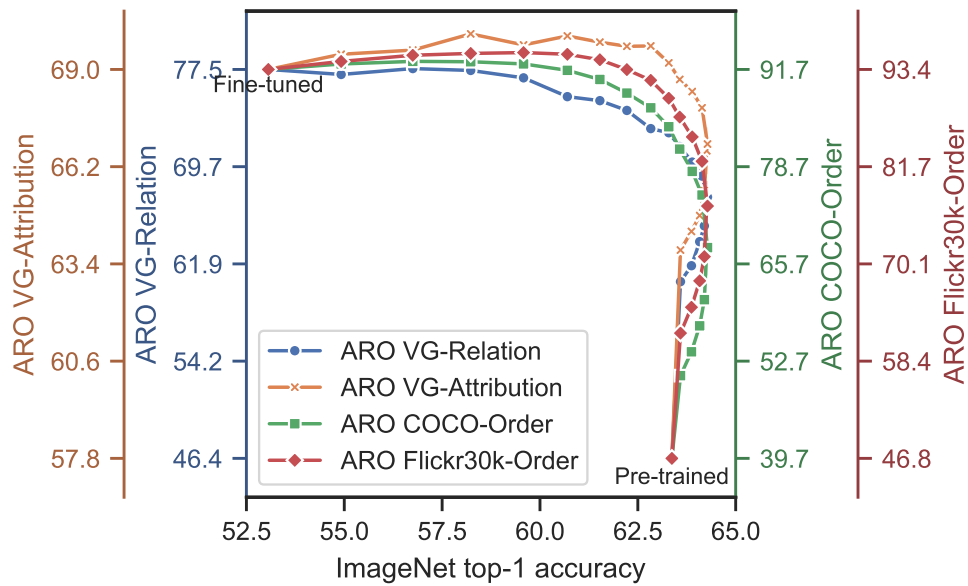


Figure 7.13: The effect of applying model patching to both an object-centric benchmark (ImageNet, [44]; x-axis) and a compositionality benchmark (ARO, [264]; the four y-axes represent its four tasks), when varying the value of the weight in the average, α . The value of α varies from 0 (the pre-trained model) to 1 (the fine-tuned model) in 0.05 increments, and the lines connect such points. We can obtain models with good zero-shot performance in ImageNet and compositionality when α is around 0.4–0.7. Note the four y-axes were adjusted to make the pre-trained and fine-tuned model points match to focus on how the lines vary between them.

generated hard negative text along with the LAION-COCO dataset [202] and compare it to fine-tuning without negatives. We present the results in Table 7.7. In this setting, we can observe that employing negatives improves performance over not using them, as measured by the ARO benchmark [264] (its tasks are, in the order that we show them: VG-Attribution, VG-Relation, COCO-Order, and Flickr30k-Order).

The Importance of Model Patching. Existing methods to improve CLIP’s compositionality by employing negatives used by [264, 88] do so by considerably hurting the model’s performance on more standard object-centric benchmarks such as ImageNet [44].

Figure 7.13 presents the effect of varying this value for both a compositionality benchmark and an object-centric one. The model performs well on both when α is around 0.4–0.7.

7.6 Conclusions

In this chapter, we proposed a simple and effective method to probe vision-language models. Our method is scalable, as it does not require data annotation and uses existing datasets. We analyzed the performance of CLIP, a popular state-of-the-art multi-modal model, on the SVO-Probes benchmark with our method. We confirmed the recent findings of [228] of CLIP behaving like a bag of words model and that of [171] of CLIP performing better with nouns and verbs. We also uncovered novel findings, such as that CLIP gets confused by concrete words but surprisingly improves performance for more ambiguous terms or that the frequency of words does not significantly change CLIP’s behavior. We hope our work contributes to ongoing efforts to discover the limitations of multi-modal models and help build more robust and reliable systems. Our framework can be easily used to analyze other benchmarks, features, and multi-modal models, and it is publicly available at github.com/MichiganNLP/Scalable-VLM-Probing.

We then introduced CLOVE – a framework to considerably improve the compositionality of pre-trained Contrastive VLMs while preserving their performance on other tasks, unlike existing methods. Our approach combines fine-tuning contrastive VLMs with hard negative texts by leveraging synthetically captioned images, as they can provide an excellent tradeoff between quality and quantity. Subsequently, it patches the original model with the fine-tuned one to convey the best of two worlds: compositional skills while maintaining performance on other tasks.

We showed experimentally that CLOVE improves the performance of CLIP-like models on multiple benchmarks, both compositionality-related and non-compositionality-related. We presented improvements over the baseline on RareAct [156], a dataset of actions formed

by infrequent verb-noun pairs. We ablated the different components of our framework and showed their importance: data quality, the use of hard negatives in training, and model patching.

Our code and pre-trained models are publicly available at github.com/Netflix/clove. Our code allows for effortlessly replacing CLIP-like weights with the ones we provide, considerably boosting language composition performance.

CHAPTER 8

Conclusions

In this dissertation, I explored different aspects to make a way through more realistic video understanding. Throughout it, I introduced novel approaches to bring data, evaluation frameworks, and methods that help with this goal.

8.1 Research Questions Revisited

The findings for the originally formulated research questions are the following:

1. Can we build a language-based video understanding benchmark for overlooked real-life domains, such as daily situations and in-the-wild scenarios?

I presented two new benchmarks considering real-life domains in Chapters 2 and 3. The first benchmark, called LifeQA, contemplates life situations within long videos, including scenes at home, at school, and on the streets. To perform well on it (i.e., to answer correctly the questions related to the displayed situations), methods should understand each real-life story and its context. I demonstrated the task’s difficulty through multiple analyses and experimental evaluations. Existing models present a significant performance gap compared to humans, indicating that further work is necessary to combine multiple modalities successfully in this task.

The second presented benchmark, WILDQA, evaluates models on five domains that portray different aspects of nature. Like LifeQA, this dataset considers minute-long videos, and its methods present a gap concerning how a typical person would perform on them. Unlike in LifeQA, I proposed an evaluation format based on generating open-ended answers instead of choosing the correct one and avoiding the distractors. Open-ended answer generation can help craft systems that meet users’ needs more intrinsically. For this benchmark, not only do methods need to be employed to generate answers, but they also need to show video evidence of where to find them. Overall, I

consider this evaluation format a significant step towards a more realistic assessment of video understanding.

Besides the evaluation, I experimentally showed that multi-task training can improve methods' performance. Video evidence selection plays another vital role, too; choosing moments when answering a question brings insight into what the model attends to. This is a crucial step for explainability and interoperability in multimodal systems.

The data for both benchmarks is publicly available for anybody to use, and I also released the code for practitioners and fellow researchers to reproduce our results.

2. Does combining vision and language help better recognize naturally occurring human behaviors in videos?

Chapter 4 showed evidence that combining visual, speech, and language features helps recognize sarcasm better in videos when compared to using a single modality (only vision, only text, or only audio). I introduced a multi-source dataset and benchmark, MUsTARD, to evaluate the extent to which methods can recognize sarcasm in short videos.

After exploring multiple approaches, I tackled the issue of collecting sarcasm in videos. I finally gathered sarcastic videos using two approaches: annotating many samples from one source and explicitly searching for sarcasm from other sources on the web. Two raters annotated the videos through multiple rounds with a binary value indicating if they were sarcastic. In cases of disagreements, we employed a third annotator to break ties. I accomplished a suitable inter-annotator agreement, with a Kappa of 0.2326 for the first approach and 0.5877 for the second. For each sample, MUsTARD includes a video of the utterance with its binary sarcastic value and a second video that consists of the dialog up to that point. It also includes the transcripts for the videos.

I crafted text, audio, and visual features to conduct several experiments. Apart from showing the importance of multiple modalities, I revealed that audio plays a more critical role in generalizing across speakers and sources. I also experimentally demonstrated that knowing information about the speaker and modeling the context improves results. The dataset and code are published online for others to use and replicate.

3. Can language be leveraged to build an automatic evaluation framework for video understanding that better reflects real-life situations?

In Chapter 5, I introduced a format of evaluating the extent to which methods understand short videos by filling in a sentence with a blank. These methods must fill these

blanks with noun phrases representing entities from the videos. After looking at various ways we could employ language to evaluate models, I decided to use noun phrases. This dataset and evaluation framework, FIBER, proves to be challenging for models while preserving a higher human agreement than video captioning evaluation metrics (i.e., it is more precise, as it better accounts for diversity). Even though multiple-choice evaluations can achieve a high human agreement, the presented evaluation method is more challenging and suited for real-life situations since it forces methods to generate arbitrary-length noun phrases.

FIBER is based on VaTeX [241], containing 28,000 ten-second videos and manual annotations we collected for the blanked captions to consider more correct reference answers and thus reduce the false negative rate during evaluation. I conducted multiple analyses to evaluate the quality of the data and the annotations. Naturally, we find that the more annotators per blanked caption, the better to consider diversity when filling in a blank. I presented the data diversity and complexity of the task through multiple analyses (including in Appendix B), including the long-tail distribution of answers, many examples of them, and their par-of-speech distribution.

I proposed novel multimodal methods to tackle this task. Such methods leverage T5 [184] in a principled manner, given that it was pre-trained to fill in text blanks. I showed that these methods perform well, but there is still a gap regarding human performance.

4. Can large pre-trained image-text alignment models be used for robust zero-shot video understanding?

I revealed in Chapter 6 that a minor refinement can be applied over a large pre-trained image-text model such as CLIP [183] to boost the performance on unseen video tasks and domains noticeably. This method is state-of-the-art in four out of five zero-shot benchmarks evaluated. It allows practitioners to use a single model for multiple use cases, making deployment scalable to new tasks and domains. This presented method could be replicated for future image-text models, making it easy to leverage them for video tasks, unlike methods that rely on extensive and expensive training procedures. Practitioners can simply swap CLIP’s weights with FitCLIP’s and receive a free boost in performance in zero-shot video tasks with no extra inference cost.

I showed the importance of the model patching step in recovering the performance of the original model. Without it, the method underperforms the baseline. I ablated the choice of the model patching alpha value and the ratio of pseudo-labels to labels in Appendix B. I also compared the text-to-video retrieval ranking distribution of

MSR-VTT between FitCLIP and CLIP, given that some methods are better in earlier rankings while sacrificing the long tail. Still, I demonstrated that FitCLIP is better or equal to the baseline at virtually all points. I compared the best and worst-performing Moments-in-Time classes of FitCLIP vs. CLIP and found that the model improves on more abstract actions while presenting a slight deterioration in low-level actions such as “slicing.”

The code is available for anybody to reproduce.

5. Can we align vision and language models so that they better generalize to unseen verb-object compositions?

In Chapter 7, I presented a method that improves the compositionality of CLIP [183] while keeping or improving the performance in standard benchmarks. In particular, I introduced a noteworthy improvement in unseen verb-object compositions.

I first proposed a method to probe CLIP. This framework, which anyone can replicate in a different setting, consists of selecting features from images and captions (e.g., how many words the caption has) and checking if there is a significant correlation with CLIP’s performance. I leveraged SVO-Probes [82], a dataset of pairs of images that differ in either their subject, verb, or object, along with their associated captions. This dataset allowed me to consider only the difference between the images while controlling for the other factors. By taking advantage of this framework, I arrived at several findings.

Many factors affect CLIP’s performance, even when orthogonal to the alignment between the visual world and language. I showed that CLIP’s performance worsens when comparing captions containing concrete words (e.g., “dog” and “apple”, as opposed to “mathematics” and “acceleration”). CLIP’s performance is affected by the number of words in the caption, preferring average-length sentences. CLIP scores are higher when the image contains some elements described by the caption, even when the caption does not align (e.g., referring to an apple but in a completely different context).

The method I proposed to fix compositionality issues, CLOVE, succeeds at significantly improving the baseline performance on challenging captions that test for compositional behavior (e.g., two captions with the same words but in a different order). Most importantly, CLOVE improves the compositionality skills while keeping the performance on standard classification. Additionally, my method improves retrieval, which requires understanding more complex language than class names. In Appendix D, I showed that our method is more robust to not using prompts than the baseline.

I ablated the different decisions and components of CLOVE. I revealed that the choice of the dataset is essential and found that synthetic captions can be extensive and high-quality compared to the typical training set. I experimentally proved that negative-text training is crucial to bringing compositionality, even when keeping the total batch size the same. I exhibited the importance of model patching and experimented with several values for α .

My method’s code and pre-trained models are available on GitHub for other researchers and practitioners.

8.2 Future Directions

I have carried out steps toward achieving a realistic understanding of videos in this work. However, some critical questions remain in this domain, as well as questions I have identified that derive from my work.

8.2.1 Other Common Realistic Human Behavior

Sarcasm is a typically manifested human behavior that I considered in this work. Still, other forms of expression have been understudied in the video understanding area. These forms include humor (e.g., jokes and other non-serious ways of communication), deception (e.g., lies and misdirection), and non-verbal communication (e.g., facial expressions and gestures such as pointing and using mime). How can we build methods that consider all these forms of communication to understand realistic videos better? How can we holistically represent how people express and act differently?

8.2.2 Understanding Novel Actions Compositions Through Motion

In this work, I explored methods for understanding short out-of-domain videos through zero-shot learning, which involves representing a video as an unordered set of frames. This approach to treating the temporal dimension often fails, particularly when considering long videos or multiple actions. A natural extension of my work considers video-text alignments such that the temporal dimension is represented accordingly. Given this, some interesting questions arise. How can we build such a video-text alignment method? Can image datasets or pre-trained models be leveraged to avoid gathering large video datasets and a resource-intensive training procedure? Can the time dimension be represented so that the actions are understood for the right reasons instead of relying only on accidental features?

8.2.3 Extending the Fill-in-the-Blank Framework to Other Tasks

This dissertation proposes a challenging way to evaluate methods for video understanding that provide high human agreement. However, other tasks also suffer from noisy automatic evaluations due to human diversity. Such tasks include Text Summarization, Story Generation, and Image Captioning. How can we apply the fill-in-the-blank evaluation to other tasks?

8.2.4 Fixing Compositionality Generalization Root Cause with Inductive Biases

In this work, I presented a method to patch existing vision-language models to fix their compositionality generalization skills. However, the performance is still far from a human’s, and we did not provide guarantees with other types of compositions (e.g., multiple sentences). To avoid necessitating counter-examples for every phenomenon, I speculate that part-whole inductive biases [85] could force the model only to use the parts correctly in a composition. For example, “wooden” should not influence “cat” in “The cat is on top of the wooden table,” even when a model has never seen this phrase before. I believe that interventions from at least the language encoder side are required, as reported by [105]. As evidence, [189] shows that text-to-image generation can improve attribute binding (an adjective correctly modifying a noun for a generated image) when restricting the attention maps to follow the sentence structure.

APPENDIX A

Video Understanding in In-The-Wild Scenarios: Supplementary Material

A.1 Annotation Details

A.1.1 Video Selection and Processing

Video Selection. For the video selection part, as mentioned in Section 3.3, first, we identify five domains, *Agriculture*, *Geography*, *Human Survival*, *Natural Disasters*, and *Military*, to collect videos recorded in the outside world. We then identify eight (8) YouTube channels and crawl videos from those channels. During crawling, we manually substitute irrelevant videos, such as advertisements, with videos that contain scenes primarily recorded in the outside world from the same channel.

Video Processing. As mentioned in Section 3.3, we clip the raw videos into short clips by PySceneDetect because the raw videos can be as long as an hour. We then concatenate these short clips so that the output video will be around 1 minute. **The output videos are used for the following annotation process.** We want to include longer videos because the videos recorded in the outside world usually contain less information than videos about human interactions. Besides, if the concatenated video is at the end of the original video, it is allowed to be shorter than 1 minute. We select the concatenated videos that only contain scenes recorded in the outside world. If none of the concatenated videos satisfies, we manually clip the original videos to get an output video.

A.1.2 Annotation Instructions

As mentioned in Section 3.3, we have 2 phases in our annotation process as shown in Fig. 3.3. In Phase 1, annotators develop a hypothetical motivation, ask questions, and provide the

corresponding answers with relevant parts of the video as evidence. Phase 2 is to collect answers and evidence for questions we collect in Phase 1. The following are the instructions for these two phases.

Instructions for Phase 1

We need help with this Video QA task based on video content (including the audio).

In this task, we suppose you can hypothetically send a robot to where you want to collect necessary information for many hours. In this hypothetical scenario, you have an objective that you want the robot to learn about. This robot can chart territory and can answer questions based on recorded videos. Therefore, after it comes back, you can ask questions to help you satisfy your objective; this robot will provide you with answers and video evidence clips to support the answers.

In this task, to simplify, the provided videos represent places where you could potentially have sent the robot and are much shorter (a few minutes). Given a recorded video, please help us provide one hypothetical objective that makes sense, along with questions, answers, and evidence. Specifically, you should pretend to be both the information-seeker and the robot, which means that as the robot, you could watch the recorded video, and you should provide answers and video evidence clips; as the information-seeker, you have an objective, **not** watch the whole video (because of practical reasons), and you can only ask questions and receive answers and video evidence clips as feedback.

1. Basic Instructions

- You will need to propose a hypothetical objective (or topic, intention, motivation) to motivate the questions, such that it makes sense for the given video.
- You will need to provide as many questions as you need (to satisfy your objective) about the content in the videos and that you seek to understand more about the proposed objective.
- You will first watch the video, but when you are providing the objective and questions, please pretend you **haven't** seen it before.
- You must provide at least one question for each video. **The more the better.**
- You will need to identify the source of your question (whether it is based on the visual scene or the audio) and classify your question accordingly.
- You will need to provide the correct answer to your question, as supported by the content in the video.

- You must provide video evidence (video clip) to support your question and answer.
- If one video doesn't make sense at all, or there's no possible objective for this video that makes sense, please comment at the bottom of this annotation page (and fill in the mandatory fields for the corresponding video with placeholder values).

2. How To Propose Hypothetical Objective

- For each video, you need to develop a hypothetical objective (or intention, motivation, topic) that makes sense for this video and briefly explain it.
- Your questions should all relate to this objective.
- Example 1:
 - Objective: I want to learn about the water in the territory.
 - Question 1: How big is the lake?
 - Question 2: Are there any boats in the lake?
 - Question 3: Where is the river?
 - ...
- Example 2:
 - Objective: people/life movement
 - Question 1: Is there any sign that wildlife has passed this area?
 - Question 2: How much traffic is there on the road?
 - ...

3. How To Ask Your Question

- Your question should relate to your proposed objective.
- For each video, after you finish one question, you could click the Add one more question for this video button to continue to provide another question for this video. On the contrary, if you want to delete one question, you can click the Delete this Question button.
- Ask one question at a time.
 - E.g., “Are there any people? What are they doing?” is not appropriate.
- When you provide multiple questions for the same video, make sure these questions are **independently** asked.
 - E.g., “What is growing on pine trees?” and “What is their color?” are not independent.

- The answer should be derived from the video (visual or audio).
 - E.g., “Why do they run every morning?” is not a good question.
- Ask from the 3rd person point of view.
 - E.g., “What do *we* have on this farm?” -> “What do *They* have on this farm?”
- Try to balance the questions such that the answers are not too repetitive (E.g., too many ‘yes’ answers).
- Ask questions matter-of-factly (as **objectively** as possible). Stick to what you can see or hear from the video.
 - E.g., “Does it make people feel good here?” is somehow subjective.
- Don’t ask questions about how’s the video being recorded, the camera person, or the camera itself. Ask about the content itself. Ignore what the camera person is doing.
 - E.g., “What’s the *cameraman* doing?” / “How fast is the *camera* moving?” are not good questions.

4. How to identify the Question Category

We have some basic categories: **Motion, Spatial Relationship, Temporal Relationship, Reasoning, Number, Entity, Existence, Time, Location, Other.**

If your questions fall into **multiple categories**, please check all categories that apply.

Here are some example questions under each category:

- **Motion:** What is the group of soldiers doing?
- **Spatial Relationship:** What is driving beside the motorcycle?
- **Temporal Relationship:** What happens before the black smoke rises?
- **Reasoning:** What makes changing between targets possible for the missile?
- **Number:** How many fighters are flying?
- **Entity:** What is the bullet’s target?
- **Existence:** Is there a lake by the mountain?
- **Time:** How long can the missile fly?
- **Location:** Where is the tank?
- **Others**

5. How To Provide Correct Answer

- Your answer should be written as **full sentences** (at least one).
 - E.g., “Left” → “The landspout bends toward the left.”
- The answer should be derived from the video (visual or audio).
 - E.g., “These plants are green because they contain chlorophyll.” is not a good answer.
- Provide answers matter-of-factly (as objectively as possible). Stick to what you can see or hear from the video.
 - E.g., “beautiful” is likely not a good word to use within an answer.
 - E.g., “This takes some bravery to do.” is somehow subjective.
- Don’t answer about how’s the video being recorded, the camera person, or the camera itself. Answer about the content itself. Ignore what the camera person is doing.
 - E.g., “There are two people, i.e. a running child, and the *cameraman*.” is not a good answer.
- When you enter numbers, please enter digits instead of text.
 - “Seventeen” → “17”

6. How to provide video evidence

- The video evidence consists of **all** the parts of the video that support the answer to your given question.
- You need to provide at least one video evidence clip (intervals within the video) for each question.
- You need to provide both the **start point and end point** for all the video evidence you identify in the video;
- You could use your mouse or ←/→ key to **click or drag the process bars** of the start and end points. When you click or drag the bar, the above video will change accordingly so that you can locate the points according to the video screen.
- For each video evidence clip, the end point should be **greater than zero**, and the end point should be greater or equal to the start point.
- The video evidence clips (the time gap between the start and end points) should be as short as possible.

Instructions for Phase 2

We need help with this Video Question Answering task based on video content (including the audio).

1. Basic Instructions

- You will first watch the video, then answer each question in turn.
- You must provide at least one answer for each question (ignoring differences such as upper/lower case or the article). **The more answers, the better**, but every answer should be correct.
- You will need to identify the source of your answer (whether it is based on the visual scene or the audio).
- For each answer, you must provide video evidence (video clip) to support your answers. See below for additional information.
- If one video or question is unavailable, please comment at the bottom of this annotation page (and fill in the mandatory fields for this video/question with placeholder values).
- There are five questions; you need to finish all five questions according to the content in the video (including audio).

2. How To Answer

- Provide one or more answers for each question.
- Each answer should be written as full sentences (at least one).
 - E.g., “Left” → “The landspout bends toward the left.”
- The answer should be derived from the video (visual or audio).
 - E.g., “These plants are green because they contain chlorophyll.” is not a good answer.
- Respond matter-of-factly (as objectively as possible). Stick to what you can see or hear from the video.
 - E.g., “beautiful” is likely not a good word to use within an answer.
 - E.g., “This takes some bravery to do.” is somehow subjective.
- Answer in 3rd person point of view.
 - E.g., “We raise cattle on this farm.” -> “They raise cattle on this farm.”
- Don’t answer about how’s the video being recorded, the camera person, or the camera itself. Answer about the content itself. Ignore what the camera person is doing.

- E.g., “There are two people, i.e. a running child, and the cameraman.” / “The camera is moving fast.” are not good answers.
- When you enter numbers, please enter digits instead of text.
 - “Seventeen” → “17”
- Use your best judgment.

3. How to provide video evidence

- The video evidence consists of all the frame intervals of the video that support the answer to your given question.
- You need to provide at least one video evidence clip (interval within the video) for each question.
- You need to provide both the start point and end point for all the video evidence you identify in the video;
- You can use your mouse or \leftarrow/\rightarrow key to click or drag the process bars of the start and end points. When you click or drag the bar, the above video will change accordingly so that you can locate the points according to the video screen.
- For each video evidence clip, the end point should be greater than zero, and the end point should be greater or equal to the start point.
- The video evidence clips (the time gap between the start and end points) should only cover the actual evidence and not more (in other words, it should be as short as possible).

A.1.3 Annotation Interface

Figure A.1 shows the annotation interface for Phase 1. Figure A.2 shows the annotation interface for Phase 2.


A.1.4 Pilot Study Comparison between Annotations from Experts vs. Non-Expert

Before the formal annotation, we compare the non-experts’ and experts’ annotations for both phases. For Phase 1, we randomly selected 45 videos from each domain to be annotated by experts and crowdworkers. Following [29], we set the AWS annotation qualification as HIT approve rate $>92\%$, the number of HITs approved >1000 , the location is either Canada or U.S., and the reward as \$6/HIT (around \$9/h).

Video Question Answering

▶ Instructions

Video 1



▶ 0:00 / 1:03

Please carefully read the [instructions](#) before performing the task.
Watch this video while you listen to its [audio](#) (you may need to adjust your device volume).
When the video ends, do not click on any other video. Place your mouse on the video, control buttons will then appear.

Your Hypothetical Objective:

Please type your hypothetical objective (or topic, intention, motivation) here

Video-1 Question-1

Your Question:

Please type your question here

Ask from the 3rd person point of view
The answer to this question should be derived from the video (visual or audio).
Questions for the same video should be **independent** of each other.
Ask questions matter-of-factly (as objectively as possible). Stick to what you can see or hear from the video.
Ask about the content itself ([not](#) the video making process, the camera-person, or the camera itself)

What is your question based on: (Check all that apply):

Scene Audio

How do you classify your question: (Check all that apply):

Motion Spatial relationship Temporal relationship Reasoning
 Number Entity Existence Time Location Other

If select 'Other', write down what type your question belongs to (N/A if cannot come up with any specific type)

Correct Answer:


Correct Answer

Provide answers matter-of-factly (as objectively as possible). Stick to what you can see or hear from the video.
Your answer should be written as **full sentences** (at least one).
Answer about the content itself ([not](#) the video making process, the camera-person, or the camera itself)

Please suggest how confident you are with your correct answer:

Very High Confidence High Confidence Moderate Confidence Low Confidence Very Low Confidence

Locate video evidences to support your question and answer:



▶ 0:00 / 1:03

The video evidence consists of all the parts of the video that support the answer to your given question.
You need to provide at least one video evidence clip (intervals within the video) for one question.
You need to provide both the **start point** and **end point** for all the video evidences you identify in the video.
You could use your mouse or ←/→ key to **click** or **drag** the **process bars** of start point and end point.
For each video evidence clip, the end point should be **greater than zero**, and the end point should be greater or equal to the start point.
The video evidence clips (the time gap between the start point and the end point) should be as **short as possible**.

Start point: ●
End point: ●

Click here to Add one more video evidence

Please finish above fill-in and classification tasks.

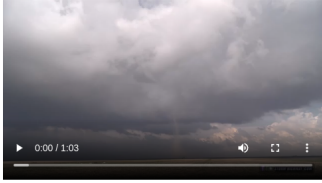
click here to Add one more question for this video

Figure A.1: Interface for annotation Phase 1. After watching the video, annotators provide a **motivation**, ask **questions**, and provide corresponding **answers** by filling in the blank. They provide parts of the videos as **evidence** to support each of the question-answer pairs by dragging the moving bar.

Video Question Answering

▶ Instructions

Video 1



Please carefully read the [instructions](#) before performing the task.
Watch this video while you listen to its [audio](#) (you may need to adjust your device volume).
When the video ends, **do not** click on any other video. Place your mouse on the video, control buttons will then appear.

Question-1:

Which direction does the landspout bend toward?

Question-1 Answer-1

Your Answer:

Please type your answer here

500/500 characters remaining

Answer in 3rd person point of view.
Respond matter-of-factly (as objectively as possible). Stick to what you can see or hear from the video.
Your answer should be written as **full sentences** (at least one).
Answer about the content itself (not the video making process, the camera-person, or the camera itself)

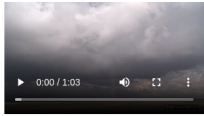
What is your answer based on: (Check all that apply):

Scene Audio

Please suggest how confident you are with your answer:

Very High Confidence High Confidence Moderate Confidence Low Confidence Very Low Confidence

Locate video evidences to support your answer:



The video evidence consists of **all** the frame intervals of the video that support the answer to your given question.
You need to provide at least one video evidence clip (interval within the video) for one question.
You need to provide both the **start point** and **end point** for all the video evidences you identify in the video;
You can use your mouse or `- / _` key to **click or drag** the **process bars** of start point and end point.
For each video evidence clip, the end point should be **greater than zero**, and the end point should be greater or equal to the start point.
The video evidence clips (the time gap between the start point and the end point) should only cover the actual evidence and not more (in other words, it should be as short as possible).

Start point:

End point:

[Click here to Add one more video evidence](#)

Please finish above fill-in and classification tasks.

[click here to Add one more answer for this question](#)

Figure A.2: Interface for annotation Phase 2. After watching the video and given the question from Phase 1, annotators provide **answers** with the corresponding **evidence**.

	Relevance	Interestingness	Professionality	Overall Score
expert	2.7	2.5	2.1	2.4
crowd	0.8	0.7	0.5	0.7

Table A.1: Average scores of the pilot study for Phase 1 (from 0 to 3).

	Objective	Question	Answer
E	Precipitation	What types of precipitation are occurring?	Rain and hail.
C	Very like	Nice	Nice
E	I want to learn about the people.	What type of weapons are they carrying?	M4's
C	The soldiers are caught on the ship.	What are they doing in this video?	They caught the ship.
E	Storm	Where is the storm?	In a field.
C	Motivation	5	Very amazing

Table A.2: Examples in pilot study for Phase 1. **E**: Expert; **C**: Crowd

After annotation, two researchers who do not know the source of annotation evaluate and score in terms of Relevance, Interestingness, and Professionality for each annotation from 0 to 3. We define Relevance, Interestingness, and Professionality as follows:

- **Relevance**: how relevant a question and an answer are to the video. Good relevance indicates that the question is related to the video and focuses on the central events, objects, or people in the video. A relevant answer should address the question and can be derived from this video.
- **Interestingness**: whether the question interests you. In other words, given a video, whether you are interested in the question and answer.
- **Professionality**: how detailed and precise the question and answer are. Good professionalism can be demonstrated by the exact usage of terminologies and numbers and an accurate answer description.
- **Overall Score**: the average score of the score for Relevance, Interestingness, and Professionality.

For each category, the higher the score, the better the annotation demonstrates that characteristic. Table A.1 lists the scores, and Table A.2 presents some annotation examples. From the empirical and numerical results, we could see a significant quality gap for annotation from experts versus crowdworkers. Therefore, we decided to employ domain experts for Phase 1.

Annotator	R1	R2	RL	IOU-F1
Expert	23.63	8.05	21.22	12.24
Crowd	20.03	3.24	17.69	8.50

Table A.3: ROUGE and IOU-F1 scores for the pilot study in Phase 2. Note that the scores here are lower than those for the human baselines in Tables 3.4 and 3.5. This is because we only compare the collected answers to a single answer here, while in Tables 3.4 and 3.5, we calculate the average scores of one annotator against the remaining as described in Section 3.4.

For Phase 2, we randomly select 104 *Geography* videos and questions from the questions annotated in Phase 1 to be annotated by both experts and crowdworkers. Moreover, we set the reward as \$3/HIT(around \$9/h) and employ the AWS **Master**¹ as the crowdworkers. Table A.3 lists the pilot study results for Phase 2. According to Table A.3, crowdworkers perform similarly to experts in Phase 2. Considering the annotation efficiency, we employ experts and crowdworkers to annotate more diversified answers for each question in Phase 2. Note that the ROUGE scores in Table A.3 are lower than the scores for the human baselines in Tables 3.4 and 3.5. This occurs because we only compare the collected answers to a single answer in Table A.3, while in Tables 3.4 and 3.5, we calculate the average scores of one annotator against the remaining as described in Section 3.4.

A.1.5 Question and Answer Correction

After we collected annotations from Phase 1, we had a group of researchers check the quality of the collected questions and answers and modify the questions and answers accordingly. Specifically, we:

- Delete the questions that somebody could answer without watching the video (e.g., Q: “If water can get through the hut’s roof; can the wind go through the hut’s roof?”, A: “Yes the wind can go through the hut’s roof.”)
- Modify the question or the answer to 3rd person view (e.g., change Q: “Do we have aircraft that we can do a touch and go landing like a helicopter?” to Q: “Do they have aircraft that can do a touch and go landing like a helicopter?”)
- Exclude the man holding the camera in the answer if it is a first-person view video.

¹https://www.mturk.com/worker/help#what_is_master_worker

Annotator ID	Expertise	Assigned Domains (# Q)
0	Geography	Geography (94) ; Natural Disaster (187)
1	Geography	Geography (16) ; Human Survival (74)
2	Veteran	Military (26) ; Human Survival (146)
3	Veteran	Military (70) ; Human Survival (89)
4	Veteran	Military (12)
5	Veteran	Military (8)
6	Veteran	Military (85)
7	Biology	Agriculture (88)
8	Biology	Agriculture (21)

Table A.4: Information about the expert annotators who annotate the questions, together with their assigned domains and the number of questions (# Q) in the parentheses.

- Modify questions that are not independently asked (e.g., “Where are they?”, where “they” refers to the “paved and unpaved roads” in the previous question. Therefore, we change the question to “Where are the roads?”)
- Split questions that include multiple sub-questions into several questions.

Some of the annotators from Phase 2 do not annotate any evidence (leaving the evidence from the start to the end of the video). Thus, we empirically filter out evidence longer than 1/4 of the video.

A.1.6 Annotator Information

Table A.4 shows the expertise of each expert, together with their assigned domains of annotation and the number of questions they annotate in their assigned domains in Phase 1.

A.1.7 Dataset Analysis

Figure A.3 presents question distributions in terms of words.

Questions Types. Table A.5 examines the frequent words for each domain, demonstrating the domain’s characteristics. Take *Natural Disaster* as an example; the three most frequent words are used in 20.63% of sentences. Besides, Fig. 3.4 in Section 3.3 lists the annotators’ self-reported question types. We observe that questions that start with “What” possess a large proportion of all the questions. Such questions might be hard to classify into certain question

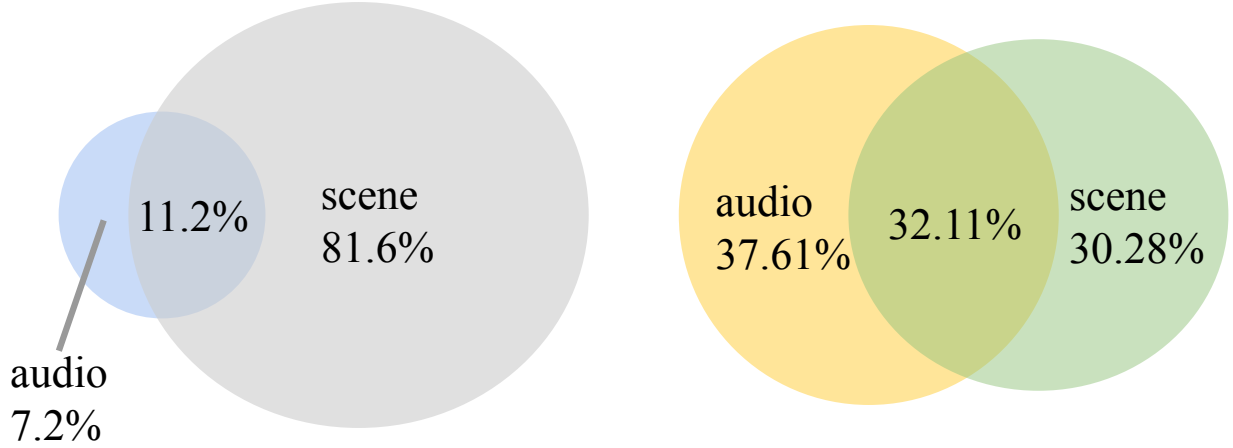


Figure A.4: Venn diagrams showing whether the question depends on visual (scene) or audio from the original video. The left is for the entire dataset, while the right is for the *Agriculture* domain.

types [26], so we allow annotators to choose multiple question types for a single question. Empirically speaking, questions that start with “is(are)”/“where”/“how many” are commonly relevant to “Existence”/“Location”/“Number” questions. In our dataset, their distribution trend (“is(are)”: 24.13% > “where”: 7.21% > “how many”: 4.48%) is akin to the trend of the distribution of the reported question types (“Existence”: 45.20% > “Location”: 12.23% > “Number”: 4.59%). Moreover, although we have “human”, “man” and “people” as the most frequent words in some domains, the most frequent words in domains such as *Military* are “military”, and “aircraft”, which demonstrates that our dataset does not only focus on human interactions as most of the existing datasets do.

Information Needed. As shown in the left Venn figure in Fig. A.4, generally, most questions are based on the visual (scene). The distribution of the question types also justifies such a distribution. The dominant kinds of questions we have in Fig. 3.4 are *Motion*, *Spatial*, *Existence* and *Entity*, which typically focus on visual information. However, in *Agriculture* (the Venn diagram on the right in Fig. A.4), the audio-based questions take more portion because videos in *Agriculture* usually focus on farming tips, instructions for using tools, etc. In this work, we do not experiment with models that use audio or transcripts from the video. Future research might look into letting models use audio and transcripts on our dataset.

Answer Similarity/Diversity. We have similar and diversified answers collected in our dataset. Figure 3.5 gives two examples: answers from the upper example are alike; for the lower example, answers vary greatly between Phase 1 and Phase 2 annotations or even

Videos	369
Duration (s)	71.22 ± 26.47
<hr/>	
Questions	916
Question per video	2.48 ± 1.38
Question length (#tokens)	7.09 ± 2.60
Answer length (#tokens)	8.62 ± 8.90
<hr/>	
Evidence per answer	1.53 ± 0.76
Evidence length (s)	9.09 ± 13.45

Table A.6: Annotation statistics for Phase 1. “#tokens” represent the number of tokens.

Crowd annotated answers	932
Expert annotated answers	182
Total	1114
<hr/>	
Answer per question	1.22 ± 0.69
Answer length (#tokens)	9.45 ± 7.46
<hr/>	
Evidence per answer	0.89 ± 0.72
Evidence length (s)	10.43 ± 5.81

Table A.7: Annotation statistics for Phase 2. “#tokens” represents the number of tokens.

within Phase 2. However, all of the answers are acceptable, given the video. The similarity demonstrates the reliability of the Phase 2 annotation. Meanwhile, the diversified answers help to evaluate models better.

A.2 Annotation Statistics

Tables A.6 and A.7 list the statistics for annotation in Phase 1 and Phase 2, respectively.

A.3 Details of Multi-task Learning

Tables A.8 and A.9 report the model performances under different sets of α, β for Eq. (3.1). We highlight the rows we report in Table 3.4 in Section 3.4.2, Table 3.4 in Section 3.4.2, Table 3.5 in Section 3.5.2, and Table 3.5 in Section 3.5.2.

β	R1	R2	RL	IOU-F1
0.5	33.8 \pm 0.8	18.5 \pm 0.7	32.5 \pm 0.8	3.7 \pm 2.4
1.0	32.2 \pm 0.7	17.6 \pm 0.5	31.0 \pm 0.6	1.9 \pm 1.7
1.5	33.8 \pm 0.3	18.0 \pm 0.9	32.5 \pm 0.3	1.5 \pm 0.1

Table A.8: Multi-task parameter selection results for the Evidence Selection SE method. We set $\alpha = 1$ throughout all the experiments and report the corresponding $\text{Multi}_{T+V,SE}$ performances on Video QA (ROUGE scores) and Video Evidence Selection (IOU-F1 scores). We highlight the row we report in Table 3.4 in Section 3.4.2 and Table 3.4 in Section 3.4.2.

β	R1	R2	RL	IOU-F1
0.5	34.0 \pm 0.5	18.8 \pm 0.7	32.8 \pm 0.6	1.2 \pm 0.1
1.0	33.4 \pm 0.6	18.4 \pm 0.2	32.1 \pm 0.6	1.4 \pm 0.3
1.5	32.8 \pm 0.3	18.3 \pm 0.3	31.7 \pm 0.2	1.0 \pm 0.2

Table A.9: Multi-task parameter selection results for the Evidence Selection SE method. We set $\alpha = 1$ throughout all the experiments and report the corresponding $\text{Multi}_{T+V,IO}$ performances on Video QA (ROUGE scores) and Video Evidence Selection (IOU-F1 scores). We highlight the row we report in Table 3.5 in Section 3.5.2 and Table 3.5 in Section 3.5.2.

A.4 Experiment Results

Figures A.5 and A.6 report Multi-Task model’s performance on Video QA by ROUGE-2, and ROUGE-L, respectively. Figure A.7 demonstrates that ROUGE scores follow a similar trend as mentioned in Section 3.5.3.

A.4.1 Ablation Study on Video Evidence Selection

To investigate whether baseline models indeed need the vision part for the Video Evidence Selection task, we conduct an ablation study using T5 IO and T5 SE (introduced in Section 3.5). We take a random sequence of the same length as the original video sequence and feed the random sequence instead of the original video sequence to the model. Table A.10 shows the comparison results between these different settings. T5 IO performs roughly the same as $\text{T5 IO}_{\text{random}}$, which indicates that the model struggles to utilize visual information. T5 IO even underperforms the random baseline which can achieve an IOU-F1 score of 2.5 ± 0.3 (as shown in Table tab:few-shot-evidence-results-10-epochs). However, T5 SE outperforms $\text{T5 SE}_{\text{random}}$, suggesting that T5 SE uses visual features to locate the evidence of the question.

Model name	IOU-F1
T5 IO _{random}	1.1 ± 0.3
T5 IO	1.1 ± 0.2
T5 SE _{random}	2.7 ± 1.9
T5 SE	4.5 ± 0.8

Table A.10: Ablation study on the Video Evidence Selection. We feed T5 IO_{random} and T5 SE_{random} the question concatenated with a random sequence, while we feed T5 IO and T5 SE the question with the actual video sequence.

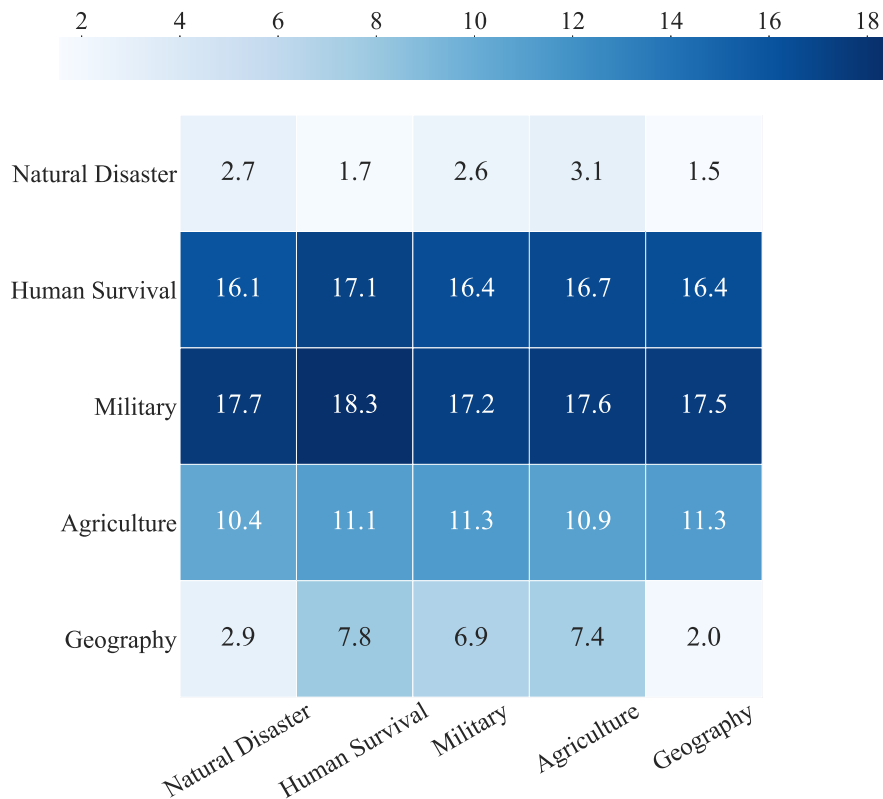


Figure A.5: Multi-Task ROUGE-2 scores for Video QA when tuned on a single domain (y-axis) and tested against each domain (x-axis).

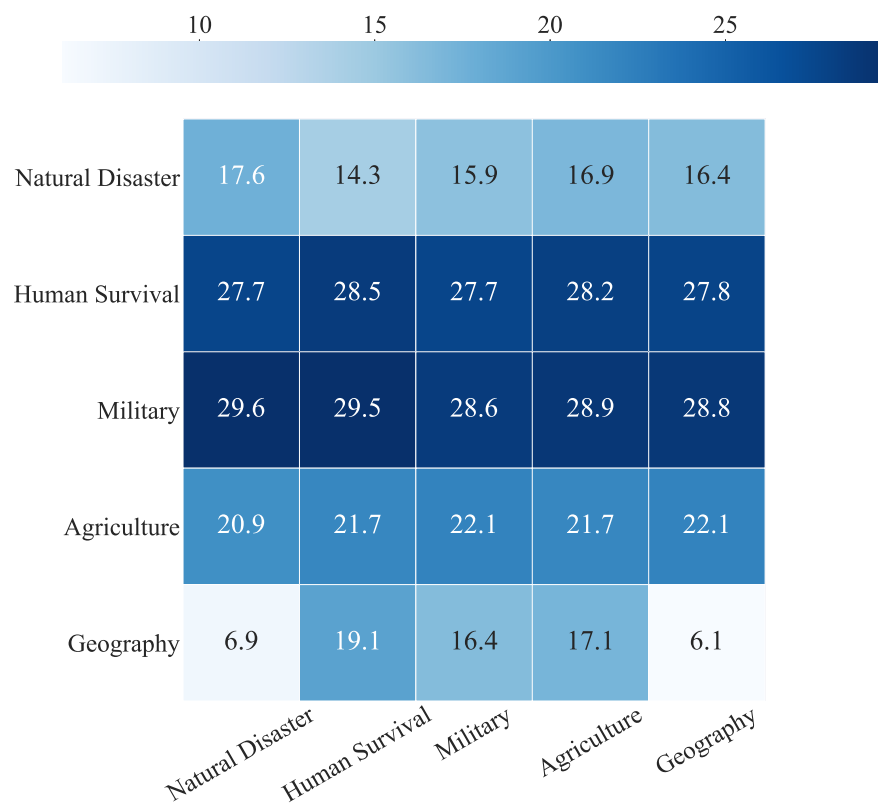


Figure A.6: Multi-Task ROUGE-L scores for Video QA when tuned on a single domain (y-axis) and tested against each domain (x-axis).

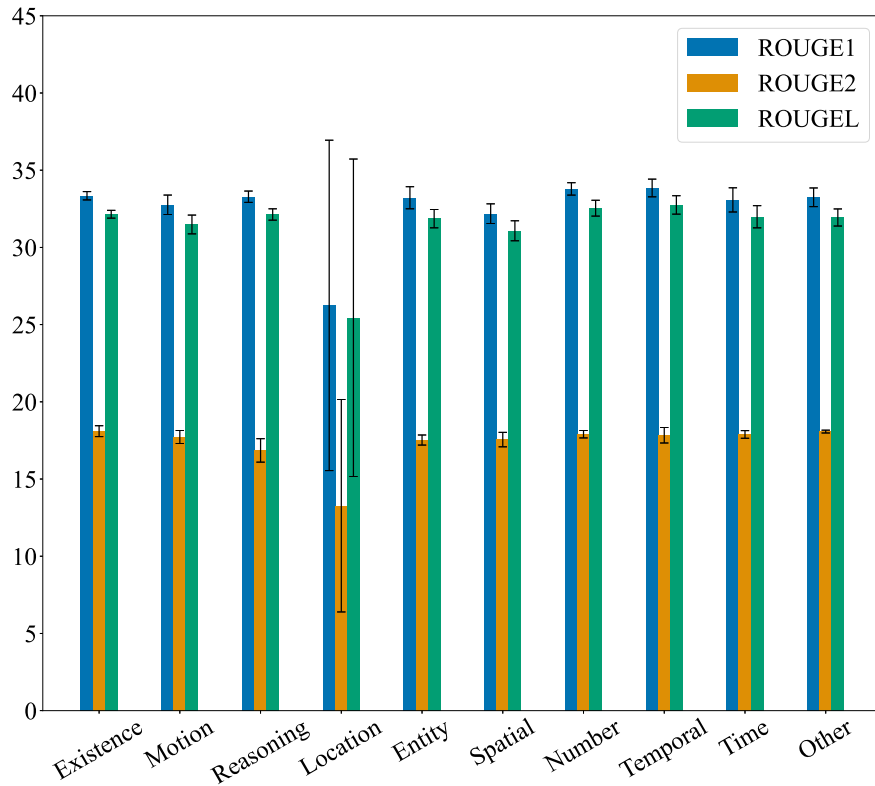


Figure A.7: $\text{Multi}_{T+V,SE}$ performance on different question types for Video QA. We report ROUGE-1, ROUGE-2, and ROUGE-L scores from left to right for each question type. Different ROUGE scores follow similar trends; therefore, we only report ROUGE-1 in Table 3.6 in Section 3.5.3.

APPENDIX B

Realistic and Robust Video Understanding Evaluation: Supplementary Material

B.1 Dataset

B.1.1 Most-Frequent Noun Phrases

We report the most frequent noun phrases in the original labels and in the annotations we collected, in Fig. B.1. The most frequent nouns in both answer sets tend to refer to people, which makes sense considering the videos’ content. In the annotation data, we see a greater variety of synonyms for the same kind of person (“male”, “man”, “guy”), likely due to the task definition, which encourages paraphrasing.

B.1.2 Part-of-speech Distribution

We compare the use rate of words in different part-of-speech categories for the initially blanked phrases and the annotations, using the same parser specified earlier to label part-of-speech tags in the noun phrases. We show the distributions in Fig. B.2, and we see that the annotations have roughly the same rate of part-of-speech tag use in all categories, except among adjectives and pronouns where the initially blanked phrases have a higher rate of use. This outcome is likely an artifact of the data collection strategy, which encouraged annotators to generate unique noun phrases rather than phrases with adjectives or pronoun references.

B.1.3 Part-of-speech Sequence Distribution

Although the candidate answers collected from crowd workers consist of noun phrases, they may include different part-of-speech (POS) sequences within the noun phrases. The distributions of POS sequences in Fig. B.3 show that the annotators tended to write “bare” nouns without extra determiners and proper nouns more than the original phrases. This

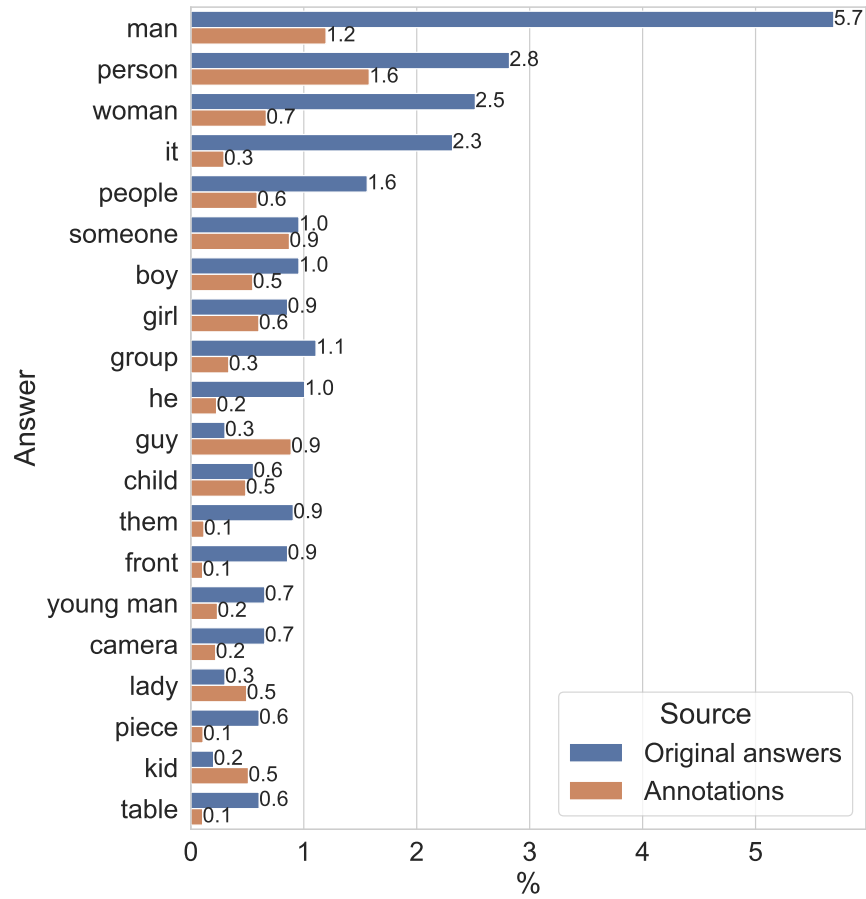


Figure B.1: Top 20 nouns for the originally blanked phrases and the annotations in the validation and test data.

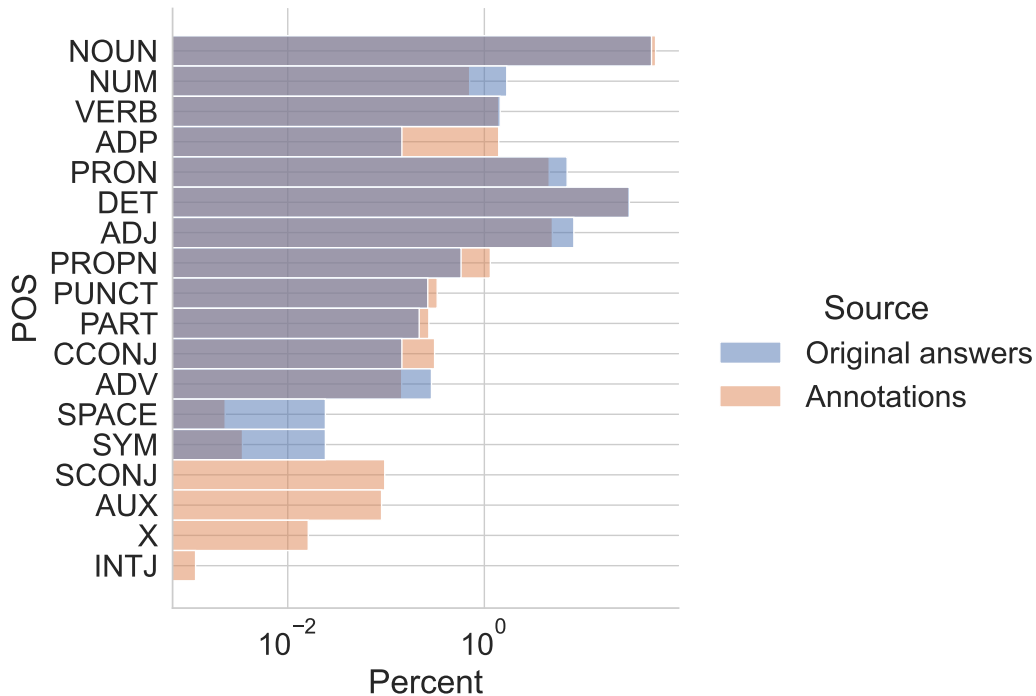


Figure B.2: Relative frequency of part-of-speech tags in the originally blanked phrases and the annotated answers.

phenomenon makes sense, considering that the task asked annotators to provide many unique nouns without consideration for the nouns’ structure.

B.1.4 Dependency Categories

Due to the sampling process, some of the answers occur in different syntactic contexts, e.g. in a prepositional phrase in “A woman does push-ups on _____” or as a subject in “_____ at a driving range demonstrating...” (see Table 5.1). We plot the distribution of dependency categories in Fig. B.4, showing that nouns occur in a wide range of positions but mostly in preposition, subject, and direct object positions.

Next, we test whether specific syntactic contexts tend to attract more answers from the annotators than others by computing the mean unique number of answers per annotator within each syntactic context (based on the dependency parse connected to the masked NP). We show the distribution in Fig. B.5. Captions that mask noun phrases that occur in preposition (`pobj`) and direct object (`dobj`) positions tend to attract slightly fewer unique answers per annotator than the runner-up most-frequent categories, subject (`nsubj`) and compounds (`compound`). This result intuitively makes sense since annotators would likely have fewer options for noun phrases when faced with a preposition or a direct object than

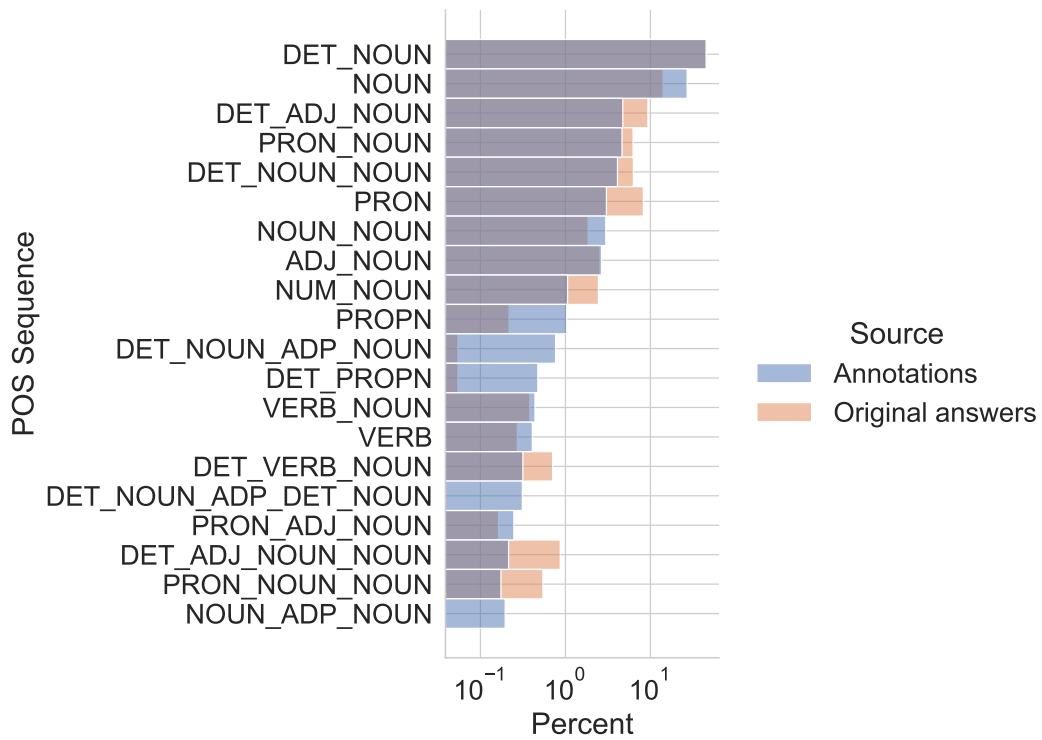


Figure B.3: Relative frequency of POS tag sequences in the originally blanked phrases and the annotated answers.

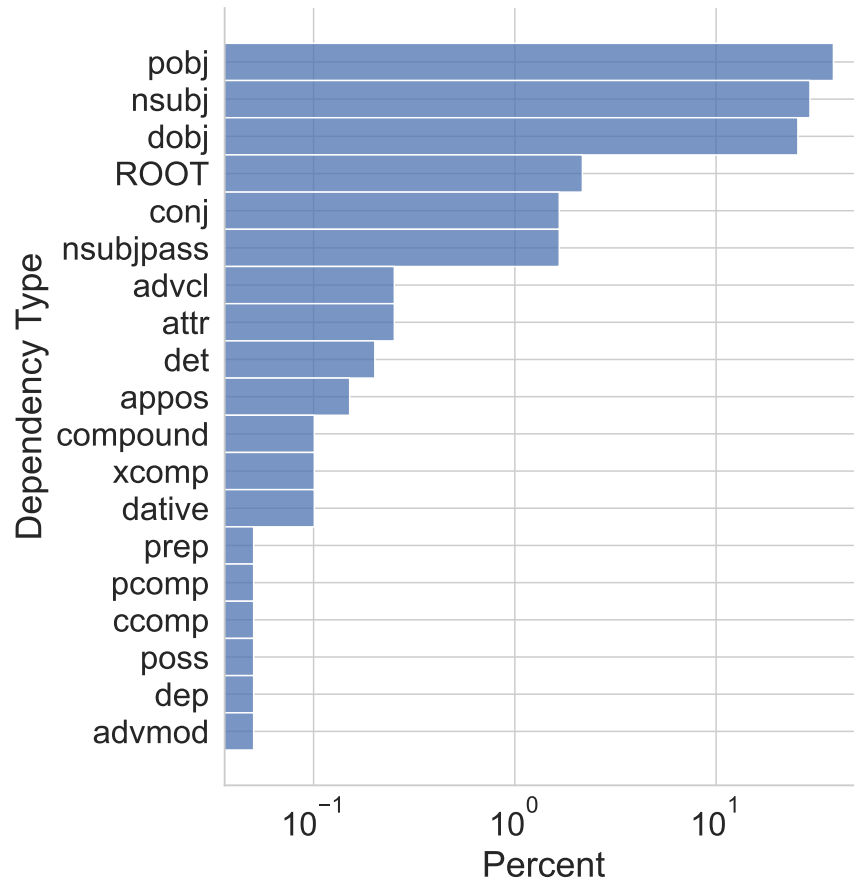


Figure B.4: Relative frequency of dependency types for the root token of the original blanked phrases.

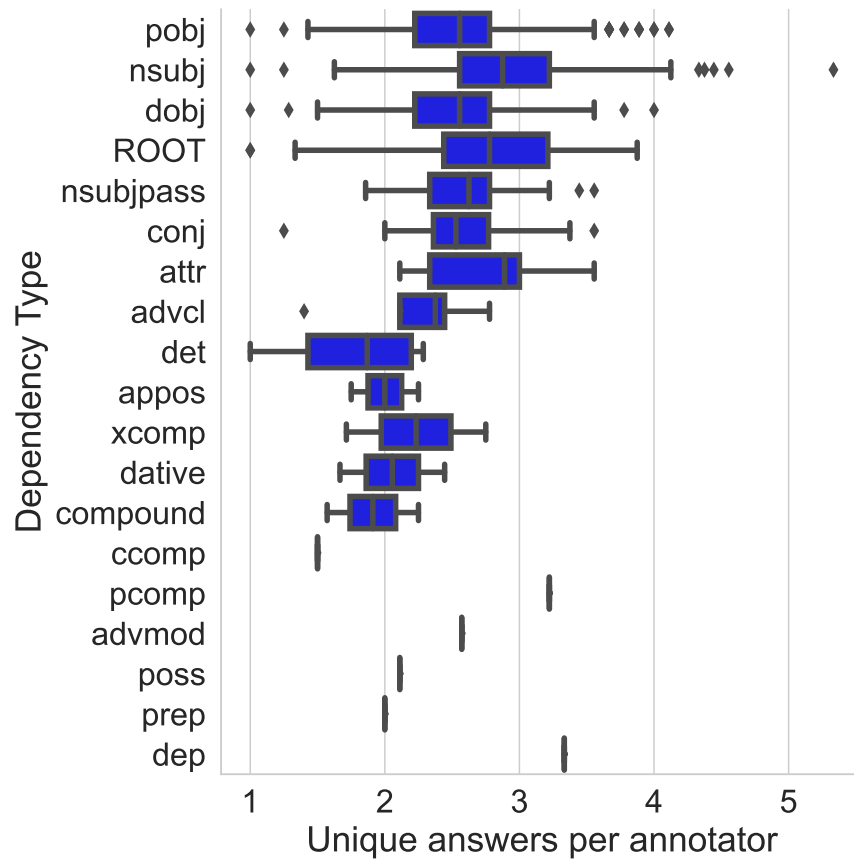


Figure B.5: Average number of unique answers per caption, grouped by the dependency type of the root word of the originally blanked phrases. We sort the categories by their frequency.

with the less restrictive subject noun position.

B.1.5 Gender Representation

Language processing models can often learn to encode social bias due to non-representative training data, such as image captions for photos of men and women taken in stereotypical environments [275]. We find a slight gender gap in our data. Using a gender word list, we find that about 10.9% of the originally blanked phrases are male-related words in contrast to 6.2% that are female-related, and 9.1% of the annotations are male-related while 5.9% are female-related. We note that the gender imbalance is less severe for the annotations than the original phrases, and the annotations use more gender-neutral human words than the labels (6.6% for annotations vs. 6.0% for original phrases). While some of the annotators may undoubtedly be biased in their decisions, some of the bias may also result from the original video clips. We acknowledge this limitation as a direction for future work collecting video caption data.

We used the following lists for gendered words, which were chosen to be in similar semantic categories (e.g. male “brother”, female “sister”, neutral “sibling”):

- Male-oriented words: “boy”, “brother”, “father”, “guy”, “he”, “him”, “himself”, “his”, “male”, “man”, “son”
- Female-oriented words: “daughter”, “female”, “girl”, “her”, “herself”, “lady”, “mother”, “she”, “sister”, “woman”
- Gender-neutral words: “adult”, “baby”, “child”, “human”, “kid”, “parent”, “people”, “person”, “sibling”

B.1.6 Spatiotemporal Trends of the Blanked Entities

One of the authors of the original paper related to this chapter randomly sampled 50 videos to analyze spatiotemporal information on the blanked entities. Figures B.6 to B.8 show trends on where, when, and for how long the blanked entities appear in the videos. As expected, the blanked entity generally appears at the center of frames, with a slight tendency to be on the lower side. We observe that around 93% of the time, the blanked entity appears between seconds 2 and 4 of the video but that there is still a high chance (75%) of seeing it at any given moment. 68% of the time, the blanked entities appear for the entire duration of their corresponding video.

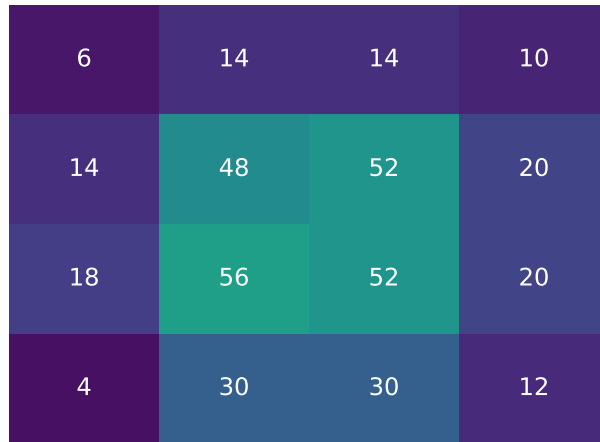


Figure B.6: Heat map showing how frequently (%) the blanked entity appears within a given location of the video, for a sample of 50 videos. Each frame is divided into a four-by-four grid. A blank entity is counted for a given cell if it touches the cell at any moment of a given video. Note that multiple cells can be counted for a given video because the entity is big enough or because the entity or the camera moves.

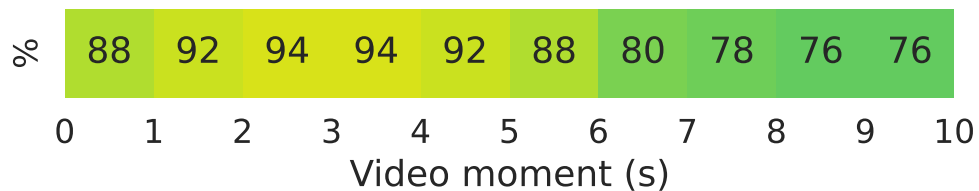


Figure B.7: Frequency (%) that the blanked entity appears at each one-second interval in a given video, for a sample of 50 videos. A time interval is counted if the entity appears at any moment of the one-second duration interval.

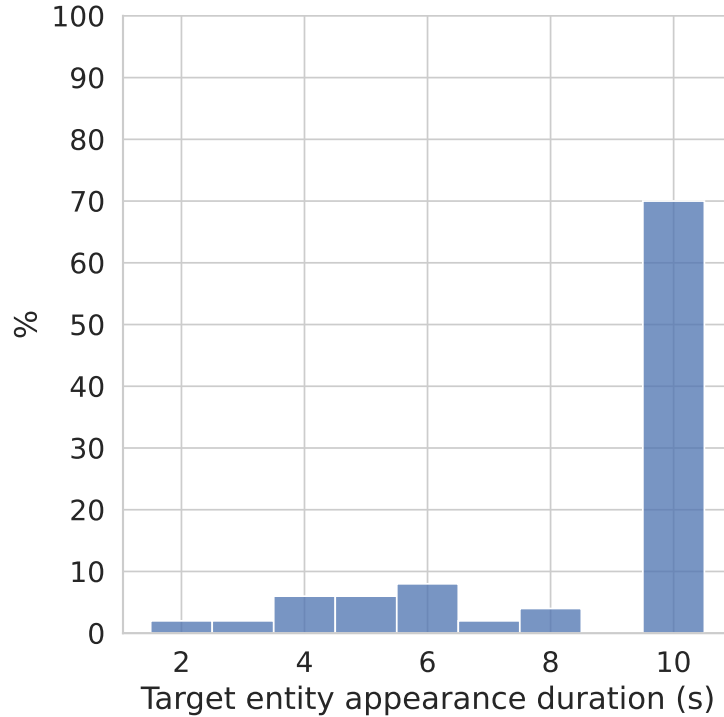


Figure B.8: Distribution of the total time that each blanked entity is seen within its video for a sample of 50 videos.

B.2 Experiments and Results

B.2.1 More Implementation Details

We use the T5 model from the HuggingFace Transformers library [246]. We train the model with Adam [113] on a V100-16GB with a batch size of 64 for ten epochs (4,000 steps) using a learning rate of $1e-4$ with a warm-up of one epoch and a linear decay. The training time is short, less than an hour. We compute the loss as the cross-entropy between the model-generated output and the initially blanked phrase.

For test-time decoding, we use beam search with a beam size of 4 for the early-fusion model and 8 for the late-fusion one, with a maximum token length of 10. We stop the decoding early if an example has seen as many complete hypotheses as the beam size (beam search early-stopping¹). We penalize the repetitions of bigrams within a decoded text. For each example, we choose the first noun phrase beam, as detected by spaCy [87], or the first if none are noun phrases. We show the effect of varying the beam size in Appendix B.2.2. Modifying the beam search early-stopping property does not lead to significant performance changes.

¹https://huggingface.co/transformers/internal/generation_utils.html#transformers.BeamSearchScorer

	1	2	4	8
T5 fine-tuned	72.9	74.2	73.8	73.8
T5 + I3D	73.0	74.0	74.3	74.2
Late-fusion T5 + I3D	69.0	69.6	69.7	69.7

Table B.1: F1 scores on the validation set for the beam sizes 1 (greedy search), 2, 4, and 8.

	EM	F1
<code>t5-small</code>	20.2	37.1
<code>t5-base</code>	34.9	50.2
<code>t5-large</code>	43.5	59.5
<code>t5-3b</code>	44.9	62.6

Table B.2: Results on the validation set for different model sizes of the T5 text-only zero-shot model.

B.2.2 Beam Search

Table B.1 shows the effect of varying the beam size during the beam search decoding. In all cases, using a beam search of at least size two is better than a greedy search. However, the results are marginally better or inconclusive when using beam size four or eight. This behavior is probably related to the phenomenon described by Meister et al. [155] in which beam search does get us closer to the true maximum a posteriori solution. Still, the answers start to get worse after a certain point.

B.2.3 Model Size

In Table B.2, we show the result of changing the T5 model size for the text-only zero-shot baseline. We could not fit the model variant `t5-11b` into GPU memory. As expected, we note an increase in the evaluation metrics as the model capacity increases.

B.2.4 Qualitative Analysis

In Table B.3, we show several examples of answers correctly predicted by the best multimodal method but incorrectly answered by the best text-only method. Even though the answers provided by the text-only method are plausible by just looking at the text, they do not make sense with the given videos. In the second example, one can quickly tell the person is not at a gym but in some kind of indoor room. For these examples, the multimodal method seems




			
	A person at the top of _____ with ropes hanging down.	A guy is by the stairs in _____ doing the moonwalk in socks.	A man is showing and describing a rock sample to _____.
correct answers	adirondacks, cliff, climb, frozen waterfall, gully, hill, ice, icy cliff, ledge, mountain , ravine, slope, snow	building, doors, entryway, foyer, his home, his house, home, house, living room, room , shorts, t-shirt	audience, camera , consider where its hinge goes, describe how it looks, discuss its hinge, explain his viewers, his audience, his followers, his subscribers, his viewers, people, students, viewer, viewers
T5 fine-tuned	a tree (0)	a gym (0)	a woman (0)
T5 + I3D	a mountain (100)	a room (100)	a camera (100)

Table B.3: Examples of instances correctly predicted by the best multimodal method but incorrectly predicted by the best text-only method. The F1 score obtained by each answer is shown in parentheses. We show the correct answers normalized and separated by commas. We show the model predictions verbatim. From each video, we show a single frame illustrating the key moment.

to have identified what is visually relevant.

APPENDIX C

Practical and Scalable Video Understanding with a Single Model: Supplementary Material

C.1 Pretraining Datasets

Table C.1a summarizes existing datasets for pretraining visual-language models. CC3M [206] is one of the first datasets to bridge images with natural language supervision leveraging the internet (HTML image alt texts). This dataset collects about 3M clean images through a pipeline that guarantees a clean supervision signal. The MS COCO Captions [32] (COCO) dataset contains 500k human-curated caption-image pairs. The images come from the MS COCO [141] dataset, which were collected from Flickr. WIT [214] contains 37.5M image-caption pairs obtained from the Wikipedia. CLIP authors [183] constructed a dataset with more than 400M text-image pairs scrapped from the internet. The dataset contains images from queries formed with the 1000 most common visual concepts in Wikipedia. While the dataset does not rely on manual cleaning to verify the image-text pairs, it is assumed that a person provided a good enough image caption before uploading it to the internet. In the same spirit, the WebVid-2.5M dataset [11] crawls 2.5M text-video pairs leveraging manually curated titles from Stock footage. Differently, the HowTo100M (HT100M) dataset [158] contains 100M pairs of noisy aligned video-text pairs. In this dataset, the video-text pairs and their automatically transcribed speech come from long YouTube videos.

C.2 FitCLIP vs. CLIP per-class performance

Previous experiments showed that FitCLIP offers a simple strategy to boost zero-shot performance in video understanding tasks; however, where are those improvements emerging from? To better understand the differences between FitCLIP and CLIP (our teacher), we compute the performance difference per class between both models in the Moments

Dataset	Domain	Supervision	Size	Dataset	# Classes	# Samples	Dataset	# Samples	Genre
COCO [32]	Images	Clean	600k	MiT [160]	339	33,900	MSR-	1000	UGC
CC3M [206]	Images	Clean	3M						
WIT [214]	Images	Clean	37.5M	UCF101 [211]	101	1,794	VTT [251]	3305	Cooking
CLIP [183]	Images	Weak	400M						
WebVid [11]	Videos	Weak	2.5M				DiDeMo [7]	4021	UGC
HT100M [158]	Videos	Noisy	100M						

(a) Pretraining datasets (b) ZS action recognition (c) ZS text-to-video retrieval

Table C.1: **Pretraining and zero-shot datasets.** (a) Diverse image and video datasets are available for pretraining visual-language models. (b) We benchmark zero-shot (ZS) action recognition in two popular datasets. MiT denotes Moments in Time [160]. (c) To benchmark zero-shot (ZS) text-to-video retrieval, we rely on three well-established datasets. UGC stands for user-generated content, and Genre refers to the type of videos in the dataset.

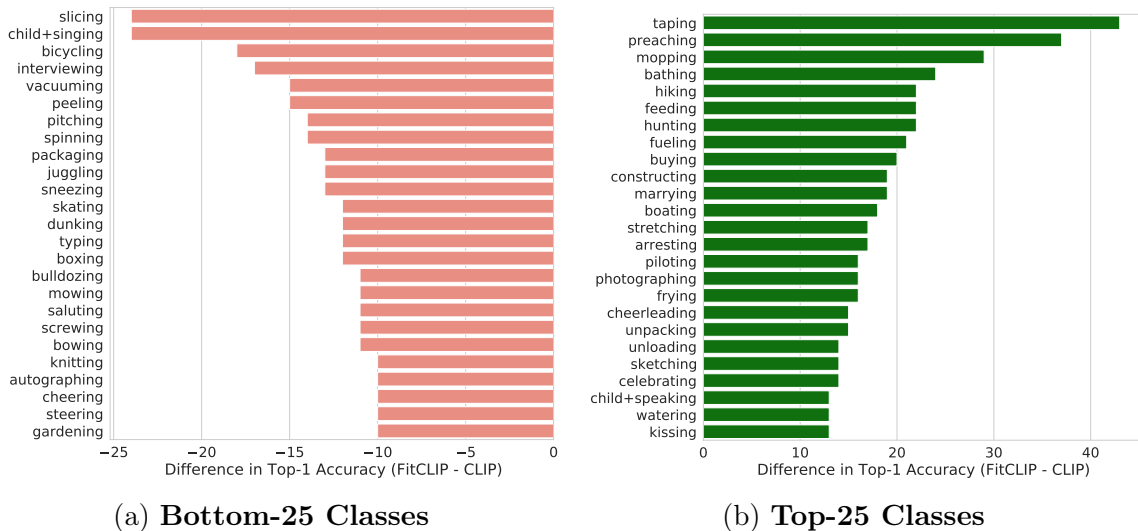


Figure C.1: **FitCLIP vs. Teacher per-class improvements.** The plots show the per-class difference between FitCLIP and CLIP performances (Top-1) on the Moments in Time (MiT) dataset. Noticeably, the performance difference varies significantly across various action classes, reinforcing our intuition that FitCLIP encodes complementary video information compared to CLIP. Interestingly, FitCLIP improves performance for abstract action classes such as *preaching and tapping*, while CLIP does so for actions involving common actions like *cycling, boxing, or skating*.

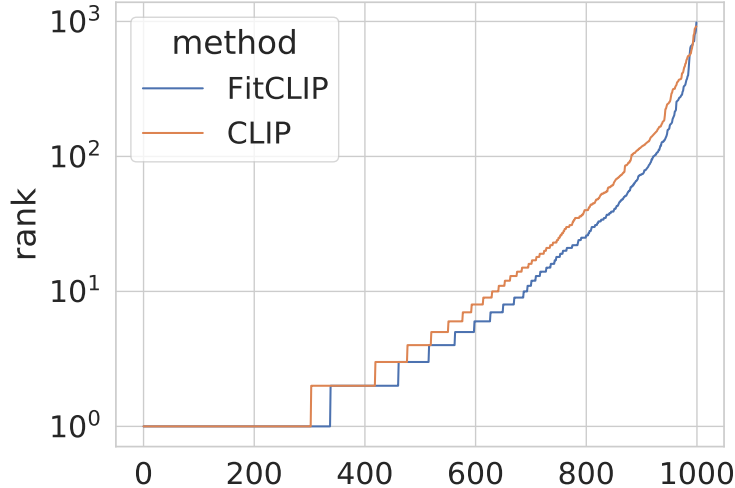


Figure C.2: **FitCLIP vs. CLIP distribution of Text-to-Video Retrieval rankings.** The x-axis represents each text in the MSR-VTT validation set (1K-A split), and the y-axis (in log scale) represents the rank each model gave to the corresponding video. We sort the x-axis by rank (the values increase).

in Time dataset. Figure C.1 summarizes the results by plotting the largest and smallest (including actions with worst performance) 25 changes in performance. First, we observe that several classes’ performance accuracy (Top-1) changes drastically. This result validates our hypothesis that the Student provides FitCLIP with complementary information concerning the knowledge CLIP (the Teacher) already provides. Interestingly, FitCLIP obtains better overall performance for abstract action classes such as *preaching and taping*. On the contrary, CLIP tends to do better for standard actions often captured in photographs such as *skating, or boxing*.

C.3 FitCLIP vs. CLIP ranking distributions

The Text-to-Video Retrieval results show that FitCLIP outperforms CLIP at multiple points of this zero-shot setting. However, it is not clear how the methods behave for the rest of them. Figure C.2 shows the rankings distribution for the validation set of MSR-VTT for both methods. We can see that FitCLIP is under the CLIP curve for virtually all points. FitCLIP ranks the videos better for this dataset, regardless of the cutting point.

Dataset	Top 1	Top 5
WebVid	11.4	27.2
CC3M+WebVid	13.2	29.3
CC3M+WebVid+COCO	14.0	31.8

(a) **Moments in Time (MiT)**

Dataset	Top 1	Top 5
WebVid	36.9	61.1
CC3M+WebVid	49.2	61.1
CC3M+WebVid+COCO	51.9	76.1

(b) **UCF101**

Table C.2: Zero-shot action recognition results of Frozen in Time [11] pre-trained on different datasets.

C.4 Frozen in Time Variants

Tables C.2 and C.3 show the results on zero-shot action recognition and text-to-video retrieval for Frozen in Time [11] on different pre-training datasets. Its authors provide these pre-trained checkpoints¹. They use different combinations of Conceptual Captions [206] (CC3M), WebVid [11], and Microsoft COCO Captions [32] (COCO). Combining the three of them presents the best results. However, note the captions in COCO Captions were obtained using an expensive data collection procedure and are richly annotated. In contrast, the other two datasets were obtained from data available on the internet and thus have weaker annotations.

C.5 Impact of Fusing the Teacher-Student Knowledge

Tables C.4 and C.5 present all the metrics for the results on the impact of our method on zero-shot action recognition and zero-shot text-to-video retrieval. Overall, FitCLIP presents the best results. We highlight the importance of fusing the teacher and student’s knowledge as they perform worse than in combination.

C.6 Alpha Value

We analyze the effect of changing the value of α necessary for the weight-space ensembling step when fusing the teacher and student knowledge in our method. Figure C.3 shows the effect of this hyperparameter by varying it from 0 to 1, with increments of size 0.1, where 0 is

¹<https://github.com/m-bain/frozen-in-time#-pretrained-weights>

Dataset	R@1	R@5	R@10	MdR
WebVid	12.9	31.0	41.2	16
CC3M+WebVid	17.1	39.1	49.6	11
CC3M+WebVid+COCO	21.3	43.6	55.9	7

(a) **MSR-VTT**

Dataset	R@1	R@5	R@10	MdR
WebVid	1.1	4.2	6.8	329
CC3M+WebVid	2.7	9.5	14.2	162
CC3M+WebVid+COCO	3.2	10.1	16.2	135

(b) **YouCook2**

Dataset	R@1	R@5	R@10	MdR
WebVid	14.5	34.9	45.4	14
CC3M+WebVid	20.3	42.7	53.5	9
CC3M+WebVid+COCO	23.2	45.8	56.8	7

(c) **DiDeMo**

Table C.3: Zero-shot text-to-video retrieval results of Frozen in Time [11] pre-trained on different datasets.

Dataset	Top 1	Top 5
Teacher (CLIP)	19.9	40.3
Student	17.7	39.1
FitCLIP	21.8	44.6
Δ	$\uparrow 1.9$	$\uparrow 4.3$
Error rate reduction	$\uparrow 2.4$	$\uparrow 7.2$

(a) **Moments in Time (MiT)**

Dataset	Top 1	Top 5
Teacher (CLIP)	74.5	94.3
Student	64.7	90.4
FitCLIP	73.3	95.3
Δ	$\downarrow 1.2$	$\uparrow 1.0$
Error rate reduction	$\downarrow 4.7$	$\uparrow 17.5$

(b) **UCF101**Table C.4: **Impact of fusing teacher-student knowledge on zero-shot action recognition.** Δ denotes the absolute difference in performance between FitCLIP and the Teacher model.

Dataset	R@1	R@5	R@10	MdR
Teacher (CLIP)	30.4	55.1	64.1	4
Student	28.1	52.6	63.7	4
FitCLIP	33.8	59.8	69.4	3
Δ	$\uparrow 3.4$	$\uparrow 4.7$	$\uparrow 5.3$	$\uparrow 1$
Error rate reduction	$\uparrow 4.9$	$\uparrow 10.5$	$\uparrow 14.8$	$\uparrow 25.0\%$

(a) **MSR-VTT**

Dataset	R@1	R@5	R@10	MdR
Teacher (CLIP)	5.3	14.6	20.9	94
Student	2.9	9.7	14.1	159
FitCLIP	5.8	15.5	22.1	75
Δ	$\uparrow 0.5$	$\uparrow 0.9$	$\uparrow 1.2$	$\uparrow 19$
Error rate reduction	$\uparrow 0.5$	$\uparrow 1.1$	$\uparrow 1.5$	$\uparrow 20.2\%$

(b) **YouCook2**

Dataset	R@1	R@5	R@10	MdR
Teacher (CLIP)	26.2	49.9	60.6	5
Student	20.7	42.4	54.0	8
FitCLIP	28.5	53.7	64.0	4
Δ	$\uparrow 2.3$	$\uparrow 3.8$	$\uparrow 3.4$	$\uparrow 1$
Error rate reduction	$\uparrow 3.1$	$\uparrow 7.6$	$\uparrow 8.6$	$\uparrow 20.0\%$

(c) **DiDeMo**

Table C.5: **Impact of fusing teacher-student knowledge on zero-shot text-to-video retrieval.** Δ denotes the absolute difference in performance between FitCLIP and the Teacher model. To measure the error rate reduction for the median rank, we directly use its reduction rate.

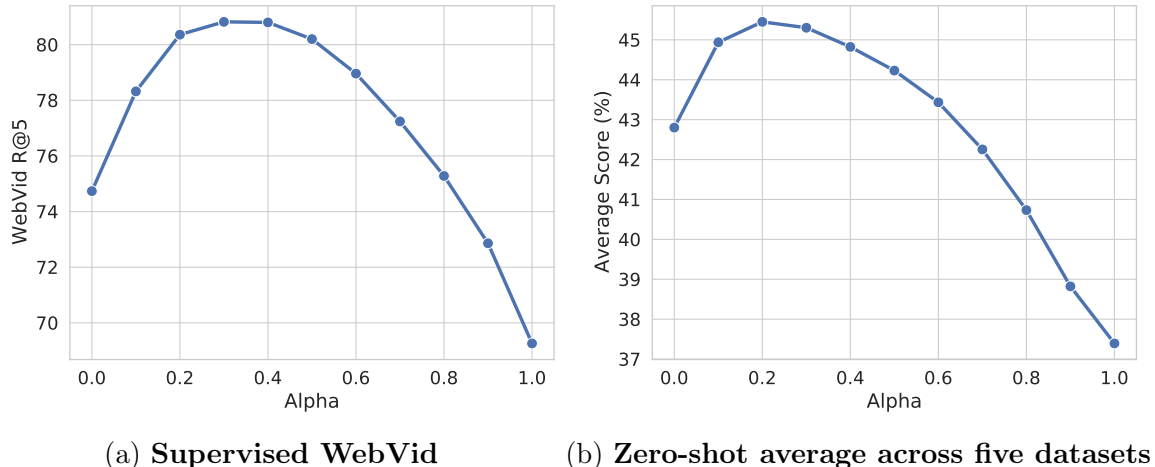


Figure C.3: **Impact of changing the value of weight-ensembling α value when fusing the teacher and the student.** We report (a) supervised text-to-video retrieval WebVid R@5 (recall we trained on this domain) and (b) an average across five other datasets. DiDeMo, MSR-VTT, and YouCook2 (R@5) are the zero-shot text-to-video retrieval datasets used. The zero-shot action recognition datasets are Moments in Time and UCF-101 (top-1 accuracy). The average value across these datasets is shown.

only the teacher, and 1 is only the student. We show the results on a different split from the training distribution (Fig. C.3a) and on the other datasets we have reported throughout the chapter (Fig. C.3b). For WebVid, we obtain the best value when $\alpha = 0.3$. Still, we decided to use $\alpha = 0.4$, close enough, and the best value obtained by [247]. The best value we obtain for the other datasets is when $\alpha = 0.2$. For $\alpha = 0.4$, the score is still high.

C.7 Impact of the Labeled Data Size

The more labeled data for training typically implies better results. However, more training means that the obtained checkpoint in the weight landscape is further away from the point of origin, making it harder for weight-ensembling to work well. We study the impact of the labeled data size and try to find a good trade-off point. Figure C.4 show the results of preliminary experiments, which are performed by fine-tuning with different subset sizes of the training set from WebVid and applying weight-space ensembling (without distillation). Each subset was sampled from the whole dataset (they are unlikely subsets of each other). We find the best value when the WebVid-2.5M training subset size is 4500. We recognize that we indirectly use other parts of WebVid, which can boost the selected subset’s in-distribution performance. However, note that this doesn’t imply better out-of-distribution performance.

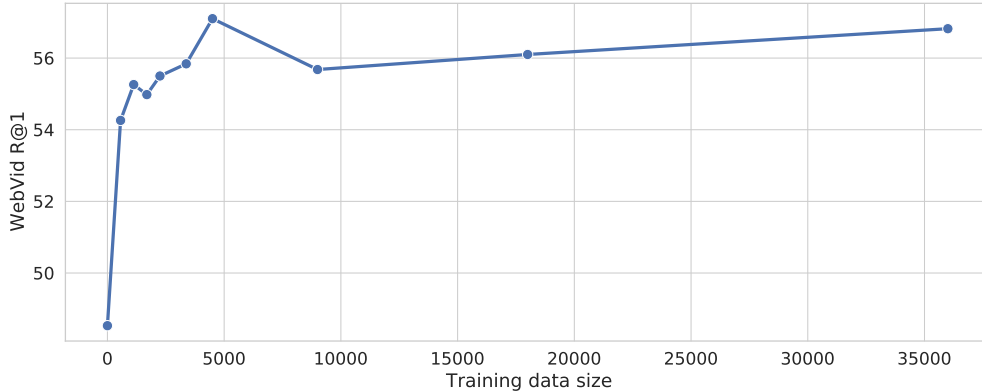


Figure C.4: **Text-to-Video top-1 recall on WebVid-2.5M (supervised) of different training subset sizes when fine-tuning CLIP ViT-B/16 and applying weight-space ensembling.** The evaluated subset sizes are 0, 563, 1125, 1688, 2250, 3375, 4500, 9000, 18000, and 36000. The subset size 0 represents the evaluation of the pre-trained model without fine-tuning. We exclude large values as we have observed a great drop in performance. Note this experiment doesn't employ distillation.

We skip showing results for large values as we have observed a remarkable drop in performance. In particular, we obtained results that were considerably worse than those obtained by the pre-trained model when using the whole training set (2.5M).

C.8 Share of Pseudo-Labels/Labels

We are interested in comparing the effect of applying weight-ensembling to a distilled model to using it with a model trained only on labeled data. Figure C.5 shows the effect of varying the proportion of the labeled loss in the final loss in our zero-shot benchmarks. The use of the distillation loss with $\lambda = 10^{-4}$ outperforms the usage of only the labeled loss in YouCook2 and UCF101 and shows similar performance on MSR-VTT. In contrast, the performance on DiDeMo and Moments in Time seems to be better when using only the labeled loss. We hypothesize our method especially benefits from datasets whose distribution is more distant from the training-time dataset (e.g., YouCook2 is significantly distant from WebVid-2.5M).

	Action Recognition		Text-to-video Retrieval		
	UCF101	MiT	MSR-VTT	YouCook2	DiDeMo
CLIP	74.5	19.9	55.1	14.6	49.9
w/o PL	72.5	22.0	59.9	15.1	55.4
FitCLIP	73.3	21.8	59.8	15.5	53.7

Table C.6: **Importance of the Pseudo-Labels.** We report the top-1 accuracy for the zero-shot action recognition datasets and the top-5 recall for the zero-shot text-to-video retrieval ones. We show in bold the best results between w/o PL (without pseudo-labels) and FitCLIP for each dataset.

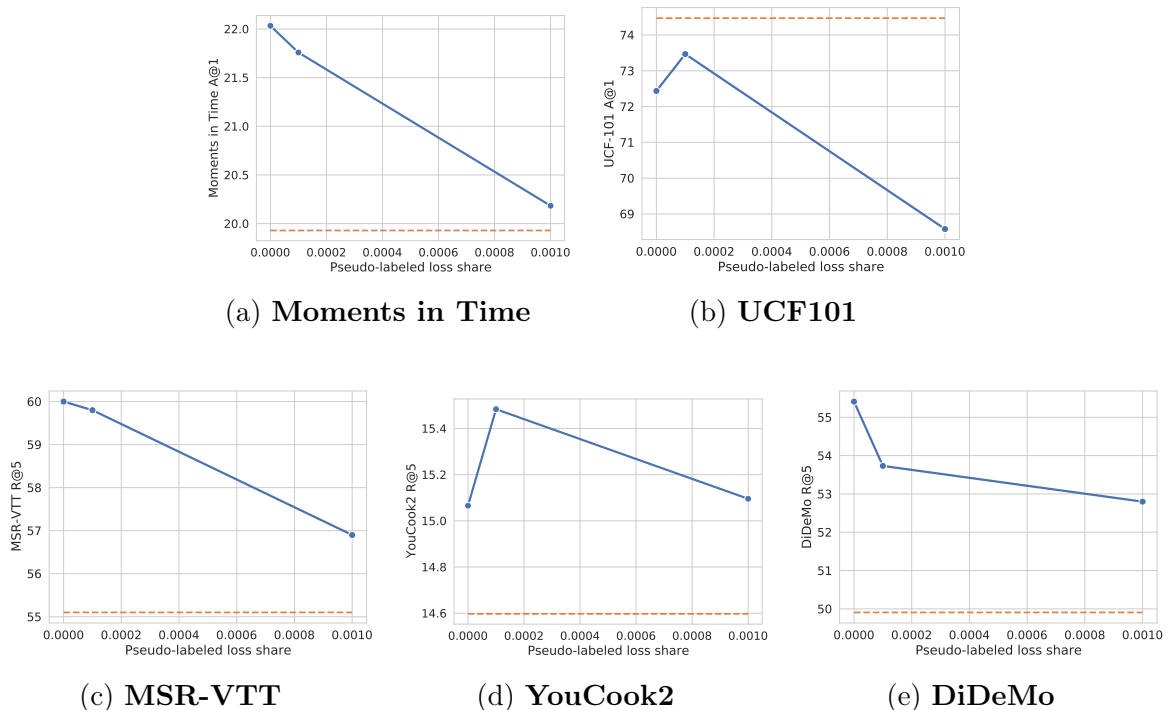


Figure C.5: **The effect on the zero-shot performance of the share of the pseudo-labeled and labeled losses in FitCLIP.** Each plot shows how the proportion of the pseudo-labeled loss (x-axis) affects the zero-shot performance on a given dataset. The dashed orange line shows the performance of CLIP as a reference. We skip the sampled values greater than 0.01 to visualize the plots better since they tend to bring a worse performance.

APPENDIX D

Compositional Generalization with Image-Text Models: Supplementary Material

D.1 SugarCrepe Fine-Grained Performance

In Table D.1, we show SugarCrepe’s fine-grained task results.

D.2 Classification without Prompts

CLIP-like models are evaluated with multiple prompts for classification, typically relying on the ones originally tested by OpenAI’s CLIP [183], as we do in this chapter. For example, there are 80 prompts (templates) used for ImageNet, such as “a photo of a {class name}” and “itap of the {class name}”. These prompts are used because the text representations are usually noisy, and a satisfactory average class representation can be obtained from the embeddings for all these texts. These prompts have been carefully crafted to match the characteristics of the classes and the dataset. In Table D.2, we show the classification results without employing any prompts, just using the class name as the input. Without patching, our method presents a little drop (2.5%) in performance with respect to the results from

	Replacement				Swap			Addition			task avg.	avg.
	Obj.	Att.	Rel.	avg.	Obj.	Att.	avg.	Obj.	Att.	avg.		
pre-trained	90.8	80.2	69.1	80.1	61.0	63.8	62.3	77.1	68.5	72.8	71.7	72.9
NegCLIP	92.6	85.9	76.8	85.1	75.6	75.1	<u>75.3</u>	88.8	83.0	85.9	82.1	82.5
REPLACE	<u>93.5</u>	<u>90.2</u>	<u>80.9</u>	<u>88.2</u>	74.0	75.5	74.8	90.9	88.0	<u>89.5</u>	<u>84.2</u>	<u>84.7</u>
CLIP+CLoVE w/o patching	<u>93.0</u>	91.0	81.6	88.6	74.4	77.9	76.1	86.2	94.7	90.5	85.1	85.5
CLIP+CLoVE ($\alpha = .6$)	93.8	89.1	78.2	87.0	74.4	74.8	74.6	84.4	87.3	85.8	82.5	83.1

Table D.1: Results on SugarCrepe. The best results are in **bold**. An underline indicates results within 1% of best.

	ImageNet	Cars	CIFAR10	CIFAR100	MNIST	EuroSAT	Flowers	DTD	UCF101	HMDB51	average	average drop
pre-trained	<u>59.0</u>	58.2	87.4	55.3	32.5	48.3	62.4	40.5	66.9	39.2	55.0	5.1
NegCLIP	54.4	45.6	85.1	57.9	31.8	30.3	51.3	37.2	64.1	38.4	49.6	3.4
REPLACE	52.4	41.9	83.3	58.0	29.3	32.8	45.4	33.8	60.7	39.5	47.7	2.4
CLIP+CLoVE w/o patching	50.3	50.6	85.2	61.8	37.8	39.7	37.9	36.3	61.7	35.2	49.7	<u>2.5</u>
CLIP+CLoVE ($\alpha = .6$)	59.3	<u>57.5</u>	88.6	64.6	34.6	<u>47.7</u>	54.7	43.5	68.0	42.3	56.1	4.3

Table D.2: Zero-shot classification results without employing text prompts, which is typically used for CLIP-like models. The best results are in **bold**. An underline indicates results within 1% of best.

	Text-to-Image		Image-to-Text		avg.
	Flickr30k	COCO Captions	Flickr30k	COCO Captions	
pre-trained	83.3	56.0	94.7	75.0	77.3
NegCLIP	<u>89.5</u>	68.5*	95.2	79.3*	83.1
REPLACE	<u>90.0</u>	73.8*	94.8	83.6*	85.6
CLIP+CLoVE w/o patching	87.2	65.8	87.4	68.8	77.3
CLIP+CLoVE ($\alpha = .6$)	90.3	68.1	96.3	80.0	83.7

Table D.3: Retrieval results for Flickr30k and COCO Captions. The evaluation is zero-shot except for those marked with an asterisk (*). The best results are in **bold**. An underline indicates results within 1% of best.

Table 7.3, even when it was tuned to see fully-formed sentences (as opposed to just class names like “husky”). When we apply the patching, it drops less in performance than the pre-trained model in seven out of ten benchmarks and is on par in two.

D.3 Performance in Flickr and COCO Retrieval Tasks

We evaluate the retrieval performance on Flickr30k [259] and COCO Captions [32], as it is sometimes reported with CLIP-like models [183]. We do not include these results with the main retrieval results because we believe they are near-shot or not zero-shot (in-domain). NegCLIP and REPLACE fine-tuned on COCO’s training set. Our method is trained on LAION-COCO, whose captions follow a format similar to COCO’s. At the same, COCO images come from Flickr. We present the results in Table D.3.

BIBLIOGRAPHY

- [1] Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. Video description: A survey of methods, datasets, and evaluation metrics. *ACM Comput. Surv.*, 52(6), October 2019.
- [2] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. YouTube-8M: A large-scale video classification benchmark, 2016.
- [3] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4971–4980, June 2018.
- [4] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C Lawrence Zitnick, Devi Parikh, and Dhruv Batra. VQA: Visual question answering. *International Journal of Computer Vision*, 123(1):4–31, 2017.
- [5] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. VATT: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34, 2021.
- [6] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 64–73, June 2016.
- [7] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5803–5812, October 2017.
- [8] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433, December 2015.

- [9] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6836–6846, October 2021.
- [10] Salvatore Attardo, Jodi Eisterhold, Jennifer Hay, and Isabella Poggi. Multimodal markers of irony and sarcasm. *Humor*, 16(2):243–260, 2003.
- [11] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in Time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*, 2021.
- [12] Ondrej Bajgar, Rudolf Kadlec, and Jan Kleindienst. Embracing data abundance: Booktest dataset for reading comprehension. In *ICLR 2017 — Workshop Track*, 2016.
- [13] Farah Benamara. Cooperative question answering in restricted domains: the WEBCOOP experiment. In *Proceedings of the Conference on Question Answering in Restricted Domains*, pages 31–38, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [14] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 37–40. Springer, 2009.
- [15] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions, 2023.
- [16] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O’Reilly Media, Inc., 2009.
- [17] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000.
- [18] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4613–4623, 2020.
- [19] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911, 2014.
- [20] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the ”video” in video-language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2917–2927, June 2022.
- [21] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. COYO-700M: Image-text pair dataset. <https://github.com/kakaobrain/coyo-dataset>, 2022.

- [22] Evgeny Byvatov, Uli Fechner, Jens Sadowski, and Gisbert Schneider. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *Journal of chemical information and computer sciences*, 43(6):1882–1889, 2003.
- [23] Michele Cafagna, Kees van Deemter, and Albert Gatt. What vision-language models ‘see’ when they see scenes. *ArXiv*, abs/2109.07301, 2021.
- [24] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, July 2017.
- [25] Richard Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Morgan Kaufmann, 1993.
- [26] Santiago Castro, Mahmoud Azab, Jonathan Stroud, Cristina Noujaim, Ruoyao Wang, Jia Deng, and Rada Mihalcea. LifeQA: A real-life dataset for video question answering. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4352–4358, Marseille, France, May 2020. European Language Resources Association.
- [27] Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. Towards multimodal sarcasm detection (an `_Obviously_` perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4619–4629, Florence, Italy, July 2019. Association for Computational Linguistics.
- [28] Santiago Castro, Oana Ignat, and Rada Mihalcea. Scalable performance analysis for vision-language models. In Alexis Palmer and Jose Camacho-collados, editors, *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 284–294, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [29] Santiago Castro, Ruoyao Wang, Pingxuan Huang, Ian Stewart, Oana Ignat, Nan Liu, Jonathan Stroud, and Rada Mihalcea. FIBER: Fill-in-the-blanks as a challenging video understanding evaluation framework. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2925–2940, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [30] David Chen and William Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, Portland, Oregon, USA, 2011. Association for Computational Linguistics.
- [31] Shizhe Chen and Dong Huang. Elaborative rehearsal for zero-shot action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13638–13647, 2021.

- [32] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.
- [33] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: Learning UNiversal image-TExt representations. In *European conference on computer vision*, pages 104–120. Springer, 2020.
- [34] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *arXiv preprint arXiv:2109.04290*, 2021.
- [35] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.
- [36] Seongho Choi, Kyoung-Woon On, Yu-Jung Heo, Ahjeong Seo, Youwon Jang, Minsu Lee, and Byoung-Tak Zhang. DramaQA: Character-centered video story understanding with hierarchical qa. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2):1166–1174, May 2021.
- [37] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [38] Anthony Colas, Seokhwan Kim, Franck Dernoncourt, Siddhesh Gupte, Zhe Wang, and Doo Soon Kim. TutorialVQA: Question answering dataset for tutorial videos. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5450–5455, Marseille, France, 2020. European Language Resources Association.
- [39] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, November 2011.
- [40] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [41] Casper da Costa-Luis, Stephen Karl Larroque, Kyle Altendorf, Hadrien Mary, richard-sheridan, Mikhail Korobov, Noam Raphael, Ivan Ivanov, Marcel Bargull, Nishant Rodrigues, Guangshuo Chen, Antony Lee, Charles Newey, CrazyPython, JC, Martin Zugnoni, Matthew D. Pagel, mjstevens777, Mikhail Dektyarev, Alex Rothberg, Alexander Plavin, Daniel Panteleit, Fabian Dill, FichteFoll, Gregor Sturm, HeoHeo, Hugo van Kemenade, Jack McCracken, MapleCCC, and Max Nordlund. tqdm: A fast, Extensible Progress Bar for Python and CLI, March 2023.
- [42] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and

- Michael Wray. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [43] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose M. F. Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [44] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, June 2009.
- [45] Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8599–8603. IEEE, 2013.
- [46] Karan Desai and Justin Johnson. VirTex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11162–11173, June 2021.
- [47] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [48] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics.
- [49] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.
- [50] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- [51] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. Temporal localization of moments in video collections with natural language. *ArXiv preprint*, abs/1907.12763, 2019.
- [52] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

- [53] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019.
- [54] Chenyou Fan. EgoVQA – an egocentric video question answering benchmark dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, October 2019.
- [55] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1999–2007. Computer Vision Foundation / IEEE, 2019.
- [56] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021.
- [57] Christiane Fellbaum. *Theory and Applications of Ontology: Computer Applications*, chapter WordNet, pages 231–243. Springer Netherlands, Dordrecht, 2010.
- [58] David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4991–5000, 2018.
- [59] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.
- [60] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
- [61] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. VIOLET: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021.
- [62] Chuang Gan, Yi Yang, Linchao Zhu, Deli Zhao, and Yueting Zhuang. Recognizing an action using its name: A knowledge-based approach. *International Journal of Computer Vision*, 120(1):61–77, 2016.
- [63] Jiyang Gao, Runzhou Ge, Kan Chen, and Ram Nevatia. Motion-appearance co-memory networks for video question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6576–6585. IEEE Computer Society, 2018.
- [64] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: temporal activity localization via language query. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 5277–5285. IEEE Computer Society, 2017.

- [65] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8303–8311, Jul. 2019.
- [66] Lianli Gao, Pengpeng Zeng, Jingkuan Song, Yuan-Fang Li, Wu Liu, Tao Mei, and Heng Tao Shen. Structured two-stream attention network for video question answering. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6391–6398. AAAI Press, 2019.
- [67] Qiaozhi Gao, Malcolm Doering, Shaohua Yang, and Joyce Yue Chai. Physical causality of action verbs in grounded language understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.
- [68] Qiaozhi Gao, Shaohua Yang, Joyce Yue Chai, and Lucy Vanderwende. What action causes this? towards naive physical action-effect prediction. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 934–945. Association for Computational Linguistics, 2018.
- [69] Noa Garcia, Mayu Otani, Chenhui Chu, and Yuta Nakashima. KnowIT VQA: answering knowledge-based questions about videos. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 10826–10834. AAAI Press, 2020.
- [70] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8803–8812, 2018.
- [71] Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo. Bridging video-text retrieval with multiple choice questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16167–16176, June 2022.
- [72] Rohit Girdhar, Du Tran, Lorenzo Torresani, and Deva Ramanan. DistInit: Learning video representations without a single labeled video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

- [73] Ross B. Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1440–1448. IEEE Computer Society, 2015.
- [74] Shreyank N Gowda, Marcus Rohrbach, and Laura Sevilla-Lara. Smart frame selection for action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2):1451–1459, May 2021.
- [75] Shreyank N. Gowda, Laura Sevilla-Lara, Kiyoon Kim, Frank Keller, and Marcus Rohrbach. A new split for evaluating true zero-shot action recognition. In Christian Bauckhage, Juergen Gall, and Alexander Schwing, editors, *Pattern Recognition*, pages 191–205, Cham, 2021. Springer International Publishing.
- [76] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. *ArXiv preprint*, abs/2110.07058, 2021.
- [77] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.
- [78] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [79] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: a large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, June 7-12*, pages 961–970. IEEE Computer Society, 2015.
- [80] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 204–207. IEEE, 2018.
- [81] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.
- [82] Lisa Anne Hendricks and Aida Nematzadeh. Probing image-language transformers for verb understanding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3635–3644, Online, August 2021. Association for Computational Linguistics.
- [83] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. Localizing moments in video with natural language. In *IEEE*

- International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 5804–5813. IEEE Computer Society, 2017.
- [84] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.
- [85] Geoffrey Hinton. How to represent part-whole hierarchies in a neural network, 2021.
- [86] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *Deep Learning and Representation Learning Workshop at the Twenty-eighth Conference on Neural Information Processing Systems*, 2014.
- [87] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.
- [88] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. SugarCreme: Fixing hackable benchmarks for vision-language compositionality. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.
- [89] Deng Huang, Peihao Chen, Runhao Zeng, Qing Du, Mingkui Tan, and Chuang Gan. Location-aware graph convolutional networks for video question answering. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 11021–11028. AAAI Press, 2020.
- [90] Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. Natural language to structured query generation via meta-learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 732–738, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [91] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiase Wang, and Dahua Lin. MovieNet: A holistic dataset for movie understanding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 709–727, Cham, 2020. Springer International Publishing.
- [92] Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6700–6709, June 2019.
- [93] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [94] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary

- models by interpolating weights. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 29262–29277. Curran Associates, Inc., 2022.
- [95] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, July 2021.
- [96] Allan Jabri, Armand Joulin, and Laurens van der Maaten. Revisiting visual question answering baselines. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 727–739, Cham, 2016. Springer International Publishing.
- [97] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online, 2020. Association for Computational Linguistics.
- [98] Mihir Jain, Jan C Van Gemert, Thomas Mensink, and Cees GM Snoek. Objects2action: Classifying and localizing actions without any video example. In *Proceedings of the IEEE international conference on computer vision*, pages 4588–4596, 2015.
- [99] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2758–2766, 2017.
- [100] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021.
- [101] Jianwen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 11101–11108. AAAI Press, 2020.
- [102] Weike Jin, Zhou Zhao, Mao Gu, Jun Yu, Jun Xiao, and Yueting Zhuang. Multi-interaction network with object relation for video question answering. In *Proceedings of the 27th ACM International Conference on Multimedia, MM ’19*, page 1193–1201, New York, NY, USA, 2019. Association for Computing Machinery.
- [103] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

- [104] Armand Joulin, Laurens Van Der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pages 67–84. Springer, 2016.
- [105] Amita Kamath, Jack Hessel, and Kai-Wei Chang. Text encoders bottleneck compositionality in contrastive vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4933–4944, Singapore, December 2023. Association for Computational Linguistics.
- [106] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
- [107] Boris Katz, Sue Felshin, Deniz Yuret, Ali Ibrahim, Jimmy Lin, Gregory Marton, Alton Jerome McFarland, and Baris Temelkuran. Omnibase: Uniform access to heterogeneous data for question answering. In *International Conference on Application of Natural Language to Information Systems*, pages 230–234. Springer, 2002.
- [108] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.
- [109] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.
- [110] Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*, 2020.
- [111] Kyung-Min Kim, Seong-Ho Choi, Jin-Hwa Kim, and Byoung-Tak Zhang. Multimodal dual attention memory for video story question answering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 673–688, 2018.
- [112] Kyung-Min Kim, Min-Oh Heo, Seong-Ho Choi, and Byoung-Tak Zhang. DeepStory: Video story QA by deep embedded memory networks. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 2016–2022. AAAI Press, 2017.
- [113] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, December 2014.

- [114] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter development team. Jupyter Notebooks – a publishing format for reproducible computational workflows. In Fernando Loizides and Birgit Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90, Netherlands, 2016. IOS Press.
- [115] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. MoViNets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16020–16030, June 2021.
- [116] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, June 2013.
- [117] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 706–715, Oct 2017.
- [118] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.
- [119] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- [120] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563, 2011.
- [121] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563, 2011.
- [122] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2, AAAI’08*, page 646–651, 2008.
- [123] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The MNIST database of handwritten digits., 1994.
- [124] Chia-Hsuan Lee, Oleksandr Polozov, and Matthew Richardson. KaggleDBQA: Realistic evaluation of text-to-SQL parsers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2261–2273, Online, August 2021. Association for Computational Linguistics.

- [125] Jie Lei, Tamara L. Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning, 2022.
- [126] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: ClipBERT for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7331–7341, June 2021.
- [127] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. TVQA: Localized, compositional video question answering. In *2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
- [128] Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. TVQA+: Spatio-temporal grounding for video question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8211–8225, Online, 2020. Association for Computational Linguistics.
- [129] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. TVR: A large-scale dataset for video-subtitle moment retrieval. In *European Conference on Computer Vision*, pages 447–463. Springer, 2020.
- [130] Beth Levin. *English verb classes and alternations: A preliminary investigation*. University of Chicago press, 1993.
- [131] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In Heike Adel and Shuming Shi, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
- [132] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 17–23 Jul 2022.
- [133] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: Hierarchical encoder for Video+Language omni-representation pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2046–2065, Online, November 2020. Association for Computational Linguistics.

- [134] Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, Yu Cheng, Luwei Zhou, Xin Eric Wang, William Yang Wang, et al. VALUE: A multi-task benchmark for video-and-language understanding evaluation. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks*, 2021.
- [135] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A simple and performant baseline for vision and language. *ArXiv*, abs/1908.03557, 2019.
- [136] Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2592–2607, Online, 2021. Association for Computational Linguistics.
- [137] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 8658–8665. AAAI Press, 2019.
- [138] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. TGIF: A New Dataset and Benchmark on Animated GIF Description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4641–4650, June 2016.
- [139] Zhuang Li, Lizhen Qu, Shuo Huang, and Gholamreza Haffari. Few-shot semantic parsing for new predicates. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1281–1291, Online, April 2021. Association for Computational Linguistics.
- [140] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [141] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.
- [142] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing human actions by attributes. In *CVPR 2011*, pages 3337–3344. IEEE, 2011.

- [143] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *30th British Machine Vision Conference (BMVC 2019)*. British Machine Vision Association, April 2020.
- [144] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach, 2019.
- [145] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.
- [146] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.
- [147] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [148] Huaishao Luo, Lei Ji, Botian Shi, H. Huang, N. Duan, Tianrui Li, X. Chen, and M. Zhou. UniVL: A unified video and language pre-training model for multimodal understanding and generation. *ArXiv*, abs/2002.06353, 2020.
- [149] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*, 2021.
- [150] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. Crepe: Can vision-language foundation models reason compositionally? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10910–10921, June 2023.
- [151] Tegan Maharaj, Nicolas Ballas, Anna Rohrbach, Aaron C. Courville, and Christopher Joseph Pal. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 7359–7368. IEEE Computer Society, July 2017.
- [152] TorchVision maintainers and contributors. TorchVision: PyTorch’s computer vision library. <https://github.com/pytorch/vision>, 2016.
- [153] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1682–1690, 2014.

- [154] Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. Out-of-distribution detection for generalized zero-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9985–9993, 2019.
- [155] Clara Meister, Ryan Cotterell, and Tim Vieira. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online, November 2020. Association for Computational Linguistics.
- [156] Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. RareAct: A video dataset of unusual interactions, 2020.
- [157] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [158] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2630–2640, October 2019.
- [159] Abhijit Mishra, Diptesh Kanojia, and Pushpak Bhattacharyya. Predicting readers’ sarcasm understandability by modeling gaze behavior. In *AAAI*, pages 3747–3753, 2016.
- [160] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, and Aude Oliva. Moments in time dataset: One million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):502–508, 2020.
- [161] Yasuhide Mori, Hironobu Takahashi, and Ryuichi Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *First international workshop on multimedia intelligent storage and retrieval management*, pages 1–9. Citeseer, 1999.
- [162] Subhabrata Mukherjee and Ahmed Awadallah. Uncertainty-aware self-training for few-shot text classification. *Advances in Neural Information Processing Systems*, 33:21199–21212, 2020.
- [163] Jonghwan Mun, Paul Hongsuck Seo, Ilchae Jung, and Bohyung Han. MarioQA: Answering questions by watching gameplay videos. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2886–2894. IEEE Computer Society, 2017.
- [164] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, and Cordelia Schmid. Learning audio-video modalities from image captions, 2022.

- [165] J. Zvi Namenwirth and Robert Philip Weber. *Dynamics of culture*. Allen & Unwin – Boston, Mass., USA, 1987.
- [166] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@NIPS*, 2016.
- [167] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008.
- [168] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [169] OpenAI. GPT-4V(ision) System Card. Technical report, OpenAI, 2023.
- [170] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- [171] Letitia Parcalabescu, Michele Cafagna, Lilitta Muradjan, Anette Frank, Iacer Calixto, and Albert Gatt. VALSE: A task-independent benchmark for vision and language models centered on linguistic phenomena. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8253–8280, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [172] Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword fifth edition. *Linguistic Data Consortium*, 2011.
- [173] Barbara Partee et al. Compositionality. *Varieties of formal semantics*, 3:281–311, 1984.
- [174] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [175] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [176] James W. Pennebaker, Roger John Booth, and Martha E. Francis. Linguistic inquiry and word count (LIWC2007), 2007.

- [177] James W. Pennebaker, Ryan L. Boyd, Kayla Jordan, and Kate G. Blackburn. The development and psychometric properties of LIWC2015, 2015.
- [178] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [179] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007.
- [180] Jonas Pfeiffer, Gregor Geigle, Aishwarya Kamath, Jan-Martin Steitz, Stefan Roth, Ivan Vulić, and Iryna Gurevych. xGQA: Cross-lingual visual question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2497–2511, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [181] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy, July 2019. Association for Computational Linguistics.
- [182] Jesús Andrés Portillo-Quintero, José Carlos Ortiz-Bayliss, and Hugo Terashima-Marín. A straightforward framework for video retrieval using clip. In Edgar Roman-Rangel, Ángel Fernando Kuri-Morales, José Francisco Martínez-Trinidad, Jesús Ariel Carrasco-Ochoa, and José Arturo Olvera-López, editors, *Pattern Recognition*, pages 3–12, Cham, 2021. Springer International Publishing.
- [183] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.
- [184] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [185] Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM '15*, page 97–106, New York, NY, USA, 2015. Association for Computing Machinery.
- [186] Nazneen Rajani, Weixin Liang, Lingjiao Chen, Margaret Mitchell, and James Zou. SEAL: Interactive tool for systematic error analysis and labeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System*

- Demonstrations*, pages 359–370, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics.
- [187] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.
- [188] Lance Ramshaw and Mitch Marcus. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, 1995.
- [189] Royi Rassin, Eran Hirsch, Daniel Glickman, Shauli Ravfogel, Yoav Goldberg, and Gal Chechik. Linguistic binding in diffusion models: Enhancing attribute correspondence through attention map alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [190] Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan A. Plummer, Ranjay Krishna, and Kate Saenko. Cola: A benchmark for compositional text-to-image retrieval. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.
- [191] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [192] Mengye Ren, Ryan Kiros, and Richard S. Zemel. Exploring models and data for image question answering. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2953–2961, 2015.
- [193] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [194] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 193–203, 2013.
- [195] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 123(1):94–120, May 2017.
- [196] Amelie Royer and Christoph H Lampert. Classifier adaptation at prediction time. In *CVPR*, pages 1401–1409, 2015.

- [197] Arka Sadhu, Kan Chen, and Ram Nevatia. Video question answering with phrases via semantic roles. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2460–2478, Online, June 2021. Association for Computational Linguistics.
- [198] Emmanuelle Salin, Badreddine Farah, S. Ayache, and Benoit Favre. Are vision-language transformers learning multimodal representations? a probing perspective. In *AAAI Conference on Artificial Intelligence*, pages 11248–11257, June 2022.
- [199] Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. How2: a large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. NeurIPS, 2018.
- [200] R Schifanella, P de Juan, J Tetreault, L Cao, et al. Detecting sarcasm in multimodal social platforms. In *ACM Multimedia*, pages 1136–1145. ACM, 2016.
- [201] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.
- [202] Christoph Schuhmann, Andreas Köpf, Theo Coombes, Richard Vencu, Benjamin Trom, and Romain Beaumont. LAION COCO: 600M synthetic captions from LAION2B-EN, September 2022.
- [203] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open dataset of CLIP-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.
- [204] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.
- [205] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *ICLR*, 2017.
- [206] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics.
- [207] Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurélie Herbelot, Moin Nabi, Enver Sangineto, and Raffaella Bernardi. FOIL it! find one mismatch between image and language caption. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the*

- 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 255–265, Vancouver, Canada, July 2017. Association for Computational Linguistics.
- [208] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 510–526, Cham, 2016. Springer International Publishing.
- [209] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. FLAVA: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15638–15650, June 2022.
- [210] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
- [211] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CRCV-TR-12-01*, November 2012.
- [212] Radu Soricut and Eric Brill. Automatic question answering: Beyond the factoid. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, 2004.
- [213] Robyn Speer. ftfy. Zenodo, 2019. Version 5.5.
- [214] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. *WIT: Wikipedia-Based Image Text Dataset for Multimodal Multilingual Machine Learning*, page 2443–2449. Association for Computing Machinery, New York, NY, USA, 2021.
- [215] Nitish Srivastava, Ruslan Salakhutdinov, et al. Multimodal learning with deep boltzmann machines. In *NIPS*, volume 1, page 2. Citeseer, 2012.
- [216] Philip J. Stone, Dexter C. Dunphy, and Marshall S. Smith. The general inquirer: A computer approach to content analysis. *American Educational Research Journal*, 4:397, 1967.
- [217] Student. The probable error of a mean. *Biometrika*, pages 1–25, 1908.
- [218] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2440–2448. Curran Associates, Inc., 2015.

- [219] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019.
- [220] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, October 2019.
- [221] Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 2019.
- [222] Yibo Sun, Duyu Tang, Nan Duan, Yeyun Gong, Xiaocheng Feng, Bing Qin, and Daxin Jiang. Neural semantic parsing in low-resource settings with back-translation and meta-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8960–8967, Apr. 2020.
- [223] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [224] O. Tange. Gnu parallel - the command-line power tool. *login: The USENIX Magazine*, 36(1):42–47, Feb 2011.
- [225] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelwagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016.
- [226] The pandas development team. `pandas-dev/pandas`: Pandas, 2021.
- [227] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.
- [228] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5238–5248, June 2022.
- [229] Du Tran, Maksim Bolonkin, Manohar Paluri, and Lorenzo Torresani. VideoMCC: a new benchmark for video comprehension, 2016.
- [230] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

- [231] Michael Tschannen, Manoj Kumar, Andreas Peter Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [232] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.
- [233] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [234] Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, Raymond Mooney, Trevor Darrell, and Kate Saenko. Captioning images with diverse objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [235] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [236] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17(3):261–272, 2020.
- [237] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.
- [238] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.
- [239] Mengmeng Wang, Jiazheng Xing, and Yong Liu. ActionCLIP: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021.
- [240] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc., 2020.
- [241] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. VaTeX: A large-scale, high-quality multilingual dataset for video-and-language research. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 4580–4590. IEEE, 2019.

- [242] Michael Lawrence Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 60(6), April 2021.
- [243] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.
- [244] Ross Wightman. PyTorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019.
- [245] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.
- [246] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.
- [247] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7959–7971, June 2022.
- [248] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 9777–9786. Computer Vision Foundation / IEEE, 2021.
- [249] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 1645–1653, New York, NY, USA, 2017. Association for Computing Machinery.
- [250] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. VideoCLIP: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference*

- on *Empirical Methods in Natural Language Processing*, pages 6787–6800, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
- [251] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, June 2016.
- [252] Hongyang Xue, Zhou Zhao, and Deng Cai. Unifying the video and question attentions for open-ended video question answering. *IEEE Transactions on Image Processing*, 26(12):5656–5666, 2017.
- [253] Omry Yadan. Hydra – a framework for elegantly configuring complex applications. GitHub, 2019.
- [254] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just Ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1686–1697, October 2021.
- [255] Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. TACo: Token-aware cascade contrastive learning for video-text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11562–11572, October 2021.
- [256] Shaohua Yang, Qiaozhi Gao, Sari Saba-Sadiya, and Joyce Yue Chai. Commonsense justification for action explanation. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun’ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2627–2637. Association for Computational Linguistics, 2018.
- [257] Yunan Ye, Zhou Zhao, Yimeng Li, Long Chen, Jun Xiao, and Yueting Zhuang. Video question answering via attribute-augmented attention network learning. In Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White, editors, *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 829–832. ACM, 2017.
- [258] Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1744–1753, 2013.
- [259] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

- [260] Ting Yu, Jun Yu, Zhou Yu, Qingming Huang, and Qi Tian. Long-term video question answering via multimodal hierarchical memory attentive networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(3):931–944, 2020.
- [261] Ting Yu, Jun Yu, Zhou Yu, and Dacheng Tao. Compositional attention networks with two-stream fusion for video question answering. *IEEE Transactions on Image Processing*, pages 1204–1218, 2019.
- [262] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [263] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. ActivityNet-QA: A dataset for understanding complex web videos via question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9127–9134, Jul. 2019.
- [264] Mert Yuksekogunul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2023.
- [265] Amir Zadeh, Michael Chan, Paul Pu Liang, Edmund Tong, and Louis-Philippe Morency. Social-IQ: A question answering benchmark for artificial social intelligence. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8807–8817. Computer Vision Foundation / IEEE, 2019.
- [266] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [267] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. MERLOT: Multimodal neural script knowledge models. In *Advances in Neural Information Processing Systems 34*, 2021.
- [268] Kuo-Hao Zeng, Tseng-Hung Chen, Ching-Yao Chuang, Yuan-Hong Liao, Juan Carlos Niebles, and Min Sun. Leveraging video descriptions to learn video question answering. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4334–4340. AAAI Press, 2017.
- [269] Da Zhang, Xiyang Dai, Xin Wang, and Yuan-Fang Wang. S3D: Single shot multi-span detector via fully 3d convolutional network. In *Proceedings of the British Machine Vision Conference*, 2018.
- [270] Lingling Zhang, Xiaojun Chang, Jun Liu, Minnan Luo, Sen Wang, Zongyuan Ge, and Alexander Hauptmann. Zstad: Zero-shot temporal activity detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 879–888, 2020.

- [271] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5014–5022. IEEE Computer Society, 2016.
- [272] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. VinVL: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5579–5588, June 2021.
- [273] Wenqiao Zhang, Siliang Tang, Yanpeng Cao, Shiliang Pu, Fei Wu, and Yueting Zhuang. Frame augmented alternating attention network for video question answering. *IEEE Transactions on Multimedia*, 22(4):1032–1041, 2019.
- [274] Ziqi Zhang, Zhongang Qi, Chunfeng Yuan, Ying Shan, Bing Li, Ying Deng, and Weiming Hu. Open-book video captioning with retrieve-copy-generate network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9837–9846, June 2021.
- [275] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
- [276] Tiancheng Zhao, Tianqi Zhang, Mingwei Zhu, Haozhan Shen, Kyusong Lee, Xiaopeng Lu, and Jianwei Yin. VL-CheckList: Evaluating pre-trained vision-language models with objects, attributes and relations. *arXiv preprint arXiv:2207.00221*, 2022.
- [277] Zhou Zhao, Qifan Yang, Deng Cai, Xiaofei He, and Yueting Zhuang. Video question answering via hierarchical spatio-temporal attention networks. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3518–3524. ijcai.org, 2017.
- [278] Zhou Zhao, Zhu Zhang, Shuwen Xiao, Zhou Yu, Jun Yu, Deng Cai, Fei Wu, and Yueting Zhuang. Open-ended long-form video question answering via adaptive hierarchical reinforced networks. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 3683–3689. ijcai.org, 2018.
- [279] Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges, 2022.
- [280] Luwei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598, April 2018.

- [281] Linchao Zhu, Zhongwen Xu, Yi Yang, and Alexander G. Hauptmann. Uncovering the temporal context for video question answering. *International Journal of Computer Vision*, 124(3):409–421, Sep 2017.
- [282] Linchao Zhu and Yi Yang. ActBERT: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8746–8755, June 2020.
- [283] Yuke Zhu, Oliver Groth, Michael S. Bernstein, and Li Fei-Fei. Visual7W: Grounded question answering in images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4995–5004. IEEE Computer Society, 2016.